

Table of Contents

Table of Contents	1
ABS	8
ACOS	9
ADDTEXT	10
AFTER	11
ANIMATIONHEIGHT	12
PLAYANIMATION	13
ANIMATIONWIDTH	14
AS	15
ASC	16
ASIN	18
ATAN	19
BEFORE	20
BIN	21
BOX	22
BOXOVERLAP	23
CALLDLL	24
CAMERAANGLE	25
CAMERAFOLLOW	26
CAMERAPICK	28
CAMERAX	30
CAMERAY	31
CENTERTEXT	32
CHDIR	33
CHR	34
CIRCLE	35
CLEARARRAY	36
CLEARCOLLISIONS	37
CLEARKEYS	38
CLEARMOUSE	39
CLEAROBJECTS	40
CLEARTEXT	41
CLONECAMERAORIENTATION	42
CLONECAMERAPOSITION	43
CLONEIMAGE	44
CLONEOBJECT	45
CLONEOBJECTORIENTATION	47
CLONEOBJECTPOSITION	48
CLOSEFILE	49
CLOSEINPUT	50
CLS	51
CLSCOLOR	52
COLLISIONANGLE	53
COLLISIONX	55
COLLISIONY	56
COLOR	57
COMMANDLINE	59
CONST	60
CONVERTTOINTEGER	61
CONVERTTOTYPE	62

COPYBOX	63
COPYFILE	64
COS	65
COUNTCOLLISIONS	67
COUNTWORDS	68
CRC32	69
CURRENTDIR	70
CURVEANGLE	71
CURVEVALUE	72
DATA	73
DATE	74
DECRYPT	75
DEFAULTMASK	76
DEFAULTVISIBLE	77
DELETE	78
DELETEFILE	80
DELETEFONT	81
DELETEIMAGE	82
DELETEMEMBLOCK	83
DELETEOBJECT	84
DELETESOUND	85
DIM	86
DISTANCE	87
DISTANCE2	88
DOT	89
DOWNKEY	90
DRAWANIMATION	91
DRAWGAME	92
DRAWGHOSTIMAGE	93
DRAWIMAGE	94
DRAWIMAGEBOX	95
DRAWSCREEN	96
DRAWTOIMAGE	97
DRAWTOSCREEN	98
DRAWTOWORLD	99
FOR	101
EDITMAP	102
ELLIPSE	103
ENCRYPT	104
END	105
ENDSEARCH	106
EOF	107
ERRORS	108
ESCAPEKEY	109
EXECUTE	110
EXIT	111
TYPE ja FIELD	112
FILEEXISTS	114
FILEOFFSET	115
FILESIZE	116
FINDFILE	117
FIRST	118

FLIP	119
FLOAT	120
FOR	121
FPS	122
FRAMELIMIT	123
FUNCTION	124
GETANGLE	125
GETANGLE2	126
GETCOLLISION	127
GETEXENAME	128
GETKEY	129
GETMAP	130
GETMAP2	131
GETMOUSE	132
GETPIXEL	133
GETPIXEL2	134
GETRGB	135
GETWORD	137
GFXMODEEXISTS	138
GHOSTOBJECT	139
GLOBAL	140
GOSUB	141
GOTO	142
GOTOSAVEDLOCATION	143
HEX	144
HOTSPOT	145
IF	146
IMAGE	148
IMAGEHEIGHT	149
IMAGESCOLLIDE	150
IMAGESOVERLAP	151
IMAGEWIDTH	152
INCLUDE	153
Komennot	154
INITOBJECTLIST	155
INPUT	156
INSERT	158
INSTR	159
INT	160
ISDIRECTORY	161
KEYDOWN	162
KEYHIT	163
KEYUP	164
LAST	165
LEFT	166
LEFTKEY	167
LEN	168
LINE	169
LOADANIMIMAGE	170
LOADANIMOBJECT	171
LOADFONT	173
LOADIMAGE	174

LOADMAP	175
LOADOBJECT	177
LOADPROGRAM	178
LOADSOUND	179
LOCATE	180
LOCK	181
LOG	182
LOG10	183
Loogiset operaattorit AND, OR, XOR ja NOT	184
LOOPOBJECT	187
LOWER	189
LSET	190
MAKEDIR	191
MAKEEMITTER	192
MAKEERROR	193
MAKEIMAGE	194
MAKEMAP	195
MAKEMEMBLOCK	196
MAKEOBJECT	197
MAKEOBJECTFLOOR	198
MAPHEIGHT	199
MAPWIDTH	200
MASKIMAGE	201
MASKOBJECT	202
MAX	203
MEMBLOCKSIZE	204
MEMCOPY	205
MID	206
MIN	207
MIRROROBJECT	208
MOD	209
MOUSEDOWN	210
MOUSEHIT	211
MOUSEMOVEX	212
MOUSEMOVEY	213
MOUSEMOVEZ	214
MOUSEUP	215
MOUSEWX	216
MOUSEWY	217
MOUSEX	218
MOUSEY	219
MOUSEZ	220
MOVECAMERA	221
MOVEOBJECT	222
NEW	223
NEXTOBJECT	225
OBJECTANGLE	226
OBJECTFLOAT	227
OBJECTFRAME	228
OBJECTINTEGER	229
OBJECTLIFE	230
OBJECTORDER	231

OBJECTPICK	232
OBJECTPICKABLE	234
OBJECTPLAYING	236
OBJECTRANGE	237
OBJECTSIGHT	238
OBJECTSIZEX	239
OBJECTSIZEY	240
OBJECTSOVERLAP	241
OBJECTSTRING	243
OBJECTX	244
OBJECTY	245
OPENTOEDIT	246
OPENTOREAD	247
OPENTOWRITE	248
PAINTOBJECT	249
PARTICLEANIMATION	250
PARTICLEEMISSION	251
PARTICLEMOVEMENT	252
PEEKBYTE	253
PEEKFLOAT	254
PEEKINT	255
PEEKSHORT	256
PICKCOLOR	257
PICKEDANGLE	258
PICKEDOBJECT	259
PICKEDX	261
PICKEDY	263
PICKIMAGECOLOR	265
PICKIMAGECOLOR2	266
PIXELPICK	267
PLAYANIMATION	269
PLAYOBJECT	270
PLAYSOUND	272
POINTCAMERA	273
POINTOBJECT	274
POKEBYTE	275
POKEFLOAT	276
POKEINT	277
POKESHORT	278
POSITIONCAMERA	279
POSITIONMOUSE	280
POSITIONOBJECT	281
PRINT	282
PUTPIXEL	283
PUTPIXEL2	284
RAND	285
RANDOMIZE	286
READ	287
READBYTE	288
READFLOAT	289
READINT	290
READLINE	291

READSHORT	292
READSTRING	293
REDIM	294
REPEAT	295
REPLACE	296
RESETOBJECTCOLLISION	297
RESIZEIMAGE	298
RESIZEMEMBLOCK	299
RESTORE	300
RETURN	301
RIGHT	302
RIGHTKEY	303
RND	304
ROTATECAMERA	305
ROTATEIMAGE	306
ROTATEOBJECT	307
ROUNDDOWN	308
ROUNDUP	309
RSET	310
SAFEEXIT	311
SAR	312
SAVEIMAGE	313
SAVEPROGRAM	314
SCREEN	315
SCREENDEPTH	316
SCREENGAMMA	317
SCREENHEIGHT	318
SCREENPOSITIONOBJECT	319
SCREENSHOT	320
SCREENWIDTH	321
SEEKFILE	322
SELECT	323
SETFONT	326
SETMAP	327
SETSOUND	328
SETTILE	329
SETUPCOLLISION	330
SETWINDOW	332
SHL	333
SHOWMOUSE	334
SHOWOBJECT	335
SHR	336
SIN	337
SMOOTH2D	339
SOUNDPLAYING	340
SQRT	341
STARTSEARCH	342
STOPANIMATION	343
STOPOBJECT	344
STOPSOUND	345
STR	346
STRING	347

STRINSERT	348
STRMOVE	349
STRREMOVE	350
TAN	351
TEXT	352
TEXTHEIGHT	353
TEXTWIDTH	354
TIME	355
TIMER	356
TRANSLATECAMERA	357
TRANSLATEOBJECT	358
TRIM	360
TURNCAMERA	361
TURNOBJECT	362
TYPE ja FIELD	363
UNLOCK	365
UPDATEGAME	366
UPKEY	367
UPPER	368
VERTICALTEXT	369
WAIT	370
WAITKEY	371
WAITKEY	372
WHILE	373
WRAPANGLE	374
WRITE	375
WRITEBYTE	376
WRITEFLOAT	377
WRITEINT	378
WRITELINE	379
WRITESHORT	380
WRITESTRING	381

ABS

KUVAUS

Palauttaa annetun luvun itseisarvon.

Itseisarvo on luku itse, mutta aina positiivinen. Siten esim. **-7 -> 7**.

Voit ajatella kaikkia maailman lukuja lukusuorana. Se jatkuu äärettömänä kumpaankin suuntaan. Vasemmalla on negatiiviset ja oikealla positiiviset luvut. Keskellä on nolla. Itseisarvo on siis tavallaan luvun etäisyys nolasta.

KÄYTTÖ

ABS (luku)

- **luku** = Kokonaisluku tai desimaaliluku.

ESIMERKKI

```
'First try it with some decimal numbers
For i=1 To 5

value#=Rnd(-10,10)

Text 0, i*12, "original: "+value#
Text 200, i*12, "-> ABS -> "+Abs(value#)

Next i

'Then do the same with integer numbers
For i=7 To 11

val=Rand(-10,10)

Text 0, i*12, "original: "+val
Text 200, i*12, "-> ABS -> "+Abs(val)

Next i

DrawScreen
WaitKey
```


ACOS

KUVAUS

Tämä on Cos-funktion päinvastainen laskutoimitus. Eli palautetaan alkuperäinen kulma.

Tulos on desimaaliluku.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

ACOS (luku)

- **luku** = Kokonaisluku tai desimaaliluku.

Katso myös: [COS](#)

ESIMERKKI

ADDEXT

KUVAUS

Lisää ruudulle tekstiä. Pysyvästi. Käytä tätä komentoa silloin kun haluat luoda ruudulle tekstiä, joka muuttuu harvoin tai ei milloinkaan. Esimerkiksi pelaajan healtti, jäljellä olevat patit, pelaajan nimi, kokemus tai vaikkapa konsoli.

Voit asettaa tekstin sijainnin käyttämällä **Locate**-komentoa. Jokaisen AddTextin jälkeen riviä vaihdetaan automaattisesti, mutta X-sijainti pysyy samana.

Huomaa, että komento ei sovellu kutsuttavaksi silmukassa. Sitä tulisi käyttää vain kerran, ja teksti jää sen jälkeen ruudulle. Siitä ei tarvitse huolehtia sen enempää :)

Tekstin väri ja paikka pysyvät muistissa, joten esimerkiksi Color-komento ei sotke tekstirivejä.

Lisätyt rivit ovat paljon nopeampia kuin Text-komennon avulla aina tulostettu teksti. Käytä aina kun mahdollista.

Tällä komennolla lisätyt teksti voi pyyhkiä ruudulta pois vain komennolla **ClearText**.

KÄYTTÖ

```
ADDEXT teksti
```

- **teksti** = Tulostettava merkkijono.

Katso myös: [CLEARTEXT](#)

ESIMERKKI

```
FrameLimit 40

Locate 30,30

AddText "This text is sticky and won't go away"
AddText "unless you use the Cleartext command."
AddText "Thank god, this demo features also that"
AddText "command."
AddText ""

Color 255,0,0
AddText "Press SPACE to clear the text."

plusx=2
plusy=2

Repeat

    'update the game itself
    x=x+plusx
    y=y+plusy

    If x<0 Or x>400 Then plusx= -plusx
    If y<0 Or y>300 Then plusy= -plusy

    Color Rand(255),Rand(255),Rand(255)
    Circle x-20,y-20,40

    'clear the text?
    If KeyHit(cbkeyspace) Then ClearText

DrawScreen

Until EscapeKey()
```

AFTER

KUVAUS

Tämä funktio kertoo jäsenen jälkeisen jäsenen.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

AFTER (tyyppi_muuttuja)

- **tyyppi_muuttuja** = Jo valmis tyyplitetty muuttuja, joka toimii osoittimena listaan.

Katso myös: [FIRST](#), [LAST](#), [BEFORE](#)

ESIMERKKI

```
Type test
  Field value As Float
End Type

'First create 10 elements
For i=1 To 10
  t.test = New(test)

  t\value=Rnd(1)
Next i

'Print them all
Repeat

  t.test = First(test)
  i=10
  While t<> NULL
    Text 10,i,t\value
    i=i+16
    t = After(t)
  Wend

  DrawScreen

Until EscapeKey()
```

ANIMATIONHEIGHT

KUVAUS

Kertoo videon korkeuden pikseleinä. Käytä tätä funktiota määrittämään videon oikea koko, jos haluat esimerkiksi venyttää ikkunan videon kokoiseksi. Esimerkistä lisätietoa.

Palautettu luku on kokonaisluku.

KÄYTTÖ

```
ANIMATIONHEIGHT (videomuuttuja)
```

- **videomuuttuja** = Se muuttuja, mihin video käynnistettiin PlayAnimation-komennolla.

Katso myös: [ANIMATIONWIDTH](#)

ESIMERKKI

```
'load the video. Find out its size and restart it
video= PlayAnimation("myVideo.avi")
w= AnimationWidth(video)
h= AnimationHeight(video)
StopAnimation video

SCREEN w,h

video= PlayAnimation("myVideo.avi")

'draw video
Repeat
    DrawAnimation video
    DrawScreen

Until EscapeKey() Or KeyHit(28)

'close the video input
StopAnimation video
```

PLAYANIMATION

KUVAUS

Kertoo, jos video on vielä kesken. Palauttaa nollan (video on päättynyt) tai ykkösen (video soi vielä).

KÄYTTÖ

ANIMATIONPLAYING (videomuuttuja)

- **videomuuttuja** = Se muuttuja, jossa videota soitetaan.

ESIMERKKI

SCREEN 640,480

```
'load the video  
video= PlayAnimation("myMovie.avi")
```

```
'draw video
```

Repeat

```
    DrawAnimation video
```

```
    DrawScreen
```

```
Until EscapeKey() Or KeyHit(28) Or AnimationPlaying(video) = 0
```

```
'close the video input
```

```
StopAnimation video 'just in case it's not finished yet
```

ANIMATIONWIDTH

KUVAUS

Kertoo videon leveyden pikseleinä. Käytä tätä funktiota määrittämään videon oikea koko, jos haluat esimerkiksi venyttää ikkunan videon kokoiseksi. Esimerkistä lisätietoa.

Palautettu luku on kokonaisluku.

KÄYTTÖ

```
ANIMATIONWIDTH (videomuuttuja)
```

- **videomuuttuja** = Se muuttuja, mihin video käynnistettiin PlayAnimation-komennolla.

Katso myös: [ANIMATIONHEIGHT](#)

ESIMERKKI

```
'load the video. Find out its size and restart it
video= PlayAnimation("myVideo.avi")
w= AnimationWidth(video)
h= AnimationHeight(video)
StopAnimation video

SCREEN w,h

video= PlayAnimation("myVideo.avi")

'draw video
Repeat
    DrawAnimation video
    DrawScreen

Until EscapeKey() Or KeyHit(28)

'close the video input
StopAnimation video
```

AS

KUVAUS

Kätetään esittämään muuttuja tietyn tyyppikseksi. Osana Function-, Field-, Dim- tai Global-rakenteita.

Perään kirjoitetaan tyyppitunniste, joka voi olla Integer, Float, String, Byte tai Short.

Tämä on ainoa tapa esittää Byte- ja Short-muuttujat.

KÄYTTÖ

```
DIM muuttuja AS tyyppi
```

Katso myös: [DIM](#)

ESIMERKKI

```
Dim enemy_plusx(100) As Float, enemy_plusy(100) As Float
```

```
Dim i As Integer
```

```
Dim angle As Float
```

ASC

KUVAUS

Jokaisella tietokoneen merkillä (oli sitten kirjan, numero tai symboli) on ASCII-koodiksi kutsuttu tunniste. Toisin sanoen, jokaisella merkillä on myös oma numero (0-255). Tämä funktio on päinvastainen toimitus **Chr**:lle. Se palauttaa merkin ASCII-koodin. Esim. **"A"** -> **65**.

Esimerkiksi **GetKey** palauttaa painetun näppäimen ASCII-merkin.

Asc- ja Chr-funktioita käytetään yleisesti yksinkertaiseen tiedon salaukseen.

Katso [ASCII-taulukko](#).

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

ASC (merkki)

- merkki** = Yhden merkin merkkijono. Esim. **"m"**

Katso myös: [CHR](#)

ESIMERKKI

```
AddText "This example shows how to encrypt a string"

'Ask a string
Repeat
    k$=Input("Give a sentence: ")
    DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
ClearKeys

'encrypt
encrypted$=encrypt(k$,"secret")

'decrypt
decrypted$=decrypt(encrypted$,"secret")

AddText "Original:"+k$
AddText "Encrypted:"+encrypted$
AddText "Decrypted:"+decrypted$

Repeat
    DrawScreen
Until EscapeKey()

End

'-----

'This function encrypts a string with a given password
'The longer the password the better the encryption
Function encrypt(stri$,pass$)
    passchar=1
    build$=""

    For i=1 To Len(stri$)
        newvalue=Asc(Mid(stri$,i,1))+Asc(Mid(pass$,passchar,1))
        If newvalue>255 Then newvalue=newvalue-255
        build$=build$+Chr(newvalue)

        passchar=passchar+1
        If passchar>Len(pass) Then passchar=1
    Next i

    Return build$
End Function

'This function decrypts already crypted password
Function decrypt(stri$,pass$)
    passchar=1
    build$=""
```



```
For i=1 To Len(stri$)
    newvalue=Asc(Mid(stri$,i,1)) -Asc(Mid(pass$,passchar,1))
    If newvalue<0 Then newvalue=newvalue+255
    build$=build$+Chr(newvalue)

    passchar=passchar+1
    If passchar>Len(pass) Then passchar=1
Next i

Return build$
End Function
```

ASIN

KUVAUS

Tämä on Sin-funktion päinvastainen laskutoimitus. Eli palautetaan alkuperäinen kulma.

Tulos on desimaaliluku.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

ASIN (luku)

- luku** = Kokonaisluku tai desimaaliluku.

Katso myös: [SIN](#)

ESIMERKKI

ATAN

KUVAUS

Tämä on Tan-funktion päinvastainen laskutoimitus. Eli palautetaan alkuperäinen kulma.

Tulos on desimaaliluku.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

ATAN (luku)

- luku = Kokonaisluku tai desimaaliluku.

Katso myös: [TAN](#)

ESIMERKKI

BEFORE

KUVAUS

Tämä funktio kertoo jäsentä edellisen jäsenen.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

BEFORE (tyyppi_muuttuja)

- tyyppi_muuttuja = Jo valmis tyyplitetty muuttuja, joka toimii osoittimena listaan.

Katso myös: [FIRST](#), [LAST](#), [AFTER](#)

ESIMERKKI

```
Type test
  Field value As Float
End Type

'First create 10 elements
For i=1 To 10
  t.test = New(test)

  t\value=Rnd(1)
Next i

'Print them all from end to start
Repeat

  t.test = Last(test)
  i=10
  While t<> NULL
    Text 10,i,t\value
    i=i+16
    t = Before(t)
  Wend

  DrawScreen

Until EscapeKey()
```

BIN

KUVAUS

Palauttaa lukuarvosta binääri-muotoisen merkkijonon.

Jos et tiedä mikä binäärimuoto on, et tarvitse tätä komentoa :).

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

BIN (luku)

- **luku** = Kokonaisluku.

ESIMERKKI

```
Repeat
  num=Input("Give a number: ")

  Text 0,50,"BIN: "+Bin(num)
  DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
```

BOX

KUVAUS

Piirtää laatikon. Käyttää nykyistä edusväriä (**Color**-käsky).

KÄYTTÖ

BOX x, y, leveys, korkeus, [kiinteä]

- **x** = Kokonaisluku tai desimaaliluku. Alkupisteen vaakakoordinaatti.
- **y** = Kokonaisluku tai desimaaliluku. Alkupisteen pystykoordinaatti.
- **leveys** = Kokonaisluku tai desimaaliluku. Laatikon leveys pikseleinä.
- **korkeus** = Kokonaisluku taid desimaaliluku. Laatikon korkeus pikseleinä.
- **kiinteä** (valinnainen)
 - 0** = ontto
 - 1** = täytetty (oletus)

Katso myös: [DOT](#), [LINE](#), [CIRCLE](#), [ELLIPSE](#)

ESIMERKKI

AddText "Use arrows to move around"

```
'first generate two random sized boxes
box1width=Rand(20,100)
box1height=Rand(20,100)

box2width=Rand(20,100)
box2height=Rand(20,100)

Repeat

    'update controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1

    'box 1 (movable)
    Box x,y,box1width,box1height,OFF

    'box 2 (static)
    Box 260,180,box2width,box2height,OFF

    Text 20,20,"Overlapping: "+BoxOverlap(x,y,box1width,box1height,260,180,box2width,box2height)

    DrawScreen

Until EscapeKey()
```

BOXOVERLAP

KUVAUS

Kertoo ovatko kaksi laatikon muotoista aluetta päällekkäin. Palauttaa nollan, jos ei ole, ja ykkösen, jos ovat.

Funktiota kannattaa käyttää nopeaan matemaattiseen törmäyksentunnistukseen.

KÄYTTÖ

```
BOXOVERLAP (x1, y1, leveys1, korkeus1, x2, y2, korkeus2, leveys2)
```

- **x1** = Desimaaliluku tai kokonaisluku. Laatikon 1 vasemman yläkulman vaakasijainti
- **y1** = Desimaaliluku tai kokonaisluku. Laatikon 1 vasemman yläkulman pystysijainti
- **leveys1** = Desimaaliluku tai kokonaisluku. Laatikon 1 leveys pikseleinä
- **korkeus1** = Desimaaliluku tai kokonaisluku. Laatikon 1 korkeus pikseleinä
- **x2** = Desimaaliluku tai kokonaisluku. Laatikon 2 vasemman yläkulman vaakasijainti
- **y2** = Desimaaliluku tai kokonaisluku. Laatikon 2 vasemman yläkulman pystysijainti
- **leveys2** = Desimaaliluku tai kokonaisluku. Laatikon 2 leveys pikseleinä
- **korkeus2** = Desimaaliluku tai kokonaisluku. Laatikon 2 korkeus pikseleinä

ESIMERKKI

```
AddText "Use arrows to move around"
```

```
'first generate two random sized boxes  
box1width=Rand(20,100)  
box1height=Rand(20,100)
```

```
box2width=Rand(20,100)  
box2height=Rand(20,100)
```

```
Repeat
```

```
  'update controls  
  If LeftKey() Then x=x-1  
  If RightKey() Then x=x+1  
  If UpKey() Then y=y-1  
  If DownKey() Then y=y+1
```

```
  'box 1 (movable)  
  Box x,y,box1width,box1height,OFF
```

```
  'box 2 (static)  
  Box 260,180,box2width,box2height,OFF
```

```
  Text 20,20,"Overlapping: "+BoxOverlap(x,y,box1width,box1height,260,180,box2width,box2height)
```

```
  DrawScreen
```

```
Until EscapeKey()
```

CALLDLL

KUVAUS

Kutsuu CoolBasicin ulkopuolisia DLL-tiedostoja suorittamaan nopeutta vaativia tehtäviä.

Parametrien ja tuloksen välittämiseen käytetään muistipaloja (kts. [Muisti-komennot](#)). Silloin kutsu menisi esimerkiksi näin:

```
SISÄÄN=MakeMEMBlock(...)
ULOS=MakeMEMBlock(...)
'kirjoita parametrit muistipalalle SISÄÄN
CallDLL "dll_nimi.dll","funktion_nimi",SISÄÄN,ULOS
'lue paluuarvo muistipalasta ULOS
```

DLL-tiedostojen avulla CoolBasicin komentotarjontaa voidaan kasvattaa loputtomasti!

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

CALLDLL DLL_nimi, DLL_funktion_nimi, [muistipala_sisään], [muistipala_ulos]

- **DLL_nimi** = Merkkijono. Sisältää kutsuttavan DLL:n nimen. Tiedostopäätteen voi jättää pois, jos kutsut Windowsin omia kirjastoja.
- **DLL_funktion_nimi** = Merkkijono. Kutsuttavan funktion nimi.
- **muistipala_sisään** (*valinnainen*) = Parametrit sisältävä muistipala. Tämän voi jättää pois.
- **muistipala_ulos** (*valinnainen*) = Muistipala, johon kutsuttavan funktion tulos tallentuu. Tämän voi jättää pois.

ESIMERKKI

CallDLL "Media\sample.dll","MyFunction"

CAMERAANGLE

KUVAUS

Kertoo kameran kulman.
Tulos on desimaaliluku.

KÄYTTÖ

CAMERAANGLE ()

Katso myös: [CAMERAX](#), [CAMERAY](#)

ESIMERKKI

```
FrameLimit 40

'Make scene
floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

Repeat

    TurnCamera 5
    MoveCamera 5

    DrawGame
    Text 20,20,CameraAngle()

    DrawScreen

Until EscapeKey()
```

CAMERA FOLLOW

KUVAUS

Asettaa kameran seuraamaan jotakin objektiä automaattisesti! Näin sinun ei tarvitse huolehtia siitä, että kamera pysyy ohjettavan pelihahmon kintereillä tai lukittuna siihen.

Valittavana on useita jäljitystyyppejä (laita seurauksen tyyppiä):

1 (Pehmeä seuraus)

Tämän tyyppinen jäljitys seuraa pelihahmoa pehmeästi. Vastaa *CurveValuen* käyttöä. Mitä suurempi on tarkennearvo, sitä pehmeämmin, mutta hitaammin kamera seuraa.

2 (Rajakartoitin)

Voit määrittää etäisyyden ruudun reunasta pikseleissä, jota lähempänä kamera seuraa mukana. Toisin sanoen: pelihahmo voi liikkua vapaasti ruudun keskellä, mutta jos se yrittää mennä ruudun ulkopuolelle, kamera seuraa perässä. Katso esimerkki #1. Tarkennearvo määrittää kuinka lähelle ruudun reunaa pelaaja voi mennä, pikseleissä.

3 (Näkökenttä)

Tämä vastaa pelihahmon näkökenttää; näet eteen päin suhteellisen kauas, mutta et kunnolla taaksesi tai suoraan sivuille. Katso esimerkki #2. Tarkennearvo määrittää kuinka pitkälle kamera näyttää. Toisin sanoen: efektin voimakkuuden. Kokeile muuttaa tarkennearvoa esimerkissä #2, ja katso kuinka se vaikuttaa peliin.

KÄYTTÖ

CAMERA FOLLOW objekti, seuraustyyppi, tarkennearvo

- **objekti** = Objektimuuttuja, mihin objekti ladattiin tai luotiin.
- **seuraustyyppi**

1 = pehmeä

2 = rajakartoitin

3 = näkökenttä

- **tarkennearvo** = Desimaaliluku, joka riippuu käytettävästä seuraustyyppistä. Katso yllä oleva ohje.

Katso myös: [CLONECAMERA POSITION](#)

ESIMERKKI #1

```
FrameLimit 40 'don't run too fast

grass= MakeObjectFloor()
lawn= LoadImage("Media\grass.bmp")
PaintObject grass, lawn

'load the character
hero= LoadObject("Media\guy.bmp", 72)

'set the camera to follow it (follow style 2 is used)
CameraFollow hero,2,90

AddText "Try to guide the hero outside screen..."

Repeat

    'guide
    If LeftKey() Then TurnObject hero, 5
    If RightKey() Then TurnObject hero, -5

    If UpKey() Then MoveObject hero, 2
    If DownKey() Then MoveObject hero, -2

    DrawScreen

Forever
```

ESIMERKKI #2

```
FrameLimit 40 'don't run too fast

grass= MakeObjectFloor()
lawn= LoadImage("Media\grass.bmp")
PaintObject grass, lawn

'load the character
hero= LoadObject("Media\guy.bmp", 72)
```

```
'set the camera to follow it (follow style 3 is used)
CameraFollow hero,3,100

AddText "Guide and turn"

Repeat
    'guide
    If LeftKey() Then TurnObject hero, 5
    If RightKey() Then TurnObject hero, -5

    If UpKey() Then MoveObject hero, 2
    If DownKey() Then MoveObject hero, -2

    DrawScreen

Forever
```

CAMERAPICK

KUVAUS

Tällä komennolla voit poimia pelikentältä objekteja esimerkiksi hiiren avulla samaan tapaan kuin strategiapeleissä valitaan hiirellä yksikkö, jota hallitaan. Komentoa on tarkoitettu käytettäväksi yhden kerran ennen varsinaista valitun objektin tutkiskelua. Poimitun objektin saat selville funktiolla **PickedObject**.

Huomaa myös, että poimittavat objektit pitää esitellä "poimittaviksi" ennen kuin CoolBasic tunnistaa niitä. Tämä tapahtuu komennolla **ObjectPickable**. Tämä mahdollistaa sen, että vain haluttuja objekteja voi poimia.

Koordinaatit esitellään ruudun mukaan (ei pelimaailman mukaan), eli ruudun vasen yläkulma on (0,0). Yleensä koordinaatteina toimii hiirikursori.

KÄYTTÖ

CAMERAPICK x, y

- **x** = Ruudun x-koordinaatti.
- **y** = Ruudun y-koordinaatti.

Katso myös: [OBJECTPICKABLE](#), [PICKEDOBJECT](#), [OBJECTPICK](#)

ESIMERKKI

```
FrameLimit 40

'Enable graphics commands to cast shapes to
'the game world
DrawToWorld ON

pacman1=LoadObject("Media\pacman.bmp",360)
pacman2=CloneObject(pacman1)
pacman3=CloneObject(pacman1)

ObjectPickable pacman1,1 'box-pick
ObjectPickable pacman2,2 'circle-pick
ObjectPickable pacman3,3 'pixel-perfect

PositionObject pacman1, -100,0
PositionObject pacman3, 100,0

'Alter mouse pointer
cursor=LoadImage("Media\crosshair.bmp")
ShowMouse cursor

Color cbblack

Locate 70,80
AddText "box-pick"
Locate 160,80
AddText "circle-pick"
Locate 260,80
AddText "pixel-perfect"

Color cbblack
ClsColor cbwhite

Repeat

    'Make them spin, add some excitement
    TurnObject pacman1,1
    TurnObject pacman2,1
    TurnObject pacman3,1

    'Pick
    CameraPick MouseX(),MouseY()

    DrawGame

    'Something WAS picked, let's see what it is
    e=PickedObject()
    Color cbblack
    If e Then
```

```
Color cbred
If e=pacman1 Then
    Text 0,0,"pacman on the left"
ElseIf e=pacman2 Then
    Text 0,0,"pacman at the center"
ElseIf e=pacman3 Then
    Text 0,0,"pacman on the right"
EndIf
EndIf

'draw the check-zones
Box ObjectX(pacman1) -32,ObjectY(pacman1)+32,64,64,OFF
Circle ObjectX(pacman2) -32,ObjectY(pacman2)+32,64,OFF

DrawScreen

Until EscapeKey()
```

CAMERAX

KUVAUS

Kertoo kameran X-sijainnin. Koordinaatti palautetaan desimaalilukuna, ja se on maailmakoordinaatti-muodossa.

KÄYTTÖ

CAMERAX ()

Katso myös: [CAMERAY](#), [CAMERAANGLE](#)

ESIMERKKI

```
FrameLimit 40

'Make scene
floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

guy=LoadObject("Media\guy.bmp")

AddText "Press SPACE for default camera position"

Repeat

    'move camera as absolute
    TranslateCamera 1,1

    'return to origin?
    If KeyDown(cbkeyspace) Then PositionCamera 0,0

    'Print the camera coordinates
    DrawGame
    Text 20,20,CameraX()
    Text 20,40,CameraY()

    DrawScreen

Until EscapeKey()
```

CAMERAY

KUVAUS

Kertoo kameran Y-sijainnin. Koordinaatti palautetaan desimaalilukuna, ja se on maailmakoordinaatti-muodossa.

KÄYTTÖ

CAMERAY ()

Katso myös: [CAMERAX](#), [CAMERAANGLE](#)

ESIMERKKI

```
FrameLimit 40

'Make scene
floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

guy=LoadObject("Media\guy.bmp")

AddText "Press SPACE for default camera position"

Repeat

    'move camera as absolute
    TranslateCamera 1,1

    'return to origin?
    If KeyDown(cbkeyspace) Then PositionCamera 0,0

    'Print the camera coordinates
    DrawGame
    Text 20,20,CameraX()
    Text 20,40,CameraY()

    DrawScreen

Until EscapeKey()
```

CENTERTEXT

KUVAUS

Kirjoittaa tekstiä ruudulle nykyisellä fontilla. Teksti tasataan siten, että sen vasen yläkulma on annetuissa koordinaateissa.

Ero **Text**-komenttoon on se, että nyt voit määrittää millä tavalla teksti keskitetään ruudulle. Voit keskittää sen vaaka- tai pystysuunnassa. Tai molemmissa.

Tulostettu teksti on taustaltaan läpinäkyvää.

Joillakin laitteistoilla on havaittu hidastumista, kun CenterText-komentoa käytetään. Jos haluat tulostaa sellaista tekstiä, joka ei muutu usein, käytä mieluummin AddTextiä. Ongelman on todettu häviävän, kun hakee näytönohjaimeensa uudet ajurit.

KÄYTTÖ

```
CENTERTEXT x, y, teksti, [tyyli]
```

- **x** = Tekstin sijainti vaakasuunnassa. Kokonaisluku.
- **y** = Tekstin sijainti pystysuunnassa. Kokonaisluku.
- **teksti** = Tulostettava merkkijono.
- **tyyli** (valinnainen)
 - 0** = keskitys vain vaakasuunnassa (oletus)
 - 1** = keskitys vain pystysuunnassa
 - 2** = täydellinen keskitys

Katso myös: [TEXT](#), [ADDDTEXT](#)

ESIMERKKI

```
arial=LoadFont("arial",20)
impact=LoadFont("impact",28)
system=LoadFont("fixedsys")
coolb=LoadFont("coolb")
verdana=LoadFont("verdana",18,ON,OFF,ON) 'bold and underlined
times=LoadFont("times new roman",20,OFF,ON) 'italic
courier=LoadFont("courier new",40)
```

Repeat

```
Color cborange
SetFont arial
Text 40,20,"I'll look into it"

Color cblightgreen
SetFont impact
Text 100,100,"A sound plan!"

Color cbyellow
SetFont system
Text 260,270,"Dead man walking"

Color cbcyan
SetFont system
Text 20,200,"This..is..my BOOMSTICK!"

Color cbsilver
SetFont verdana
Text 70,220,"What do you want?!"

Color cblightred
SetFont times
Text 150,180,"Night beckons"

Color cbgreen
SetFont courier
CenterText 200,130,"Don't worry, be happy!"

DrawScreen
```

```
Until EscapeKey()
```


CHDIR

KUVAUS

Vaihtaa pelin hakemistoa, eli sitä paikkaa, josta kuvat, äänet ja muu media oletuksellisesti avataan.

Nykyisen työkansion saat selville funktiolla **CurrentDir**.

Kun siirryt aseman juureen, muista liittää kenoviiva polun loppuun:

- Toimii: CHDIR "C:\"
- Ei toimi: CHDIR "C:"

Jos haluat siirtyä yläkansioon (hakemistopuussa alaspäin), komenna **ChDir** "..".

KÄYTTÖ

CHDIR polku

- **polku** = Merkkijono, joka määrittää uuden hakemiston. Käytä kahta pistettä ("..") siirtyäksesi yläkansioon.

Katso myös: [CURRENTDIR](#)

ESIMERKKI

```
AddText "Current directory is: "  
AddText CurrentDir()  
  
AddText "Parent directory is: "  
ChDir ".."  
AddText CurrentDir()  
  
AddText "Changing to C:"  
ChDir "C:\"  
AddText CurrentDir()  
  
DrawScreen  
  
WaitKey
```

CHR

KUVAUS

Jokaisella tietokoneen merkistön merkillä (oli sitten kirjan, numero tai symboli) on ASCII-koodiksi kutsuttu tunniste. Toisin sanoen, jokaisella merkillä on myös oma numero (0-255). Tämä funktio luo sille annetun ASCII-koodin perusteella merkin. Esimerkiksi Chr(65) on "A", koska kyseisen kirjaimen numero ASCII-taulukossa on 65.

Esimerkiksi **GetKey** palauttaa painetun näppäimen ASCII-merkin.

Katso [ASCII-taulukko](#).

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

CHR (koodi)

- koodi** = Merkin ASCII-koodi. Kokonaisluku 0-255.

Katso myös: [ASC](#)

ESIMERKKI

```
'Because the listing won't fit to the screen
'we need to switch to bigger screen size
SCREEN 640,480,16,cbwindowed

x=1
y=1

'iterate all values between 0 and 255
For i=0 To 255

    'Print the ASCII code
    Text x*8,y*13,i

    'Print the correponding character
    Text x*8+28,y*13,Chr(i)

    y=y+1
    If y>35 Then x=x+10:y=1

Next i

DrawScreen

WaitKey
```

CIRCLE

KUVAUS

Piirtää ympyrän. Käyttää nykyistä edusväriä (**Color**-käsky).

KÄYTTÖ

CIRCLE x, y, säde, [kiinteä]

- **x** = Kokonaisluku tai desimaaliluku. Alkupisteen vaakakoordinaatti.
- **y** = Kokonaisluku tai desimaaliluku. Alkupisteen pystykoordinaatti.
- **koko** = Kokonaisluku tai desimaaliluku. Ympyrän halkaisija pikseleinä.
- **kiinteä** (valinnainen)
 - 0** = ontto
 - 1** = täytetty (oletus)

Katso myös: [DOT](#), [LINE](#), [BOX](#), [ELLIPSE](#)

ESIMERKKI

AddText "Use arrows to move around"

'first generate two random sized circles

radius1=Rand(20,60)

radius2=Rand(20,60)

Repeat

'update controls

If LeftKey() Then x=x-1

If RightKey() Then x=x+1

If UpKey() Then y=y-1

If DownKey() Then y=y+1

'circle 1 (movable)

Circle x-radius1,y-radius1,radius1*2,OFF

'circle 2 (static)

Circle 260-radius2,180-radius2,radius2*2,OFF

Text 20,20,"Overlapping: "+(Distance(x,y,260,180)<(radius1+radius2))

DrawScreen

Until EscapeKey()

CLEARARRAY

KUVAUS

Sallii tai estää taulukon tyhjennyksen uudelleenvenytyksen aikana. Toiminto on oletuksellisesti päällä.

Katso myös: [REDIM](#)

KÄYTTÖ

CLEARARRAY **päällä**

- **päällä**
ON = nollaa taulukoiden sisältö (oletus)
OFF = säilytä taulukoiden sisältö

ESIMERKKI

```
Dim array(8)

ClearArray OFF

For i=1 To 8
    array(i)=i
Next i

'try to comment this
Redim array(12)

For i=1 To 12
    Print array(i)
Next i

WaitKey
```

CLEARCOLLISIONS

KUVAUS

Joka kerta kun määrittelet uuden törmäyksen SetupCollisionilla, CoolBasic lisää siitä tiedon sisäiseen listaansa. Tämä komento tyhjentää tuon listan, ja samalla tuhoaa kaikki törmäykset.

KÄYTTÖ

CLEARCOLLISIONS

ESIMERKKI

```
FrameLimit 40

'load map
map=LoadMap("Media\testmap.til", "Media\negative.bmp")

'load player
guy=LoadObject("Media\guy.bmp", 72)
PositionObject guy, 0, -40

'load a neutral unit
fellow=LoadObject("Media\guy.bmp", 72)
PositionObject fellow, -100, -40

'setup some collisions
SetupCollision guy, fellow, 2, 2, 2
SetupCollision guy, map, 1, 4, 2

'Delete all collisions (which really doesn't
'make any sense since we just defined some).
ClearCollisions

Repeat

    'Controls
    If LeftKey() Then TurnObject guy, 5
    If RightKey() Then TurnObject guy, -5
    If UpKey() Then MoveObject guy, 3
    If DownKey() Then MoveObject guy, -3

    'camera follow
    UpdateGame
    CloneCameraPosition guy

    DrawScreen

Until EscapeKey()
```

CLEARKEYS

KUVAUS

Aina, kun käyttäjä painaa näppäimistöltä jotakin, tieto siitä menee ns. näppäimistöpuskuriin, josta tietokoneohjelmat niitä lukevat. Jos esimerkiksi painat 3 näppäintä yhtä aikaa, niistä muodostuu jono näppäimistöpuskuriin.

Tämä komento tyhjentää näppäimistöpuskurin, mikä tarkoittaa sitä, että se nollautuu. Komennon ideaalinen käyttötarkoitus on alustaa näppäimistö latausruudun jälkeen.

KÄYTTÖ

CLEARKEYS

Katso myös: [CLEARMOUSE](#)

ESIMERKKI

CLEARMOUSE

KUVAUS

Kannattaa lukea ensin **ClearKeys**-komennon ohje. Tämä on vähän sama, mutta hiirelle.

Paras käyttötarkoitus lienee juuri ennen **WaitMousea**, koska silloin edelliset syötteet eivät suoraan jatka ohjelman suoritusta.

KÄYTTÖ

CLEARMOUSE

Katso myös: [CLEARKEYS](#)

ESIMERKKI

Objektit : **komennot**

CLEAROBJECTS

KUVAUS

Poistaa kaikki objektit. Tämä on kätevää esimerkiksi kun pitää vaihtaa kenttää tai muuten "nollata tilanne".

KÄYTTÖ

CLEAROBJECTS

Katso myös: [DELETEOBJECT](#)

ESIMERKKI

```
FrameLimit 40

'load 100 objects
master=LoadObject("Media\cow.bmp",72)
For i=1 To 100
    temp=CloneObject(master) 'prefer to clone.. it's faster
    PositionObject temp,Rand( -200,200),Rand( -150,150)
Next i

Color cbblack
AddText "Press SPACE to clear all objects..."
ClsColor cbpink

Repeat

    If KeyHit(cbkeyspace) Then
        ClearObjects
    EndIf

    DrawScreen

Until EscapeKey()
```


CLEARTEXT

KUVAUS

Poistaa kaikki **AddText**-komennolla lisätyt tekstit.

KÄYTTÖ

CLEARTEXT

Katso myös: [ADDEXT](#)

ESIMERKKI

```
FrameLimit 40

Locate 30,30

AddText "This text is sticky and won't go away"
AddText "unless you use the Cleartext command."
AddText "Thank god, this demo features also that"
AddText "command."
AddText ""

Color 255,0,0
AddText "Press SPACE to clear the text."

plusx=2
plusy=2

Repeat

    'update the game itself
    x=x+plusx
    y=y+plusy

    If x<0 Or x>400 Then plusx= -plusx
    If y<0 Or y>300 Then plusy= -plusy

    Color Rand(255),Rand(255),Rand(255)
    Circle x-20,y-20,40

    'clear the text?
    If KeyHit(cbkeyspace) Then ClearText

DrawScreen

Until EscapeKey()
```

CLONECAMERAORIENTATION

KUVAUS

Asettaa kameran samaan kulmaan jonkun objektin kanssa. Voit esimerkiksi haluta yhdellä komennolla kääntää kameran esimerkiksi samaan suuntaan, mihin pelihahmo osoittaa ja sitten siirtää kameraa eteenpäin "zoomaten".

KÄYTTÖ

CLONECAMERAORIENTATION objekti

- **objekti** = Objektimuuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [CLONECAMERAPOSITION](#)

ESIMERKKI

```
FrameLimit 40

'Make an infinite lawn the guy can walk on
floor= MakeObjectFloor()

'Texture the EMPTY floor as grass
lawn= LoadImage("Media\grass.bmp")
PaintObject floor,lawn

guy= LoadObject("Media\guy.bmp",75)

AddText "Use arrows to move..."
AddText "Press RETURN to 'zoom'"

Repeat

    'Update controls
    If LeftKey() Then TurnObject guy,5
    If RightKey() Then TurnObject guy, -5
    If UpKey() Then MoveObject guy,2
    If DownKey() Then MoveObject guy, -2

    'Glue camera to object position
    CloneCameraPosition guy

    'The 'zooming' system
    If KeyDown(cbkeyreturn) Then
        zoom#=CurveValue(100,zoom#,20.0)
    Else
        zoom#=CurveValue(0,zoom#,20.0)
    EndIf

    CloneCameraOrientation guy
    MoveCamera zoom#

    DrawScreen

Until EscapeKey()
```

CLONECAMERAPOSITION

KUVAUS

Asettaa kameran samaan paikkaan kuin jokin objekti. Käytä tätä komentoa "liimaamaan" kamera seuraamaan haluttua objektia. Tällöin sinun on muistettava kutsua tätä komentoa **joka kerta** pelin pääsil mukaan sisällä. Voit myös käyttää komentoa kerran keskittämään näkymän johonkin tiettyyn objektiin.

Jos seurattavassa objektissa on törmäystunnistus, kamera voi nykäistä seinää päin kävellessä. Estä tämä komentamalla **UpdateGame**a tätä ennen.

KÄYTTÖ

CLONECAMERAPOSITION objekti

- **objekti** = Objektimuuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [CLONECAMERAORIENTATION](#)

ESIMERKKI

```
FrameLimit 40

'Enable graphics commands to cast shapes to
'the game world
DrawToWorld ON

'Make an infinite lawn the guy can walk on
floor= MakeObjectFloor()

'Texture the EMPTY floor as grass
lawn= LoadImage("Media\grass.bmp")
PaintObject floor,lawn

guy= LoadObject("Media\guy.bmp",75)

AddText "Use arrows to move..."

Repeat

    'Update controls
    If LeftKey() Then TurnObject guy,5
    If RightKey() Then TurnObject guy, -5
    If UpKey() Then MoveObject guy,2
    If DownKey() Then MoveObject guy, -2

    'Glue camera to object position
    CloneCameraPosition guy

    'Draw a circle after objects
    'Visualize the center of the world
    DrawGame
    Circle-15,15,30,OFF

    DrawScreen

Until EscapeKey()
```

CLONEIMAGE

KUVAUS

Kopioi jo olemassa olevan kuvan. Kopiointi on erittäin nopeaa - paljon nopeampaa kuin ladata uusi kuva levyltä. Tämä on erityisen kätevää silloin kun haluat muuttaa jotain kuvaa, mutta haluat pitää alkuperäisen koskemattomana.

Funktio palauttaa nollan, jos kuvaa ei voitu luoda.

KÄYTTÖ

CLONEIMAGE (kuvamuuttuja)

- **kuvamuuttuja** = Kuva, joka kopioidaan.

Katso myös: [CLONEOBJECT](#)

ESIMERKKI

```
cow= LoadImage ("Media\cow.bmp")

'make a clone
copy= CloneImage (cow)

ClsColor cbwhite
Color cbblack

Repeat

    'draw both

    CenterText 75,30,"Original"
    DrawImage cow,60,60

    CenterText 275,30,"Clone"
    DrawImage copy,260,60

    'don't forget this!
    DrawScreen

Until EscapeKey()
```

CLONEOBJECT

KUVAUS

Kloonaa jo olemassa olevan objektin, ja sijoittaa sen pelimaailman keskelle. Vaikka kopioitava objekti olisi piilotettu, kloonin ilmestyy näkyviin. Kulma ja sijainti nollataan. Kloonit perivät kaikki muut ominaisuudet isännältään.

Kloonaaminen on erittäin nopeaa. Siksi usein tarvittu objektit kannattaa ennemmin kloonata kuin ladata uudestaan levyiltä. Kloonatut objektit eivät myöskään kuluta tietokoneen muistia. Kloonaaminen on hyödyllistä esimerkiksi suurien vihollismäärien tai ammuksien hallintaan.

Kloonatut objektit ovat riippuvaisia isännästään. Jos isäntä poistetaan, kloonit muuttuvat epäkelvillisiksi, ja kaatavat ohjelman. Siksi alkuperäinen objekti kannattaa ennemminkin piilottaa, eikä siihen kosketa sen jälkeen. Kaikki muut ovat klooneja. Tämä ns. **Master-objektin** olemus koskee kaikkia siitä tehtyjä klooneja. Muuttamalla master-objektin ulkonäköä myös kaikki kloonit muuttuvat. Oheinen esimerkki voi olla pitkä, mutta se näyttää mihin CloneObject soveltuu parhaiten.

One object to rule them all

Just don't kill the Master

KÄYTTÖ

CLONEOBJECT (objektimuuttuja)

- objektimuuttuja** = Se muuttuja, joka sisältää kloonattavan objektin.

Katso myös: [LOADOBJECT](#)

ESIMERKKI

```
Type BULLETS
  Field obj
End Type

' _ _ _ _ _
'MAIN PROGRAM
' _ _ _ _ _

FrameLimit 40

'load the soldier, turning detail 72
guy=LoadObject("Media\soldier.bmp",72)

'load the bullet (master-object) and hide it
bullet=LoadObject("Media\bullet.bmp")
ShowObject bullet,OFF

ClsColor cbdark

AddText "Arrows to turn, SPACE to shoot"

'MAIN LOOP
Repeat

  'Update turning
  If LeftKey() Then TurnObject guy,5
  If RightKey() Then TurnObject guy, -5

  'Shoot?
  If KeyDown(cbkeyspace) And reload=0

    newBullet.BULLETS= New(BULLETS)
    newBullet\obj=CloneObject(bullet)
    CloneObjectPosition newBullet\obj,guy
    CloneObjectOrientation newBullet\obj,guy
    'put the bullet at the end of the barrel
    MoveObject newBullet\obj,24

    reload=4
  EndIf

  'Update all BULLETS (move them forward)
  i=0
  For iBullet.BULLETS= Each BULLETS

    MoveObject iBullet\obj,6 'speed=6
```

```
'delete bullet from collection if it
'gets too far away
If Distance2(iBullet\obj,guy)>200 Then
    DeleteObject iBullet\obj
    Delete iBullet
EndIf

i=i+1

Next iBullet

'Update reloading
If reload>0 Then reload=reload-1

Text 30,30,"Bullets to update: "+i

'Don't forget this :)
DrawScreen

Until EscapeKey()
```

CLONEOBJECTORIENTATION

KUVAUS

Asettaa jonkun objektin täsmälleen samaan kulmaan toisen objektin kanssa. Esimerkiksi kun uusi ammus luodaan, se sijoitetaan samaan paikkaan pelaajan kanssa, ja siirretään hieman eteenpäin niin, että se lähtisi aseeseen piipusta.

KÄYTTÖ

CLONEOBJECTORIENTATION objektimuuttuja, kohde_objekti

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **kohde_objekti** = Se objekti, jonka kulma monistetaan.

Katso myös: [CLONEOBJECTPOSITION](#)

ESIMERKKI

```
FrameLimit 40 'limit game speed

fireball1=LoadObject("Media\fireball.bmp",18)
fireball2=LoadObject("Media\fireball.bmp",18)
fireball3=LoadObject("Media\fireball.bmp",18)
fireball4=LoadObject("Media\fireball.bmp",18)

car=LoadObject("Media\car.bmp",72)

Color cbblue
AddText "Arrows to turn"

ClsColor cbdarkblue

Repeat
    'Update controls
    If LeftKey() Then TurnObject car,5
    If RightKey() Then TurnObject car, -5
    If UpKey() Then MoveObject car,2
    If DownKey() Then MoveObject car, -2

    'Move all fireballs to the same position as the car
    CloneObjectPosition fireball1,car
    CloneObjectPosition fireball2,car
    CloneObjectPosition fireball3,car
    CloneObjectPosition fireball4,car

    'Rotate all fireballs to the same angle as the car
    CloneObjectOrientation fireball1,car
    CloneObjectOrientation fireball2,car
    CloneObjectOrientation fireball3,car
    CloneObjectOrientation fireball4,car

    'Move them all to their own places
    MoveObject fireball1,34, -20
    MoveObject fireball2,34,20
    MoveObject fireball3, -34, -20
    MoveObject fireball4, -34,20

    'Update angle
    angle=WrapAngle(angle+20)

    'Rotate all fireballs
    RotateObject fireball1,angle
    RotateObject fireball2,angle
    RotateObject fireball3,angle
    RotateObject fireball4,angle

    DrawScreen

Until EscapeKey()
```

CLONEOBJECTPOSITION

KUVAUS

Asettaa jonkun objektin täsmälleen samaan paikkaan toisen objektin kanssa. Esimerkiksi kun uusi ammus luodaan, se sijoitetaan samaan paikkaan pelaajan kanssa, ja siirretään hieman eteenpäin niin, että se lähtisi aseeseen piipusta.

KÄYTTÖ

CLONEOBJECTPOSITION objektimuuttuja, kohde_objekti

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin
- **kohde_objekti** = Se objekti, jonka sijaintiin objekti siirretään

Katso myös: [CLONEOBJECTORIENTATION](#), [CLONECAMERAPOSITION](#)

ESIMERKKI

```
FrameLimit 40 'limit game speed

fireball1=LoadObject("Media\fireball.bmp",18)
fireball2=LoadObject("Media\fireball.bmp",18)
fireball3=LoadObject("Media\fireball.bmp",18)
fireball4=LoadObject("Media\fireball.bmp",18)

car=LoadObject("Media\car.bmp",72)

Color cbblue
AddText "Arrows to turn"

ClsColor cbdarkblue

Repeat

    'Update controls
    If LeftKey() Then TurnObject car,5
    If RightKey() Then TurnObject car, -5
    If UpKey() Then MoveObject car,2
    If DownKey() Then MoveObject car, -2

    'Move all fireballs to the same position as the car
    CloneObjectPosition fireball1,car
    CloneObjectPosition fireball2,car
    CloneObjectPosition fireball3,car
    CloneObjectPosition fireball4,car

    'Rotate all fireballs to the same angle as the car
    CloneObjectOrientation fireball1,car
    CloneObjectOrientation fireball2,car
    CloneObjectOrientation fireball3,car
    CloneObjectOrientation fireball4,car

    'Move them all to their own places
    MoveObject fireball1,34, -20
    MoveObject fireball2,34,20
    MoveObject fireball3, -34, -20
    MoveObject fireball4, -34,20

    'Update angle
    angle=WrapAngle(angle+20)

    'Rotate all fireballs
    RotateObject fireball1,angle
    RotateObject fireball2,angle
    RotateObject fireball3,angle
    RotateObject fireball4,angle

    DrawScreen

Until EscapeKey()
```


CLOSEFILE

KUVAUS

Käytä tätä komentoa sulkeaksesi ennestään avoinna olevan tiedoston.

Sinun tulisi sulkea tiedostot heti, kun et enää lue niitä.

KÄYTTÖ

CLOSEFILE tiedosto

- **tiedosto** = Tiedostomuuttuja.

Katso myös: [OPENTOREAD](#), [OPENTOWRITE](#), [OPENTOEDIT](#)

ESIMERKKI

```
'make window bigger as otherwise the text won't fit
SCREEN 580,420,16,cwindowed

'open the file
f=OpenToRead("Media\example.txt")

textline=0

'print the whole file to the screen
While Not EOF(f)

    Text 0,textline,ReadLine(f)

    textline=textline+12

Wend

'show the screen
DrawScreen

'don't forget this!
CloseFile f

WaitKey
```

CLOSEINPUT

KUVAUS

Kun käyttäjä on käynnistänyt konsolisyötön **Input**-funktiolla, se on suljettava käytön jälkeen (kun Enteriä painetaan). Muutoin sen sisältämä teksti ei nollaudu seuraavaa kertaa varten.

Kokeile esimerkkiä CloseInputin kanssa ja ilman.

KÄYTTÖ

CLOSEINPUT

Katso myös: [INPUT](#)

ESIMERKKI

```
'first input

AddText "This is the 1st input"
AddText "Write someting and hit RETURN"

Repeat
    m$=Input(">>")
    Text 10,50,m$
    DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput 'try to comment/uncomment this

ClearKeys
ClearText

'second input

AddText "This is the 2nd input"
AddText "Write someting and hit RETURN"

Repeat
    m$=Input(">>","*")
    Text 10,50,m$
    DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
```

CLS

KUVAUS

Tyhjentää ruudun (tai kuvan, jos piirtäminen on asetettu kuvaan) nykyiseksi taustaväriksi.

Taustaväriä muutetaan **ClsColor**-käskyllä.

KÄYTTÖ

CLS

Katso myös: [CLSCOLOR](#)

ESIMERKKI

```
'First create an image
img= MakeImage (256,256)
DrawToImage img
    ClsColor cbyellow
    Cls 'Clear background to yellow

    Color cbred
    Text 40,40,"This is a picture"
DrawToScreen

ClsColor cbwhite
Color cbblack

'Draw all, main loop
Repeat

    DrawImage img,20,20

    Text 300,150,"SCREEN"

    DrawScreen

Until EscapeKey()
```

CLSCOLOR

KUVAUS

Asettaa ruudun taustavärin. Väri annetaan kolmena komponenttina RGB-standardin mukaisesti. Lisätietoa väreistä oppitunnista 6 ja sanakirjasta. Katso esimerkki.

Jos et halua määrittää väriä kolmessa komponentissa, voit myös jättää kaksi ensimmäistä parametria nolaksi ja antaa miinusmerkkisen arvon 1-16.7 milj.

Voit käyttää väriarvojen tilalla myös CoolBasicin sisäänrakennettua värivakiota. Siten **Color cbRed** on sama kuin **Color 255,0,0**. Listan värivakioista löydät [täältä](#).

KÄYTTÖ

```
CLSCOLOR punainen, vihreä, sininen
```

```
    tai
```

```
CLSCOLOR 0, 0, pikseliarvo
```

```
    tai
```

```
CLSCOLOR värivakio
```

- **punainen** = Kokonaisluku. Punainen komponentti.
- **vihreä** = Kokonaisluku. Vihreä komponentti.
- **sininen** = Kokonaisluku. Sininen komponentti.
- **pikseliarvo** = Kokonaisluku. Muuttujaan luettu 32-bittinen pikseli tai kokonaisluku -16,7 milj -> 0.
- **värivakio** = Joku CoolBasicin [värivakioista](#).

Katso myös: [COLOR](#)

ESIMERKKI

```
Repeat
```

```
    'Set new background colour
```

```
    ClsColor w,w,w
```

```
    'Some normal graphics and text
```

```
    Color 255,0,0
```

```
    Box 100,100,200,100
```

```
    Color 255,255,0
```

```
    Text 0,0,"Going Down... Again"
```

```
    'Update fading
```

```
    w=w+10
```

```
    If w>255 Then w=0
```

```
    DrawScreen
```

```
Forever
```

COLLISIONANGLE

KUVAUS

Palauttaa tietyn törmäyksen kulman. Pääset jokaiseen törmäykseen erikseen käsiksi. (1 -> **CountCollisions**).

Ihanteellinen käyttötarkoitus on selvittää missä kulmassa kranaatin tulee kimmota seinästä. Palautettu kulma on kohtisuoraan pois päin seinästä.

Palautettu luku on desimaaliluku.

KÄYTTÖ

COLLISIONANGLE (objektimuuttuja, indeksi)

- objektimuuttuja** = Objekti, jonka törmäykset tarkistetaan.
- indeksi** = Kokonaisluku ykkösestä eteenpäin. Törmäyksen järjestysnumero.

Katso myös: [PICKEDANGLE](#), [COLLISIONX](#), [COLLISIONY](#), [COUNTCOLLISIONS](#), [GETCOLLISION](#)

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

'Load the map
map=LoadMap("Media\cdm2.til","Media\tileset.bmp")

'Load player
soldier=LoadObject("Media\soldier.bmp",72)
ObjectRange soldier,32

'Load "bullet"
bullet=LoadObject("Media\fireball.bmp",9)
ShowObject bullet,OFF

'setup some collisions
SetupCollision soldier,map,1,4,2
SetupCollision bullet,map,2,4,1

AddText "Arrows to move, SPACE to shoot"

Repeat

  'Update controls
  If LeftKey() Then TurnObject soldier,5
  If RightKey() Then TurnObject soldier, -5
  If UpKey() Then MoveObject soldier,3.5
  If DownKey() Then MoveObject soldier, -2

  'shooting
  If KeyDown(cbkeyspace) And alive=0 Then
    ShowObject bullet,ON 'create bullet again
    ResetObjectCollision bullet
    CloneObjectPosition bullet,soldier
    CloneObjectOrientation bullet,soldier
    bulletplusx#=Cos(ObjectAngle(bullet))*6
    bulletplusy#=Sin(ObjectAngle(bullet))*6
    alive=120
  EndIf

  'update bullet if alive
  If alive Then
    TranslateObject bullet,bulletplusx#,bulletplusy#
    TurnObject bullet,40
    If CountCollisions(bullet) Then
      angle=CollisionAngle(bullet,1)
      'mirror movement components
      If angle=0 Or angle=180 Then
        bulletplusx#= -bulletplusx#
      EndIf
      If angle=90 Or angle=270 Then
        bulletplusy#= -bulletplusy#
      EndIf
      TranslateObject bullet,bulletplusx#,bulletplusy#
    EndIf
```

```
'update bullet life state  
alive=alive-1  
If alive=0 Then 'death  
    ShowObject bullet,OFF  
EndIf  
EndIf
```

```
'Camera follow  
CloneCameraPosition soldier
```

```
DrawScreen
```

```
Until EscapeKey()
```

COLLISIONX

KUVAUS

Palauttaa tietyn törmäyksen paikan. Tämä funktio kertoo X-koordinaatin. Pääset jokaiseen törmäykseen erikseen käsiksi. (1 -> **CountCollisions**).

Funktio on omiaan kertomaan esimerkiksi tarkan sijainnin, missä raketti räjähti seinään tai esimerkiksi huomioimaan kuolemaan johtavia check-zoneja.

Palautettu luku on kokonaisluku. Katso myös **CollisionY**.

KÄYTTÖ

COLLISIONX (objektimuuttuja, indeksi)

- **objektimuuttuja** = Objekti, jonka törmäykset tarkistetaan.
- **indeksi** = Kokonaisluku ykkösestä eteenpäin. Törmäyksen järjestysnumero.

Katso myös: [COLLISIONY](#), [COUNTCOLLISIONS](#), [COLLISIONANGLE](#), [GETCOLLISION](#)

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

'Load the map
map=LoadMap("Media\cdm2.til","Media\tileset.bmp")

'Load player
soldier=LoadObject("Media\soldier.bmp",72)
ObjectRange soldier,32

'Load dumb immobile bot
bot=LoadObject("Media\soldier.bmp")
ObjectRange bot,32
PositionObject bot,50,200

'setup some collisions
SetupCollision soldier,bot,2,2,2
SetupCollision soldier,map,1,4,2

Repeat

    'Update controls
    If LeftKey() Then TurnObject soldier,5
    If RightKey() Then TurnObject soldier, -5
    If UpKey() Then MoveObject soldier,3.5
    If DownKey() Then MoveObject soldier, -2

    'Camera follow
    CloneCameraPosition soldier

    'Enable additional graphics
    DrawGame

    collisions=CountCollisions(soldier)

    'iterate through all collisions
    For i=1 To collisions
        Color cbwhite
        Text 0,(i-1)*12,"angle: "+CollisionAngle(soldier,i)+" object: "+GetCollision(soldier,i)

        'draw the collision-ball
        Color cbred
        Circle CollisionX(soldier,i) -5,CollisionY(soldier,i)+5,10
    Next i

    'render SCREEN
    DrawScreen

Until EscapeKey()
```

COLLISIONY

KUVAUS

Palauttaa tietyn törmäyksen paikan. Tämä funktio kertoo Y-koordinaatin. Pääset jokaiseen törmäykseen erikseen käsiksi. (1 -> **CountCollisions**).

Funktio on omiaan kertomaan esimerkiksi tarkan sijainnin, missä raketti räjähti seinään tai esimerkiksi huomioimaan kuolemaan johtavia check-zoneja.

Palautettu luku on kokonaisluku. Katso myös **CollisionX**.

KÄYTTÖ

COLLISIONY (objektimuuttuja, indeksi)

- **objektimuuttuja** = Objekti, jonka törmäykset tarkistetaan.
- **indeksi** = Kokonaisluku ykkösestä eteenpäin. Törmäyksen järjestysnumero.

Katso myös: [COLLISIONX](#), [COUNTCOLLISIONS](#), [COLLISIONANGLE](#), [GETCOLLISION](#)

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

'Load the map
map=LoadMap("Media\cdm2.til", "Media\tileset.bmp")

'Load player
soldier=LoadObject("Media\soldier.bmp",72)
ObjectRange soldier,32

'Load dumb immobile bot
bot=LoadObject("Media\soldier.bmp")
ObjectRange bot,32
PositionObject bot,50,200

'setup some collisions
SetupCollision soldier,bot,2,2,2
SetupCollision soldier,map,1,4,2

Repeat

    'Update controls
    If LeftKey() Then TurnObject soldier,5
    If RightKey() Then TurnObject soldier, -5
    If UpKey() Then MoveObject soldier,3.5
    If DownKey() Then MoveObject soldier, -2

    'Camera follow
    CloneCameraPosition soldier

    'Enable additional graphics
    DrawGame

    collisions=CountCollisions(soldier)

    'iterate through all collisions
    For i=1 To collisions
        Color cbwhite
        Text 0,(i-1)*12,"angle: "+CollisionAngle(soldier,i)+" object: "+GetCollision(soldier,i)

        'draw the collision-ball
        Color cbred
        Circle CollisionX(soldier,i) -5,CollisionY(soldier,i)+5,10
    Next i

    'render SCREEN
    DrawScreen

Until EscapeKey()
```


COLOR

KUVAUS

Asettaa piirtovärin. Text, Print, Box, Dot, Line, Ellipse ja Circle käyttävät kaikki nykyistä piirtoväriä. Väri annetaan kolmena komponenttina RGB-standardin mukaisesti. Lisätietoa väreistä oppitunnista 6 ja sanakirjasta. Katso esimerkki.

Jos et halua määrittää väriä kolmessa komponentissa, voit myös jättää kaksi ensimmäistä parametria nolaksi ja antaa minusmerkkisen arvon 1-16.7 milj.

Voit käyttää väriarvojen tilalla myös CoolBasicin sisäänrakennettua värivakiota. Siten **Color cbRed** on sama kuin **Color 255,0,0**. Listan värivakioista löydät [täältä](#).

KÄYTTÖ

COLOR punainen, vihreä, sininen

tai

COLOR 0, 0, pikseliarvo

tai

COLOR värivakio

- **punainen** = Kokonaisluku. Punainen komponentti.
- **vihreä** = Kokonaisluku. Vihreä komponentti.
- **sininen** = Kokonaisluku. Sininen komponentti.
- **pikseliarvo** = Kokonaisluku. Muuttujaan luettu 32-bittinen pikseli tai kokonaisluku -16,7 milj -> 0.
- **värivakio** = Jokin CoolBasicin [värivakioista](#).

Katso myös: [CLSCOLOR](#)

ESIMERKKI

```
For r=0 To 255 Step 10
  Color r,0,0
  Box 10+r,10,10,10
Next r

For g=0 To 255 Step 10
  Color 0,g,0
  Box 10+g,30,10,10
Next g

For b=0 To 255 Step 10
  Color 0,0,b
  Box 10+b,50,10,10
Next b

For y=0 To 255 Step 10
  Color y,y,0
  Box 10+y,70,10,10
Next y

For o=0 To 255 Step 10
  Color o,o/2,0
  Box 10+o,90,10,10
Next o

For s=0 To 255 Step 10
  Color 0,s,s
  Box 10+s,110,10,10
Next s

For m=0 To 255 Step 10
  Color m,0,m
  Box 10+m,130,10,10
Next m

For w=0 To 255 Step 10
  Color w,w,w
  Box 10+w,150,10,10
Next w

For l=0 To 255 Step 10
  Color 0,1,l/2
```

```
    Box 10+1,170,10,10
Next 1

For l=0 To 255 Step 10
    Color 0,1/2,0
    Box 10+1,190,10,10
Next 1

For l=0 To 255 Step 10
    Color 1/2,1,1/4
    Box 10+1,210,10,10
Next 1

DrawScreen
WaitKey
```

COMMANDLINE

KUVAUS Kertoo ohjelman komentorivin merkkijonona. Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.
KÄYTTÖ COMMANDLINE ()
ESIMERKKI

CONST

KUVAUS

Määrittää ohjelmassa käytettävän vakion. Vakio on arvo, joka ei muutu. Käytännössä tämä tarkoittaa sitä, että luot muuttujan näköisen tunnisteeseen, jolle annat arvon. Kun käytät tätä tunnistetta ohjelmassasi, kääntäjä käyttää sen tilalla sille annettua arvoa. Esimerkiksi haluaisit käyttää piitä. Tiedämme että pii on vakio. Komennat sitten Const PII = 3.141592653489793. Sen sijaan, että kirjoittaisit tuon pitkän desimaaliluvun koodiin joka kerta, kun sitä tarvitaan, voit kirjoittaa sen tilalle yksinkertaisesti "PII". Vakioden käyttö tekee koodista selkeämpää ja nopeuttaa ohjelman suoritusta jonkin verran. Ne eivät myöskään vie muistia, toisin kuin muuttujat. Yleinen käytäntö on kirjoittaa vakionimet ISOLLA. Toisin kuin muuttujiin ja muihin tunnisteisiin, vakion tunnisteeseen perään ei kirjoiteta tyyppitunnisteita %, # ja \$.

KÄYTTÖ

```
CONST vakio = arvo
```

ESIMERKKI

```
Const ABC = "Howdy, fellows!"
Const GRAVITY = -0.02
```

Repeat

```
    c#=c#+GRAVITY
```

```
    Text 0,0,ABC
```

```
    Text 0,20,c#
```

```
    DrawScreen
```

```
Until EscapeKey()
```

CONVERTTOINTEGER

KUVAUS

Tämä funktio muuntaa tyyppimuuttujan (jäsenen) ns. osoittimeksi. Osoitin on tavallinen kokonaisluku, joka sisältää muistiosoitteen itse jäseneen. Koska CoolBasicissa ei suoraan toistaiseksi voida välittää tyyppimuuttujia funktioille, on käytettävä tätä kiertotietä. Ensin luodaan osoitin, joka välitetään funktiolle normaalisti kokonaislukumuuttujana. Funktion sisällä se käännetään uudestaan tyyppiksi **ConvertToType**-funktioilla.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

CONVERTTOINTEGER (tyyppi_muuttuja)

- tyyppi_muuttuja** = Tyypitetty muuttuja, josta osoitin tehdään

Katso myös: [CONVERTTOTYPE](#)

ESIMERKKI

```
'This example shows how to pass custom types to funtions

Type TEST
  Field idata
End Type

item.TEST= New (TEST)
item\idata= 34

passtype (ConvertToInteger (item) )

WaitKey
End

'-----

Function passtype (typeIN)

  myitem.TEST=ConvertToType (typeIN)

  Print myitem\idata

End Function
```

CONVERTTOTYPE

KUVAUS

Tämä funktio muuntaa osoittimen takaisin tyyppimuuttujaksi. Osoitin on tavallinen kokonaisluku, joka sisältää muistiosoitteen itse jäseneen. Koska CoolBasicissa ei suoraan toistaiseksi voida välittää tyyppimuuttujia funktioille, on käytettävä tätä kiertotietä. Ensin luodaan osoitin, joka välitetään funktiolle normaalisti kokonaislukumuuttujana. Funktion sisällä se käännetään uudestaan tyyppiksi.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

```
CONVERTTOTYPE (osoitin)
```

- **osoitin** = Osoitin, josta tyyppimuuttuja tehdään

Katso myös: [CONVERTTOINTEGER](#)

ESIMERKKI

```
'This example shows how to pass custom types to funtions
```

```
Type TEST
  Field idata
End Type
```

```
item.TEST= New(TEST)
item\idata= 34
```

```
passtype(ConvertToInteger(item))
```

```
WaitKey
End
```

```
'-----
```

```
Function passtype(typeIN)

  myitem.TEST=ConvertToType(typeIN)

  Print myitem\idata

End Function
```

COPYBOX

KUVAUS

Kopioi osan puskuria toiseen puskuriin. Mahdollisesti palan kuvasta ruudulle. Käytä **Screen**- ja **Image**-funktioita osoittamaan puskureihin. Jos puskurit jätetään esittämättä, käytetään nykyistä grafiikkapuskuria (todennäköisesti ruutua). Komento on erityisen kätevä, kun halutaan kopioida kuvasta kuvaan tai ruudulta kuvaan.

Läpinäkyvää väriä (maskia) ei oteta huomioon, vaan sekin kopioidaan sellaisenaan.

KÄYTTÖ

```
COPYBOX x1, y1, leveys, korkeus, x2, y2, [lähde_puskuri], [kohde_puskuri]
```

- **x1** = Kokonaisluku. Kopiointipaikka x-suunnassa.
- **y1** = Kokonaisluku. Kopiointipaikka y-suunnassa.
- **leveys** = Kokonaisluku. Kopioitavan palan leveys pikseleinä.
- **korkeus** = Kokonaisluku. Kopioitavan palan korkeus pikseleinä.
- **x2** = Kokonaisluku. Kohdepaikka x-suunnassa.
- **y2** = Kokonaisluku. Kohdepaikka y-suunnassa.
- **lähde_puskuri** (valinnainen) = Mistä puskurista kopiointi alkaa.
- **kohde_puskuri** (valinnainen) = Mihin puskuriin kopioidaan.

Katso myös: [SCREEN](#), [IMAGE](#)

ESIMERKKI

```
'Load the map and cow
themap=LoadImage("Media/map.bmp")
cow=LoadImage("Media/cow.bmp")

'Copy the cow to the map - permanently
CopyBox 0,0,30,25,100,80,Image(cow),Image(themap)

While Not EscapeKey()

    Cls

    'draw the map
    DrawImage themap,MouseX()-200,MouseY()-150

    DrawScreen

Wend

End
```

COPYFILE

KUVAUS

Kopioi annetun tiedoston toiseen paikkaan.

Jos kopio on jo olemassa, toimenpide epäonnistuu. Poista aina vanha tiedosto käyttämällä funktiota **FileExists** ja komentoa **DeleteFile**.

KÄYTTÖ

```
COPYFILE tiedosto1, tiedosto2
```

- **tiedosto1** = Merkkijono, joka määrittää kopioitavan tiedoston.
- **tiedosto2** = Merkkijono, joka määrittää uuden tiedoston.

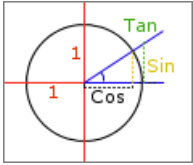
ESIMERKKI

COS

KUVAUS

Laskee kulman kosinin. Kulma annetaan asteina.

Tätä funktiota käytetään lähinnä apukeinona laskemaan paljonko pisteen pitää liikkua vaakasuunnassa, kun tiedetään liikkeen suunta (kulma) ja nopeus.



Oletetaan, että haluat laskea seuraavat ammuksen koordinaatit. Ammus on laukaistu kulmassa 60 astetta, nopeudella 5. Tästä voidaan muodostaa suorakulmainen kolmio. Ammus kulkee vinoa sivua pitkin. Sin laskee tämän kolmion pystysivun pituuden. Siten **x# = x# + Cos (60) * 5**.

Tulos on aina desimaaliluku väliltä -1 ja 1.

Kosinin pariaksi tarvitetset siniä (kts. Sin).

KÄYTTÖ

SIN (luku)

- **luku** = Kokonaisluku tai desimaaliluku.

Katso myös: [SIN](#), [ACOS](#)

ESIMERKKI

```
FrameLimit 40 'prevent the game from running too fast

AddText "Arrows to aim"
AddText "Space to shoot"
AddText "Esc to exit"

angle=90

Repeat

    'draw the ground
    Line 0,150,400,150

    'draw the turret
    Line 200,150,200+Cos(angle)*20,150-Sin(angle)*20

    If LeftKey() Then angle=angle+1
    If RightKey() Then angle=angle-1

    If angle<10 Then angle=10
    If angle>170 Then angle=170

    'shoot?
    If KeyHit(cbkeyspace) And bulletflying=0 Then
        bulletx#=200+Cos(angle)*20
        bullety#=150-Sin(angle)*20

        bulletmovex#=Cos(angle)*2 '2=speed
        bulletmovey#=-Sin(angle)*2 '2=speed

        bulletflying=1
    EndIf

    'update and draw bullet
    If bulletflying Then

        bulletx#=bulletx#+bulletmovex#
        bullety#=bullety#+bulletmovey#

        Dot bulletx#,bullety#

        'if the bullet goes outside screen, remove it
        If bulletx#<0 Or bulletx#>400 Or bullety#<0 Then
```

```
bulletflying=0
EndIf
```

```
EndIf
```

```
'draw the screen
DrawScreen
```

```
Until EscapeKey()
```

COUNTCOLLISIONS

KUVAUS

Laskee montako törmäysilmiötä objekti kohtasi sitten viime ruudun piirtämisen (tai *UpdateGamen*).

Käytä tulosta päästäksesi käsiksi kaikkiin törmäyksiin. Voit lukea jokaisen törmäyksen tapahtumapaikan, tapahtumakulman ja sen toisen objektin, joka osallistui törmäykseen. Voit käyttää funktiota myös määrittelemään tapahtuiko törmäyksiä ylipäättänsä ollenkaan.

Funktio palauttaa kokonaisluvun. Nollan, jos törmäyksiä ei tapahtunut.

KÄYTTÖ

COUNTCOLLISIONS (objektimuuttuja)

- **objektimuuttuja** = Objekti, jonka törmäykset tarkistetaan.

Katso myös: [COLLISIONX](#), [COLLISIONY](#), [COLLISIONANGLE](#), [GETCOLLISION](#)

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

'Load the map
map=LoadMap("Media\cdm2.til", "Media\tileset.bmp")

'Load player
soldier=LoadObject("Media\soldier.bmp", 72)
ObjectRange soldier, 32

'Load dumb immobile bot
bot=LoadObject("Media\soldier.bmp")
ObjectRange bot, 32
PositionObject bot, 50, 200

'setup some collisions
SetupCollision soldier, bot, 2, 2, 2
SetupCollision soldier, map, 1, 4, 2

Repeat

    'Update controls
    If LeftKey() Then TurnObject soldier, 5
    If RightKey() Then TurnObject soldier, -5
    If UpKey() Then MoveObject soldier, 3.5
    If DownKey() Then MoveObject soldier, -2

    'Camera follow
    CloneCameraPosition soldier

    'Enable additional graphics
    DrawGame

    collisions=CountCollisions(soldier)

    'iterate through all collisions
    For i=1 To collisions
        Color cbwhite
        Text 0, (i-1)*12, "angle: "+CollisionAngle(soldier, i)+" object: "+GetCollision(soldier, i)

        'draw the collision-ball
        Color cbred
        Circle CollisionX(soldier, i) -5, CollisionY(soldier, i)+5, 10
    Next i

    'render SCREEN
    DrawScreen

Until EscapeKey()
```

COUNTWORDS

KUVAUS

Laskee montako sanaa tekstissä on.

Token-ominaisuuksien vuoksi myös "tyhjät" sanat poimitaan. Sana on tyhjä, jos törmätään kahteen peräkkäiseen välilyöntiin.

Välimerkin ei ole pakko olla välilyönti. Voit määrittää myös toisen erottimen valinnaisella parametrilla (kts. alas).

KÄYTTÖ

```
COUNTWORDS (jono, [erotin])
```

- **jono** = Merkkijono, josta sanat lasketaan.
- **erotin** (*valinnainen*) = Yksi merkki. Esim. **"*"**. Oletuksena välilyönti. Tätä parametria ei ole pakko kirjoittaa.

Katso myös: [GETWORD](#)

ESIMERKKI

```
Repeat
  k$=Input("Type someting: ")

  Text 0,50,"Word count: "+CountWords(k$)
  DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
```

CRC32

KUVAUS

CRC32-algoritmia käytetään tarkastamaan korruptoitunutta dataa. Tiedostoista tai muistipaloista voidaan laskea ns. tarkistussumma, joka toimii vähän kuin sormenjälkenä tiedostoille. Monet tietoliikenneohjelmat käyttävät sitä varmistamaan, että siirrossa ei tapahtunut virheitä. Monet pakkausformaatit kuten *zip* ja *rar* sisältävät myös CRC-tarkistuksen; Viallisia tiedostoja ei pureta paketeistaan.

CRC32 on 32-bittinen sormenjälkialgoritmi, mikä tarkoittaa sitä, että tulos on 4-tavuinen kokonaisluku. Erilaisia yhdistelmiä on siten noin 4 miljardia. On olemassa teoreettinen mahdollisuus, että kahdella eri tiedostolla on sama sormenjälki, mutta se on käytännössä olematon. Tätä varten on kuitenkin kehitetty tarkempia sormenjälkimenetelmiä, kuten MD5 ja Base64.

Voit käyttää tätä funktiota tarkastamaan onko jotakin tiedostoa näpelöity, koska yhdenkin tavun muuttaminen vaikuttaa sormenjälkeen. Tämä on kätevää esim. pelin asetustiedostojen ja tallennettujen pelitilanteiden varmentamiseen.

Kryptauskomentojen tapaan tätä voi käyttää sekä tiedostoille että [muistipaloille](#).

KÄYTTÖ

CRC32 (lähde)

- **lähde** = Lähdetiedostonimi merkkijonona tai muistipalan muuttuja

ESIMERKKI

```
'get the file fingerprint for 'level.bmp' in the Media directory
checksum=CRC32("Media\level.bmp")
Print "File Fingerprint for 'level.bmp' "+Hex(checksum)
Print "Should be: ECAA0F4C"
Print

'check if someone has been modified it...
If Hex(checksum)="ECAA0F4C" Then
    Print "It's a match."
Else
    Print "Someone has been poking on the file..."
EndIf
WaitKey
```

CURRENTDIR

KUVAUS

Kertoo nykyisen hakemiston ilman päättävää kenoviivaa.

Käytä **ChDir**-komentoa vaihtamaan nykyinen hakemisto.

Tulos on merkkijono.

KÄYTTÖ

```
CURRENTDIR ()
```

Katso myös: [CHDIR](#)

ESIMERKKI

```
AddText "Current directory is: "  
AddText CurrentDir()  
  
AddText "Parent directory is: "  
ChDir ".."  
AddText CurrentDir()  
  
AddText "Changing to C:"  
ChDir "C:\"  
AddText CurrentDir()  
  
DrawScreen  
  
WaitKey
```

CURVEANGLE

KUVAUS

Sama kuin **CurveValue**, mutta pitää kulman välillä 0-360. Funktion toinen ihanuus on se, että CoolBasic valitsee automaattisesti sen lyhimmän polun, eli käännytään siihen suuntaan kun se olisi loogisinta.

Tämä funktio muuntaa lukuarvoa vähitellen toisen suuruiseksi. Voit itse määrittää muunnoksen nopeuden. Tekniikkaa kutsutaan interpoloinniksi, joka tarkoittaa "automaattista väliarvojen laskemista".

Funktio on ihanteellinen, kun halutaan liikkeisiin ja kääntymisiin pyöreyttä. Se saa pelin näyttämään sulavammalta ja paremmalta.

Funktio palauttaa desimaaliluvun. Suosittelemme käyttämään desimaalimuuttujia, kuten esimerkki näyttää.

KÄYTTÖ

```
CURVEANGLE (uusi, vanha, pyöreys)
```

- **uusi** = Tämä on se arvo, jota kohti lukua pyöristetään. Mielellään desimaaliluku tai -muuttuja.
- **vanha** = Nykyinen arvo. Mielellään desimaaliluku tai -muuttuja.
- **pyöreys** = Luku 1 (ei pyöreyttä) - 100 (erittäin pyöreä).

Katso myös: [CURVEVALUE](#)

ESIMERKKI

```
FrameLimit 40

AddText "The line will point to mouse"

Repeat

    'this is the angle that the line SHOULD point at
    newangle# = GetAngle(200,150,MouseX(),MouseY())

    'this is the interpolated (current) angle
    angle# = CurveAngle(newangle#,angle#,15.0)

    Circle 200-16,150-16,32,OFF
    Line 200,150,200+Cos(angle#)*48,150-Sin(angle#)*48

    DrawScreen

Until EscapeKey()
```

CURVEVALUE

KUVAUS

Tämä funktio muuntaa lukuarvoa vähitellen toisen suuruiseksi. Voit itse määrittää muunnoksen nopeuden. Tekniikkaa kutsutaan interpoloinniksi, joka tarkoittaa "automaattista väliarvojen laskemista".

Funktio on ihanteellinen, kun halutaan liikkeisiin ja kääntymisiin pyöreyttä. Se saa pelin näyttämään sulavammalta ja paremmalta.

Voit esimerkiksi käyttää tätä auton nopeuden muuttamiseen. Silloin esimerkiksi jarrutuksen toteuttaminen on erittäin yksinkertaista.

Funktio palauttaa desimaaliluvun. Suosittelemme käyttämään desimaalimuuttujia, kuten esimerkki näyttää.

KÄYTTÖ

```
CURVEVALUE (uusi, vanha, pyöreys)
```

- **uusi** = Tämä on se arvo, jota kohti lukua pyöristetään. Mielellään desimaaliluku tai -muuttuja.
- **vanha** = Nykyinen arvo. Mielellään desimaaliluku tai -muuttuja.
- **pyöreys** = Luku 1 (ei pyöreyttä) - 100 (erittäin pyöreä).

Katso myös: [CURVEANGLE](#)

ESIMERKKI

```
FrameLimit 40

AddText "The two circles will follow the mouse smoothly"

Repeat

    'less curving
    x1#=CurveValue(MouseX(),x1#,5.0)
    y1#=CurveValue(MouseY(),y1#,5.0)

    'more curving (smoother)
    x2#=CurveValue(MouseX(),x2#,20.0)
    y2#=CurveValue(MouseY(),y2#,20.0)

    Color cbyellow
    Circle x2-10,y2-10,20

    Color cbred
    Circle x1-8,y1-8,16

    DrawScreen

Until EscapeKey()
```


DATA

KUVAUS

Luo ohjelman sisälle datalistan, johon ohjelmoija voi tallentaa tietoa kokonaislukuina, desimaalilukuina tai merkkijonoina. Näin voit esimerkiksi sisällyttää yksinkertaisen pelikentän suoraan ohjelmaan.

Ei ole väliä, onko sinulla yksi iso datalista vai monta pienempää listaa. Kaikki yhteenkuuluvat listat pitää kuitenkin kirjoittaa peräkkäin ilman, että välissä on muuta koodia. Jos luet listan yli, saat siitä virheilmoituksen.

Voit sekoittaa erilaisia tietotyyppejä listaan, kunhan muistat lukea niitä oikeassa järjestyksessä. Datalistaa luetaan funktiolla **Read**.

KÄYTTÖ

DATA datalista

Datalista sisältää arvoja, jotka erotetaan pilkulla.

Katso myös: [READ](#), [RESTORE](#)

ESIMERKKI

```
Dim map(6,5)

'-----

'read level 1
Restore level1

For y=1 To 5
For x=1 To 6
    map(x,y)=READ()
Next x
Next y

'-----

'main loop
Repeat

    For y=1 To 5
    For x=1 To 6
        If map(x,y) Then Box x*40,y*40,40,40
    Next x
    Next y

    Box 40,40,240,200,OFF

    DrawScreen

Until EscapeKey()
End

'-----
'you can store more levels, too.

level1:

DATA 0,0,0,0,0,0
DATA 0,0,0,1,0,0
DATA 0,0,1,1,1,0
DATA 0,0,0,1,0,0
DATA 1,0,0,0,0,1
```

DATE

KUVAUS

Kertoo tietokoneen päiväyksen.

Huomaa, että kuukautta ei merkitä numeroin.

Tulos on merkkijono.

KÄYTTÖ

DATE ()

Katso myös: [TIME](#)

ESIMERKKI

```
Repeat
  Text 40,40,Date() 'print date
  Text 40,60,Time() 'print time
  Text 40,80,TIMER() 'print system timer

  DrawScreen

Until EscapeKey()
```

DECRYPT

KUVAUS

Tätä komentoa oli vaikea sijoittaa mihinkään luokkaan, joten se tipahti tänne sekalaiseen kastiin.

Purkaa aiemmin salatuntiedoston tai muistipalan (kts. **Encrypt**). Komennon kätevyys on siinä, että sillä voidaan ristiinprosessoida tiedostoja muistiin, muistista tiedostoihin, muistista muistiin ja tiedostosta tiedostoon. Voit jopa purkaa samalla lähde- ja kohdetiedostolla. Sama toimii myös muistipankeilla.

Salaamista tarvitaan esimerkiksi silloin, kun pitää varmistaa, ettei kukaan pääse muokkaamaan ns. selkokielisiä pelin tallennustiedostoja tai TOP-listoja. Sinun ei esimerkiksi tarvitse dekryptata (purkaa salausta) kovalevylle ennen lukemista, vaan senkin saa suoraan muistiin. Tutustu myös [muistipalakomentoihin](#).

VINKKI: Voit tarkastaa, onko tiedostoja kopeloitu komennolla CRC32. Voit käyttää sitä yhdessä kryptauskomentojen kanssa tiedon varmistamiseen.

KÄYTTÖ

DECRYPT lähde, kohde, salasana\$

- **lähde** = Lähdetiedostonimi tai muistipalan muuttuja
- **kohde** = Kohdetiedostonimi tai muistipalan muuttuja
- **salasana\$** = Merkkijono. Salausavain.

Katso myös: [ENCRYPT](#)

ESIMERKKI

```
'encrypt a funny joke to root C:
Encrypt "Media\Example.txt","C:\classified.dat","ILOOVB"
Print "Look at your root C:"
WaitKey

'decrypt the file so you can read it
Decrypt "C:\classified.dat","c:\decrypted.txt","ILOOVB"
Print "Look at your root C: again..."
WaitKey

'NOW.. compare the two files by opening both to Notepad
```

DEFAULTMASK

KUVAUS

Tutustu ensin komenttoon MaskImage.

Muuten sama kuin **MaskImage**, mutta tämä komento asettaa oletusmaskauksen päälle/pois. Näin sinun ei tarvitse asettaa maskia jokaiselle kuvalle erikseen, jos lataat esim. monta magenta-taustaista kuvaa.

Väri ilmoitetaan RGB-järjestelmän mukaisesti. Väri muodostuu kolmesta komponentista, joista kukin saa arvon 0-255. (0,0,0) on musta ja (255,255,255) on valkoinen. Lisätietoa RGB-koodauksesta oppitunnista [Värit](#), [värikartasta](#) ja sanakirjasta.

Yleinen vaihtoehtoinen väri on puhdas magenta (255,0,255 tai *cbMagenta*), koska se on niin harvinainen.



Huomaa komennon kaksi käyttötapaa. On esitettävä joko 1 parametria tai sitten 4. Ei 2 tai 3.

- PÄÄLLE: DefaultMask ON, r, g, b (*asettaa maskiväriksi jonkun haluamasi värin*)
- PÄÄLLE: DefaultMask ON (*asettaa maskiväriksi mustan, eli RGB 0,0,0*)
- POIS: DefaultMask OFF (*poistaa automaattimaskauksen käytöstä*)

KÄYTTÖ

DEFAULTMASK päällä, punainen, vihreä, sininen

- **päällä** = Se muuttuja, joka sisältää kuvan.
OFF = pois päältä
ON = päällä
- **punainen** = Kokonaisluku. Värin punainen komponentti.
- **vihreä** = Kokonaisluku. Värin vihreä komponentti.
- **sininen** = Kokonaisluku. Värin sininen komponentti.

Katso myös: [MASKIMAGE](#)

ESIMERKKI

```
ClsColor cbwhite: Cls
```

```
DefaultMask ON,255,0,255 'or "cbmagenta"  
img1=LoadAnimImage("Media\Zerorun1.bmp",47,51,0,13)  
DrawImage img1,0,0
```

```
DefaultMask OFF  
img2=LoadAnimImage("Media\Zerorun1.bmp",47,51,0,13)  
DrawImage img2,60,60
```

```
DrawScreen
```

```
WaitKey
```

Objektit : komennot

DEFAULTVISIBLE

KUVAUS

Tutustu ensin komentoon ShowObject.

Tämä komento asettaa objektien automaattisen piilotuksen päälle/pois. Jos et halua, että objektit tulevat näkyviin kun ne luodaan, ladataan tai kopioidaan, käytä tätä komentoa.

KÄYTTÖ

DEFAULTVISIBLE näkyvä

• näkyvä

ON = näkyvä

OFF = piilotettu

ESIMERKKI

DELETE

KUVAUS

Tämä komento poistaa kokoelman jäsenen.

Muista aina vapauttaa jäseneseen sidottu media, tai et pääse siihen enää käsiksi.

KÄYTTÖ

DELETE tyyppimuuttuja

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
Type BULLETS
  Field obj
End Type

'-----
'MAIN PROGRAM
'-----

FrameLimit 40

'load the soldier, turning detail 72
guy=LoadObject("Media\soldier.bmp",72)

'load the bullet (master-object) and hide it
bullet=LoadObject("Media\bullet.bmp")
ShowObject bullet,OFF

ClsColor cbdark

AddText "Arrows to turn, SPACE to shoot"

'MAIN LOOP
Repeat

  'Update turning
  If LeftKey() Then TurnObject guy,5
  If RightKey() Then TurnObject guy, -5

  'Shoot?
  If KeyDown(cbkeyspace) And reload=0

    newBullet.BULLETS= New(BULLETS)
    newBullet\obj=CloneObject(bullet)
    CloneObjectPosition newBullet\obj,guy
    CloneObjectOrientation newBullet\obj,guy
    'put the bullet at the end of the barrel
    MoveObject newBullet\obj,24

    reload=4
  EndIf

  'Update all BULLETS (move them forward)
  i=0
  For iBullet.BULLETS= Each BULLETS

    MoveObject iBullet\obj,6 'speed=6

    'delete bullet from collection if it
    'gets too far away
    If Distance2(iBullet\obj,guy)>200 Then
      DeleteObject iBullet\obj
      Delete iBullet
    EndIf

    i=i+1

  Next iBullet

  'Update reloading
  If reload>0 Then reload=reload-1
```

```
Text 30,30,"Bulets to update: "+i
```

```
'Don't forget this :)
```

```
DrawScreen
```

```
Until EscapeKey()
```

DELETEFILE

KUVAUS

Poistaa tiedoston tai kansion.

Tämän komennon kanssa pitää olla erityisen varovainen! Huolimaton käyttö voi poistaa tärkeitä tiedostoja. **Tiedostoja EI siirretä roskakoriin, vaan ne tuhotaan ilman erillistä varoitusta.** Kansioita, jotka sisältävät tiedostoja tai toisia kansioita, ei voi poistaa. Poistaaksesi ei-tyhjän kansion sinun pitää läpikäydä sen sisältö *tiedostonetsintäkomennolla* ja poistaa jokainen tiedosto ennen varsinaista kansion poistamista.

Ennen poistoa sinun tulisi varmistua, että tiedosto on olemassa, ja poiston jälkeen, että tiedosto ON poistettu. Tähän tarkoitukseen sopii **FileExists**.

KÄYTTÖ

DELETEFILE tiedosto

- **tiedosto** = Merkkijono, joka määrittää poistettavan tiedoston tai kansion nimen.

ESIMERKKI

DELETEFONT

KUVAUS

Poistaa aiemmin ladatun fontin muistista ja vapauttaa sen viemän muistin. Tämän jälkeen kyseistä fonttia ei voi enää käyttää, ellei sitä ladata uudestaan.

KÄYTTÖ

```
DELETEFONT fonttimuuttuja
```

- **fonttimuuttuja** = Se muuttuja, johon fontti ladattiin *LoadFont*-komennolla.

ESIMERKKI

```
bigfont=LoadFont("arial",48)

SetFont bigfont

Repeat
    Text 10,10,"Hit ESC to quit..."

    DrawScreen

Until EscapeKey()

DeleteFont bigfont
```

DELETEIMAGE

KUVAUS

Poistaa kuvan tietokoneen muistista. Tämän jälkeen sitä ei voi enää käyttää.

Sinun tulisi poistaa kuvat, joita et enää tarvitse. Näin pelisi ei syö ylimääräistä muistia. Esimerkiksi pelisi alkulogon voi huoletta poistaa, kun sitä ei tarvita enää.

Jos yrität käyttää jo poistettua kuvaa, ohjelma kaatuu Memory Access Violationiin.

KÄYTTÖ

DELETEIMAGE kuvamuuttuja

- **kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.

ESIMERKKI

```
SCREEN 640,480
```

```
'the starting logo!!
```

```
map=LoadImage("Media\map.bmp")
```

```
ResizeImage map,640,480
```

```
DrawImage map,0,0
```

```
CenterText 320,240,"This screen will end in 5 seconds..."
```

```
DrawScreen
```

```
Wait 5000
```

```
DeleteImage map
```

```
'...the game continues here
```

DELETEMEMBLOCK

KUVAUS

Poistaa muistipalan käytöstä ja vapauttaa sen varaaman muistin.

KÄYTTÖ

DELETEMEMBLOCK muistipala

- muistipala** = Se muuttuja, mihin muistipala luotiin.

ESIMERKKI

```
'create a memory block with 64 bytes
mem=MakeMEMblock (64)

WaitKey

DeleteMEMblock mem 'free the memory we used
```

Objektit : **komennot**

DELETEOBJECT

KUVAUS

Poistaa objektin muistista, ja se häviää välittömästi myös ruudulta. Voit käyttää komentoa kaikkiin objekteihin, kartat ja partikkelilähteet mukaan lukien. Kun objekti poistetaan, kaikki siihen liittyvät törmäykset mitätöityvät.

HUOM! Kloonattuja objekteja voi poistaa turvallisesti, mutta älä poista sitä objektia, mistä kloonit on tehty. Muuten kloonit muuttuvat epäkelvallisiksi ja kaatavat ohjelman. Jos teet paljon samoja objekteja (esim. ammuksia), kannattaa alkuperäinen objekti piilottaa. Kaikki muut ovat kloonit.

KÄYTTÖ

```
DELETEOBJECT objektimuuttuja
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [CLEAROBJECTS](#), [OBJECTLIFE](#)

ESIMERKKI

```
FrameLimit 40

'load object
cow=LoadObject("Media\cow.bmp", 72)

Color cbblack
AddText "Press SPACE to delete object..."
ClsColor cbyellow

Repeat

    'delete object on keyhit
    If KeyHit(cbkeyspace) Then
        DeleteObject cow
    EndIf

    'draw the screen
    DrawScreen

Until EscapeKey()
```

DELETESOUND

KUVAUS

Poistaa aiemmin ladatun äänen muistista. Tämän jälkeen ääntä ei enää voi käyttää, ellei sitä ladata uudestaan. Käytä tätä komentoa poistamaan ääni sitten, kun et enää tarvitse sitä. CoolBasic poistaa äänet muistista automaattisesti kun ohjelma lopetetaan, joten komennon ainoa käyttötarkoitus on väliaikainen muistin vapauttaminen.

KÄYTTÖ

DELETESOUND äänimuuttuja

- **äänimuuttuja** = Muuttuja, mihin ääni ladattiin *LoadSound*-funktiolla.

ESIMERKKI

```
weaponsound=LoadSound("Media\Blaster.wav")
```

Repeat

```
    If KeyHit(cbkeyspace) Then PlaySound weaponsound
```

Until EscapeKey()

```
DeleteSound weaponsound
```

DIM

KUVAUS

Luo taulukon muistiin. Taulukoissa voi olla useita ulottuvuuksia, mutta enintään viisi. Uuden ulottuvuuden määritteleminen voi kasvattaa dramaattisesti taulukon kokoa. Taulukot ovat käteviä, kun halutaan sijoittaa paljon tietoa yhden muuttujanimen taakse. Taulukoita voidaan käydä tehokkaasti läpi [For...Next-rakenteella](#) (katso esimerkki). Sen sijaan, että loisit 52*7 eri muuttujaa sijoittamaan vuoden oikeat lottorivit, voit varata taulukon (52,7). Vähentää kummasti kirjoitustyötä :)

Kuvittele, että taulukot ovat laatikoita. Jos taulukkosi on 1-ulotteinen, ja sisältää 6 solua, se sisältää siis 6 laatikkoa. Oletetaan, että määrittelet uuden ulottuvuuden, jossa on 5 solua. Silloin jokaiseen kuuteen laatikkoon ilmestyy 5 pienempää laatikkoa. Ja jos määrittelet taas uuden ulottuvuuden, vaikka 45 solulla, on sinulla kuusi laatikkoa, jokaisessa niistä 5 pienempää laatikkoa, ja jokaisen niiden sisällä vielä 45 pienempää laatikkoa. Sinulla on siis yhteensä $6*5*45=1350$ solua taulukossasi.

Huomioi kuitenkin, että kun määrittelet kuinka monta solua yhteen ulottuvuuteen tulee, myös nolla on mukana. Eli jos käsket Dim t(5), varaat oikeasti tilaa kuudelle paikalle: {0,1,2,3,4,5}.

Kaikki taulukot ovat globaaleja, eli ne "näkyvät" myös kaikissa funktioissa. Älä yritä lukea/kirjoittaa sellaisiin paikkoihin taulukossa, joita ei ole olemassa.

Voit esittää myös muuttujia tietyn tyyppisiksi. Tällöin käytetään AS-avainsanaa ja tyyppitunnistetta: **Integer, Float, String, Byte tai Short**. Lisätietoa kahdesta viimeisestä löydät manuaalin kohdasta "Tietotyytit". Huomaa, että voit määritellä useita muuttujia ja taulukoita samalla DIM-komennolla, kunhan ne on erotettu pilkulla. Esimerkki valaisee asiaa.

KÄYTTÖ

```
DIM muuttuja (ulottuvuus1_elementit [,ulottuvuus2_elementit] [,ulottuvuus3_elementit]
[,ulottuvuus4_elementit] [,ulottuvuus5_elementit])
```

- **ulottuvuus1_elementit** = Kokonaisluku, joka määrittää ensimmäisen ulottuvuuden solujen määrän.
- **ulottuvuus2_elementit** (valinnainen) = Kokonaisluku, joka määrittää toisen ulottuvuuden solujen määrän.
- **ulottuvuus3_elementit** (valinnainen) = Kokonaisluku, joka määrittää kolmannen ulottuvuuden solujen määrän.
- **ulottuvuus4_elementit** (valinnainen) = Kokonaisluku, joka määrittää neljännen ulottuvuuden solujen määrän.
- **ulottuvuus5_elementit** (valinnainen) = Kokonaisluku, joka määrittää viidennen ulottuvuuden solujen määrän.

Katso myös: [REDIM](#)

ESIMERKKI

```
Dim enemy_x#(100)
Dim enemy_y#(100)
Dim enemy_plusx(100) As Float,enemy_plusy(100) As Float

Dim i As Integer
Dim angle As Integer

'First apply a random position for all enemies
For i=1 To 100
    enemy_x(i)=Rand(390)
    enemy_y(i)=Rand(290)
    angle=Rand(360)
    enemy_plusx(i)=Cos(angle)
    enemy_plusy(i)=Sin(angle)
Next i

Color 0,255,0

'The main loop
Repeat

    'update all enemies
    For i=1 To 100
        'draw
        Box enemy_x(i),enemy_y(i),10,10,OFF

        'move
        enemy_x(i)=enemy_x(i)+enemy_plusx(i)
        enemy_y(i)=enemy_y(i)+enemy_plusy(i)

        'bounce
        If enemy_x(i)<0 Or enemy_x(i)>390 Then enemy_plusx(i)= -enemy_plusx(i)
        If enemy_y(i)<0 Or enemy_y(i)>290 Then enemy_plusy(i)= -enemy_plusy(i)
    Next i

    DrawScreen

Until EscapeKey()
```

DISTANCE

KUVAUS

Kertoo kahden pisteen välisen matkan. Käytä tätä komentoa laskemaan kuinka kaukana pisteet ovat toisistaan.

Funktio sopii esimerkiksi kertomaan, onko vihollinen liian kaukana nähdäkseen pelaajan.

Palautettu matka on desimaaliluku.

KÄYTTÖ

```
DISTANCE (x1, y1, x2, y2)
```

- **x1** = Ensimmäisen pisteen X-koordinaatti.
- **y1** = Ensimmäisen pisteen Y-koordinaatti.
- **x2** = Toisen pisteen X-koordinaatti.
- **y2** = Toisen pisteen Y-koordinaatti.

Katso myös: [DISTANCE2](#)

ESIMERKKI

```
AddText "Screen size is 400x300"
```

```
Repeat
```

```
    Circle 190,140,20,OFF
```

```
    Line 200,150,MouseX(),MouseY()
```

```
    Text 20,20,"Distance to mouse is: "+Distance(200,150,MouseX(),MouseY())
```

```
    DrawScreen
```

```
Until EscapeKey()
```

DISTANCE2

KUVAUS

Kertoo kahden objektin välisen etäisyyden pikseleissä.

Tulos on desimaaliluku.

KÄYTTÖ

DISTANCE2 (obj1, obj2)

- **obj1** = Objekti 1.
- **obj2** = Objekti 2.

Katso myös: [DISTANCE](#), [GETANGLE2](#)

ESIMERKKI

```
Dim targetx As Float
Dim targety As Float

Dim obj2x As Float
Dim obj2y As Float

DrawToWorld ON
FrameLimit 40

obj1=LoadObject("Media\guy.bmp")
obj2=LoadObject("Media\guy.bmp")

AddText "The code updates every 3 seconds..."

Repeat

    'Perform this every 3 seconds
    If TIMER()>moment+3000 Then

        targetx=Rnd( -180,180)
        targety=Rnd( -130,130)

        moment=TIMER()
    EndIf

    'curve location, round it against the target
    obj2x=CurveValue(targetx,obj2x,20.0)
    obj2y=CurveValue(targety,obj2y,20.0)

    PositionObject obj2,obj2x,obj2y
    Text 0,30,"Distance: "+Distance2(obj1,obj2)

    DrawScreen

Until EscapeKey()
```


DOT

KUVAUS

Piirtää yhden kuvapisteen. Käyttää nykyistä edusväriä (**Color**-käsky).

KÄYTTÖ

DOT x, y

- **x** = Kokonaisluku tai desimaaliluku. Pistein sijainti vaakasuunnassa.
- **y** = Kokonaisluku tai desimaaliluku. Pistein sijainti pystysuunnassa.

Katso myös: [LINE](#), [BOX](#), [CIRCLE](#), [ELLIPSE](#)

ESIMERKKI

```
Dim k As Float

FrameLimit 40 'limit game speed

ClsColor cbwhite 'background to white

Repeat

    angle=WrapAngle(angle+1) 'increase angle

    'draw dots
    k=0
    For i= angle To angle-90 Step -1
        Color k,k,k
        k=k+2.8

        Dot 200+Cos(i)*100,150-Sin(i)*100
    Next i

    'print some info
    Color cbblack
    Text 0,0,"Angle: "+angle

    'draw the center
    Dot 200,150

    DrawScreen

Until EscapeKey()
```

DOWNKEY

KUVAUS
Palauttaa 1, jos alas -nuolinäppäintä painetaan. Muutoin palautetaan nolla.

KÄYTTÖ

DOWNKEY ()

Katso myös: [LEFTKEY](#), [RIGHTKEY](#), [UPKEY](#)

ESIMERKKI

```
AddText "Use arrows to move..."

x=200:y=150

Repeat
    'update controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1

    'draw "player"
    Box x,y,10,10

    DrawScreen

Until EscapeKey()
```

DRAWANIMATION

KUVAUS

Tämä komento piirtää taustalla soivan videon.

Ruudun täytyy olla tarpeeksi iso, tai videota ei voi näyttää. Tarkasta täsmällinen koko funktioilla AnimationWidth ja AnimationHeight. Muuten videota voi venyttää ruudussa, joskin se toimii nopeimmin alkuperäisessä koossaan.

Huomaa, että kun video avataan, se lähtee välittömästi pyörimään. Video soi taustalla ilman keskeytyksiä alkuperäisen aikajanan mukaisesti, joten videota ei voi keskeyttää jättämällä kutsumatta DrawAnimation-komentoa.

Jos haluat poistua videotiostosta ennen kuin se loppuu, sinun täytyy kutsua **StopAnimation**-komentoa.

KÄYTTÖ

```
DRAWANIMATION videomuuttuja, [X], [Y], [leveys], [korkeus]
```

- **videomuuttuja** = Se muuttuja, mihin video käynnistettiin PlayAnimation-funktiolla.
- **X** (**valinnainen**) = Kokonaisluku, joka määrittää videon vasemman yläkulman X-koordinaatin. Oletuksena nolla.
- **Y** (**valinnainen**) = Kokonaisluku, joka määrittää videon vasemman yläkulman Y-koordinaatin. Oletuksena nolla.
- **leveys** (**valinnainen**) = Kokonaisluku, joka määrittää videon leveyden pikseleinä. Oletuksena videon oletuskoko.
- **korkeus** (**valinnainen**) = Kokonaisluku, joka määrittää videon korkeuden pikseleinä. Oletuksena videon oletuskoko.

Katso myös: [PLAYANIMATION](#), [ANIMATIONPLAYING](#)

ESIMERKKI

```
SCREEN 640,480

'load the video
video= PlayAnimation("myMovie.avi")

'draw video
Repeat

    DrawAnimation video

    DrawScreen

Until EscapeKey() Or KeyHit(28)

'close the video input
StopAnimation video
```

DRAWGAME

KUVAUS

Piirtää kaikki kartat, objektit ja partikkelit perustuen kameran sijaintiin.

Tätä komentoa ei ole pakko kutsua, vaan CoolBasic suorittaa sen automaattisesti ennen kuin ruutu piirretään.

Joskus peli pitää piirtää ennen muita grafiikkakomentoja. Esimerkiksi teksti voi muuten peittyä kartan tai "lattiaobjektin" alle. Käytä tätä komentoa piirtämään peli ennen tekstiä ja muuta grafiikkaa.

KÄYTTÖ

DRAWGAME

ESIMERKKI

DrawToWorld ON

FrameLimit 40

'Create scene

grass= MakeObjectFloor()

lawn= LoadImage("Media\grass.bmp")

PaintObject grass, lawn

character= LoadObject("Media\guy.bmp",72)

AddText "Use arrows to move..."

Repeat

'Guide

If LeftKey() Then TurnObject character,5

If RightKey() Then TurnObject character, -5

If UpKey() Then MoveObject character,2

If DownKey() Then MoveObject character, -2

'Camera follow

CloneCameraPosition character

DrawGame

'draw a hollow circle at the center

Circle-5,5,10,OFF

'draw a line

Line 0,0,50, -50

'draw a box

Box-100,100,200,160,OFF

DrawScreen

Until EscapeKey()

DRAWGHOSTIMAGE

KUVAUS

Tällä komennolla voit piirtää läpinäkyvän kuvan. Tekniikka kutsutaan myös nimellä alpha-blending eli alfa-blendaus, blend. Toisin kuin GhostObject, myös musta väri osallistuu läpinäkyvyyteen. Arvo nolla piilottaa objektin kokonaan (silti objekti osallistuu törmäyksen tunnistukseen). Arvo 100 on yhtä kuin täysin näkyvä kuva. Arvo 50 taas on puolinäkyvä. Voit itse päättää läpinäkyvyyden tason asteikolla 1-99.

VAROITUS! läpinäkyvien kuvien piirtäminen on melkoisen raskas urakka nykymalleillekin ilman laitteistokiihdytystä (3D). Sen liiallinen käyttö saattaa vaikuttaa dramaattisesti pelin suorituskykyyn. Käytä ominaisuutta säästeliäästi ja mieluiten lähinnä pienikokoisille kuville.

KÄYTTÖ

DRAWGHOSTIMAGE kuvamuuttuja, x, y, [frame], [läpinäkyvyys]

- **kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.
- **x** = Kokonaisluku tai desimaaliluku. Kuvan sijainti leveyssuunnassa.
- **y** = Kokonaisluku tai desimaaliluku. Kuvan sijainti pystysuunnassa.
- **frame** = Kokonaisluku. Kuvan animaatio-frame. Jos et tarvitse framea, laita arvoksi nolla tai OFF.
- **läpinäkyvyys** Arvo väliltä 1-99. Kuinka läpinäkyvä kuva on

Katso myös: [GHOSTOBJECT](#)

ESIMERKKI

```
map= LoadImage ("Media\map.bmp")
tex= LoadImage ("Media\grass.bmp")

alpha=100

AddText "A/Z to alter transparency"

Repeat

    'alter transparency
    alpha= alpha+ (KeyDown (cbkeya) -KeyDown (cbkeyz))

    DrawImage map,0,0 'draw a regular image
    DrawGhostImage tex, 136, 86, 0, alpha 'draw the transparent image

    Text 0,40,FPS()+" "+alpha 'show how fast the game is able to render
    DrawScreen ()

Forever
```

DRAWIMAGE

KUVAUS

Piirtää **ennalta ladatun** kuvan ruudulle annettuihin koordinaatteihin. Tämä komento piirtää myös animoituja kuvia (sprite-sarjoja), jotka on ladattu käyttäen **LoadAnimImage**-funktia. Tällöin koordinaattien perään kirjoitetaan vielä animaatio-frames numero (alkaen nolasta).

Voit piirtää kuvia myös ruudun ulkopuolelle, koska komento osaa itse rajata piirrettävän alueen. Koordinaateiksi kelpaavat siten myös negatiiviset luvut.

Kuva piirretään niin, että sen vasen yläkulma on annetuissa koordinaateissa. Siten (0,0) piirtää kuvan ruudun vasempaan yläreunaan.

Kun kuva piirretään, sen läpinäkyvänä värinä toimii musta (arvoltaan 0,0,0). Tätä sanotaan maskiväriksi. Jos haluat läpinäkyvyyden eri väreille, käytä **MaskImage**-komentoa. Voit myös täydellisesti kieltää läpinäkyvyyden millekään väreille kirjoittamalla valinnaisen parametrin loppuun.

Huomaa, että kuva on piirrettävä ruudulle joka kerta ennen DrawScreeniä. Yksi kerta ei riitä, tai kuva häviää.

KÄYTTÖ

```
DRAWIMAGE kuvamuuttuja, x, y, [frame], [läpinäkyvyys]
```

- **kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.
- **x** = Kokonaisluku tai desimaaliluku. Kuvan sijainti leveyssuunnassa.
- **y** = Kokonaisluku tai desimaaliluku. Kuvan sijainti pystysuunnassa.
- **frame** (**valinnainen**) = Kokonaisluku. Kuvan animaatio-frame. Oletuksena nolla.
- **läpinäkyvyys** (**valinnainen**)
0 = ei läpinäkyvyyttä
1 = normaali läpinäkyvyys (oletus)

Katso myös: [LOADIMAGE](#), [LOADANIMIMAGE](#), [MASKIMAGE](#)

ESIMERKKI

```
FrameLimit 40

'Load the map
themap=LoadImage("Media/map.bmp")

'Load an animated image
'Frame size is 30,25 \ start at the first \ 2 frames total
cow=LoadAnimImage("Media/animCow.bmp", 30, 25, 0, 2)

'Default Position
x=200:y=150

While Not EscapeKey()

    'Upadte controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1

    'Fraw the map
    DrawImage themap,0,0

    'Draw the cow
    DrawImage cow,x,y,cowframe

    'Animate through frames 0 and 1
    If TIMER()>cowtimer+100 Then 'animate every 100 millisecs
        cowframe=cowframe+1
        If cowframe>1 Then cowframe=0
        cowtimer=TIMER()
    EndIf

    DrawScreen

Wend

End
```

DRAWIMAGEBOX

KUVAUS

Piirtää **ennalta ladatun** kuvan ruudulle annettuihin koordinaatteihin. Tämä komento piirtää myös animoituja kuvia (sprite-sarjoja), jotka on ladattu käyttäen **LoadAnimImage**-funktia. Tällöin koordinaattien perään kirjoitetaan vielä animaatio-frames numero (alkaen nolasta).

Toisin kuin **DrawImage**, voit piirtää vain OSAN kuvasta. Esimerkiksi lataus/health -palkit.

Voit piirtää kuvia myös ruudun ulkopuolelle, koska komento osaa itse rajata piirrettävän alueen. Koordinaateiksi kelpaavat siten myös negatiiviset luvut.

Kuva piirretään niin, että sen vasen yläkulma on annetuissa koordinaateissa. Siten (0,0) piirtää kuvan ruudun vasempaan yläreunaan.

Kun kuva piirretään, sen läpinäkyvänä värinä toimii musta (arvoltaan 0,0,0). Tätä sanotaan maskiväriksi. Jos haluat läpinäkyvyyden eri väreille, käytä **MaskImage**-komentoa. Voit myös täydellisesti kieltää läpinäkyvyyden millekään väreille kirjoittamalla valinnaisen parametrin loppuun.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

DRAWIMAGEBOX kuvamuuttuja, x, y, aloitusX, aloitusY, leveys, korkeus, [frame], [läpinäkyvyys]

- **kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.
- **x** = Kokonaisluku tai desimaaliluku. Kuvan sijainti leveyssuunnassa.
- **y** = Kokonaisluku tai desimaaliluku. Kuvan sijainti pystysuunnassa.
- **aloitusX** = Kokonaisluku tai desimaaliluku. Mistä kohtaa kuvaa piirtäminen aloitetaan.
- **aloitusY** = Kokonaisluku tai desimaaliluku. Mistä kohtaa kuvaa piirtäminen aloitetaan.
- **leveys** = Kokonaisluku tai desimaaliluku. Kuinka leveästi (pikseleinä).
- **korkeus** = Kokonaisluku tai desimaaliluku. Kuinka korkeasti (pikseleinä).
- **frame** (valinnainen) = Kokonaisluku. Kuvan animaatio-frame. Oletuksena nolla.
- **läpinäkyvyys** (valinnainen)
- **0** = ei läpinäkyvyyttä
- **1** = normaali läpinäkyvyys (oletus)

Katso myös: [DRAWIMAGE](#), [LOADIMAGE](#), [LOADANIMIMAGE](#), [MASKIMAGE](#)

ESIMERKKI

```
FrameLimit 40

imglifebar=LoadImage("Media/life.jpg")

life=100

While Not EscapeKey()

    'draw a constant portion
    DrawImageBox imglifebar,60,140,50,5,100,15

    'draw the changing portion
    DrawImageBox imglifebar,40,40,0,0,life*2,26

    'animate
    life=life-1
    If life<0 Then life=100

    DrawScreen

Wend
```

DRAWSCREEN

KUVAUS

Piirtää ruudun. Ilman tätä et näe mitään ruudulla. Tämä on CoolBasicin tärkein komento!

VSyn on mekanismi, joka tapahtuu aina näytön piirtämisen aikana. Se varmistaa, että näyttö on piirretty loppuun ennen uuden jakson piirtämistä. Tämä vähentää välkettä ja vaakasuoria päivitysraitoja/-häiriöitä. CoolBasic voi piirtää ruudun kahdella tavalla: Pystysynkronoinnista huolehtii joko prosessori tai näytönohjain. Oletuksena prosessori, joka on yleensä näytönohjainta tarkempi. Jos näytön piirtämisessä on minkäänlaisia häiriöitä, kokeile komentaa **Drawscreen ON,ON**.

Vain **Print** ja **Write** tuottavat jotain ruudulle ilman tätä komentoa.

KÄYTTÖ

```
DRAWSCREEN [auto-CLS], [päivitystapa]
```

- **auto-CLS** (valinnainen)

ON = Tyhjennä näyttö automaattisesti joka kerralla (oletus)

OFF = Älä tyhjennä näyttöä, säilytä edellinen sisältö

- **päivitystapa** (valinnainen)

ON = näytönohjaimen VSync

OFF = prosessorin VSync (oletus)

ESIMERKKI

```
Color cbsilver
```

```
Repeat
```

```
    Box 100,50,200,200
```

```
    DrawScreen
```

```
Until EscapeKey()
```


DRAWTOIMAGE

KUVAUS

Tämä komento ohjaa kaikki piirtokomennot - mukaan lukien kuvat, tekstin ja pelimaailman - kuvaan. Voit siis kirjaimellisesti piirtää kuvaan. Esimerkiksi myöhempää tallennusta varten (**SaveImage**) tai vaikkapa kirjoittamasi suotimen käsiteltäväksi.

Yleensä komentoa tulee käytettyä, kun vasta luotuun kuvaan (**MakeImage**) pitäisi saada jotain sisältöä.

Piirtäminen palautetaan ruudulle komennolla **DrawToScreen**.

KÄYTTÖ

DRAWTOIMAGE kuvamuuttuja

- **kuvamuuttuja** = Se muuttuja, mihin kuva luotiin tai ladattiin.

Katso myös: [DRAWTOSCREEN](#)

ESIMERKKI

```
themap=LoadImage("Media/map.bmp")

angle=0

While Not EscapeKey()

    Cls

    'Draw the map to mouse coordinates
    DrawImage themap,MouseX() -200,MouseY() -150

    'Manipulate image every second
    If TIMER()>moment+1000 Then

        DrawToImage themap

        'red filled circle
        Color 255,0,0
        Circle 200+Cos(angle)*100-5,150-Sin(angle)*100-5,10

        'black hollow circle (outlines)
        Color 0,0,0
        Circle 200+Cos(angle)*100-5,150-Sin(angle)*100-5,10,OFF

        angle=angle+10

        moment=TIMER()

        'set drawing back to screen
        DrawToScreen

    EndIf

    'Let's see what we have here
    DrawScreen

Wend

End
```

DRAWTOSCREEN

KUVAUS

Tämä komento ohjaa kaikki piirtokomennot - mukaan lukien kuvat, tekstin ja pelimaailman takaisin näytölle sen jälkeen kun et enää halua jatkaa kuvaan piirtämistä (komento **DrawToImage**).

KÄYTTÖ

DRAWTOSCREEN

Katso myös: [DRAWTOIMAGE](#)

ESIMERKKI

```
themap=LoadImage("Media/map.bmp")

angle=0

While Not EscapeKey()

    Cls

    'Draw the map to mouse coordinates
    DrawImage themap,MouseX() -200,MouseY() -150

    'Manipulate image every second
    If TIMER()>moment+1000 Then

        DrawToImage themap

        'red filled circle
        Color 255,0,0
        Circle 200+Cos(angle)*100-5,150-Sin(angle)*100-5,10

        'black hollow circle (outlines)
        Color 0,0,0
        Circle 200+Cos(angle)*100-5,150-Sin(angle)*100-5,10,OFF

        angle=angle+10

        moment=TIMER()

        'set drawing back to screen
        DrawToScreen

    EndIf

    'Let's see what we have here
    DrawScreen

Wend

End
```

DRAWTOWORLD

KUVAUS

Pelimaailman koordinaatit ovat erilaisia kuin ruudun koordinaatit:

□ □

Normaalisti piirtokomennot (line, box, circle, jne.) piirtävät ruutukoordinaattien mukaan. CoolBasicissa on kuitenkin mahdollista piirtää esimerkiksi viivoja myös pelimaailman sisälle, jolloin kameran sijainti vaikuttaa siihen, mikä osa viivasta on näkyvissä. Voit käyttää viivaa tällä tavoin esimerkiksi sädetykin (*rail gun*) toteuttamiseen. Samaan tapaan voit piirtää ympyröitä ja laatikoita pelimaailman osaksi.

Jotta grafiikka piirtyisi oikein, on yleensä hyvä kutsua piirtokomentoja **DrawGame**- ja **DrawScreen**-komentojen välissä. Normaalistihan DrawGamea ei tarvitse käyttää, koska CoolBasic suorittaa sen automaattisesti, mutta muutoin on vaara, että esimerkiksi kartta peittää grafiikan, koska se piirretään grafiikan jälkeen. Kokeile esimerkkiä ilman **DrawGame**-komentoa ja sen kanssa.

Teksti ja kuvat piirtyvät myöskin oletuksellisesti ruutukoordinaattien mukaan, mutta voit tällä komennolla vaikuttaa niidenkin sijaintiin. Esimerkiksi pelihahmojen tietoja voidaan tulostaa niiden viereen pelimaailmassa.

Toiminto on oletuksellisesti pois päältä, joten piirtokomennot noudattavat totuttuja ruutukoordinaatteja.

KÄYTTÖ

DRAWTOWORLD piirtokomennot, [kuvat], [teksti]

• piirtokomennot

ON = piirrä viivat, laatikot, ympyrät, ovaalit ja pisteet pelimaailmaan

OFF = piirrä normaalisti

• kuvat (valinnainen)

ON = piirrä pelimaailmaan

OFF = piirrä kuvat normaalisti (oletus)

• teksti (valinnainen)

ON = piirrä teksti (Text-komento) pelimaailmaan

OFF = piirrä teksti normaalisti (oletus)

ESIMERKKI

```
'Set all drawing commands to draw to game world
```

```
DrawToWorld ON
```

```
FrameLimit 40
```

```
'Create scene
```

```
grass= MakeObjectFloor()
```

```
lawn= LoadImage("Media\grass.bmp")
```

```
PaintObject grass, lawn
```

```
character= LoadObject("Media\guy.bmp",72)
```

```
AddText "Use arrows to move..."
```

```
Repeat
```

```
  'Guide
```

```
  If LeftKey() Then TurnObject character,5
```

```
  If RightKey() Then TurnObject character, -5
```

```
  If UpKey() Then MoveObject character,2
```

```
  If DownKey() Then MoveObject character, -2
```

```
  'Camera follow
```

```
  CloneCameraPosition character
```

```
  DrawGame
```

```
  'draw a hollow circle at the center
```

```
  Circle-5,5,10,OFF
```

```
  'draw a line
```

```
  Line 0,0,50, -50
```

```
  'draw a box
```

```
  Box-100,100,200,160,OFF
```

```
  DrawScreen
```

```
Until EscapeKey()
```


FOR

KUVAUS

Muodostaa silmukan, joka käy läpi kaikki TYPIN jäsenet. Silmukkaan sidotaan tyyppimuuttuja, jonka avulla pääset käsiksi jäseniin.

Silmukka päätetään komenttoon NEXT. Pääsilmuksiksi For...Next -rakenteesta ei ole. Silmukasta voidaan rikkoutua väkisin ulos komennolla **Exit**.

For... Each -silmukat ja TYPIT yleensä sopivat mainiosti hallitsemaan suuria määriä ammuksia, vihollisia, objekteja jne... Lisää asiasta muiden TYPPIKOMENTOJEN oppaista.

KÄYTTÖ

```
FOR muuttuja = EACH tyyppi_nimi
    'koodia
NEXT sama_muuttuja
```

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
'First create our type
Type MYTYPE
    Field value
End Type

'Then create 10 elements
For i=1 To 10
    myvar.MYTYPE = New (MYTYPE)

    'assign a random value to each element
    myvar\value = Rand(10)
Next i

'Finally print all elements
For myvar.MYTYPE = Each MYTYPE

    AddText Str(myvar\value)

Next myvar

'Show the results
DrawScreen
WaitKey
```

EDITMAP

KUVAUS

Tämä komento muokkaa karttaa lennossa. Tämä tarkoittaa sitä, että voit rakentaa ja muokata tilekarttoja pelin aikana, eikä ohjelman suoritusnopeus kärsi ollenkaan.

Voit esimerkiksi räjäyttää seinä pois tai luoda esteitä pelaajan tielle. Tai sitten voit muuttaa veden laavaksi ja niin pois päin...

Tilekartat sisältävät tiedon siitä, mikä osa tilesettiä piirretään mihinkin paikkaan. Tilekartat ovat taulukoita, jotka sisältävät kuvien tapaan tiedon ei pikseleistä vaan piirrettävistä tileistä. Kuvitellaan, että sinulla on taulukko, joka on kirjoitettu täyteen ykköistä. Silloin koko kartta täytetään tilesetin ensimmäisellä kuvalla. Voit muokata ajon aikana tätä tietoa, ja muuttaa oleellisesti kartan ulkonäköä tai törmäysaluetta.

CoolBasicin kartoissa on neljä kerrosta: taustakerros, päälliskerros, törmäyskerros ja datakerros. **Vain kaksi ensimmäistä piirretään.** Numerointi menee 0-3. Objektit piirretään taustan ja päällisen väliin. Törmäyskerros sisältää vain nollia tai ykkösiä; nolla=ei törmäystä / 1=törmäys. Jos siis räjäytät seinän pois, sinun täytyy muokata sekä piirrettävää kerrosta (yleensä 1) että törmäyskerrosta (2).

Koordinaatit annetaan tileinä. Kartan vasen yläkulma on (1,1).

Komennon avulla onnistuu melkein mikä vain - vain mielikuvitus rajoittaa.

KÄYTTÖ

EDITMAP kartta, kerros, x, y, tile

- **kartta** = Se karttamuuttuja, mihin kartta luotiin tai ladattiin.
- **kerros**
 - 0 = taustakerros
 - 1 = päälliskerros
 - 2 = törmäyskerros
 - 3 = datakerros
- **x** = Kokonaisluku. X-koordinatti tileinä.
- **y** = Kokonaisluku. Y-koordinatti tileinä.
- **tile** = Kokonaisluku. Piirrettävän tilen järjestysnumero (1->).

Katso myös: [GETMAP2](#)

ESIMERKKI

```
FrameLimit 40

'Create a map from Blank
map=MakeMap(32,32,32,32)
u=LoadObject("Media\guy.bmp",72)

'Apply some graphics on it
tileset=LoadImage("Media\tileset.bmp")
PaintObject map,tileset

'Randomly fill the map
For y=1 To MapHeight()
For x=1 To MapWidth()
    If Rand(1,5)=5 Then
        EditMap map,0,x,y,2 'drawing
        EditMap map,2,x,y,1 'collision
    Else
        EditMap map,0,x,y,145
    EndIf
Next x
Next y

'The game loop-itself
Repeat

    If UpKey() Then MoveObject u,2
    If DownKey() Then MoveObject u, -2
    If LeftKey() Then TurnObject u,5
    If RightKey() Then TurnObject u, -5

    'Glue camera to object position
    CloneCameraPosition u

    DrawScreen

Forever
```

ELLIPSE

KUVAUS

Piirtää ellipsin. Käyttää nykyistä edusväriä (**Color**-käsky).

KÄYTTÖ

ELLIPSE x, y, leveys, korkeus, [kiinteä]

- **x** = Kokonaisluku tai desimaaliluku. Alkupisteen vaakakoordinaatti.
- **y** = Kokonaisluku tai desimaaliluku. Alkupisteen pystykoordinaatti.
- **leveys** = Kokonaisluku tai desimaaliluku. Ellipsin leveys pikseleinä.
- **korkeus** = Kokonaisluku tai desimaaliluku. Ellipsin korkeus pikseleinä.
- **kiinteä** (valinnainen)
 - 0** = ontto
 - 1** = täytetty (oletus)

Katso myös: [DOT](#), [LINE](#), [BOX](#), [CIRCLE](#)

ESIMERKKI

```
FrameLimit 40

Color cbred

ep=1
e=50

Repeat

    'animate
    e=e+ep
    If e>100 Or e<0 Then ep= -ep

    'draw
    Ellipse 200-e/2,150-((100-e)/2),e,100-e

    DrawScreen

Until EscapeKey()
```

ENCRYPT

KUVAUS

Tätä komentoa oli vaikea sijoittaa mihinkään luokkaan, joten se tipahti tänne sekalaiseen kastiin.

Kryptaa, eli salaa tiedoston tai muistipalan annetulla salasanalla. Komennon kätevyys on siinä, että sillä voidaan ristiinkryptata tiedostoja muistiin, muistista tiedostoihin, muistista muistiin ja tiedostosta tiedostoon. Voit jopa kryptata samalla lähde- ja kohdetiedostolla. Sama toimii myös muistipankeilla.

Salaamista tarvitaan esimerkiksi silloin, kun pitää varmistaa, ettei kukaan pääse muokkaamaan ns. selkokielisiä pelin tallennustiedostoja tai TOP-listoja. Sinun ei esimerkiksi tarvitse dekryptata (purkaa salausta) kovalevylle ennen lukemista, vaan senkin saa suoraan muistiin. Tutustu myös [muistipalakomentoihin](#).

VINKKI: Voit tarkastaa, onko tiedostoja kopeloitu komennolla CRC32. Voit käyttää sitä yhdessä kryptauskomentojen kanssa tiedon varmistamiseen.

KÄYTTÖ

ENCRYPT lähde, kohde, salasana\$

- **lähde** = Lähdetiedostonimi tai muistipalan muuttuja
- **kohde** = Kohdetiedostonimi tai muistipalan muuttuja
- **salasana\$** = Merkkijono. Salausavain.

Katso myös: [DECRYPT](#)

ESIMERKKI

```
'encrypt a funny joke to root C:
Encrypt "Media\Example.txt","C:\classified.dat","ILOOVB"
Print "Look at your root C:"
WaitKey

'decrypt the file so you can read it
Decrypt "C:\classified.dat","c:\decrypted.txt","ILOOVB"
Print "Look at your root C: again..."
WaitKey

'NOW.. compare the two files by opening both to Notepad
```


END

KUVAUS

Tämä komento lopettaa ohjelman. Välittömästi. Kaikki muistiin jäänyt media (objektit, kuvat, äänet jne.) tuhotaan automaattisesti. Ohjelma lopetetaan myös, kun koodi on ajettu loppuun, eikä uusiin koodiriveihin enää törmätä.

KÄYTTÖ

END

ESIMERKKI

ENDSEARCH

KUVAUS

Käytä tätä komentoa sen jälkeen kun olet etsinyt kaikki tarvittavat tiedostot käyttäen **FindFileä**.

Etsintä täytyy aloittaa komennolla **StartSearch**.

Älä unohda kutsua tätä komentoa, kun etsintä on valmis!

KÄYTTÖ

```
ENDSEARCH
```

Katso myös: [STARTSEARCH](#), [FINDFILE](#)

ESIMERKKI

```
'Change directory to Media
ChDir "Media"

'Prepare search
StartSearch

    'Scan all files and print all .BMP images
    count=0
    Repeat

        file$=FindFile()

        If file$="" Then Exit 'no more files

        If Lower(Right(file$,4))=".bmp" Then
            Print file$

            count=count+1
        EndIf

    Forever

'End search
EndSearch

Print "Found "+count+" files"

WaitKey
```

EOF

KUVAUS

Käytä tätä funktiota kertomaan, onko tiedosto jo luettu loppuun. Se palauttaa 1, jos tiedosto on lopussa, ja nolla, jos lukeminen jatkuu vielä.

KÄYTTÖ

EOF (tiedostomuuttuja)

- **tiedostomuuttuja** = Se muuttuja, mihin tiedosto avattiin.

ESIMERKKI

```
'make window bigger as otherwise the text won't fit
SCREEN 580,420,16,cwindowed

'open the file
f=OpenToRead("Media\example.txt")

textline=0

'print the whole file to the screen
While Not EOF(f)

    Text 0,textline,ReadLine(f)

    textline=textline+12

Wend

'show the screen
DrawScreen

'don't forget this!
CloseFile f

WaitKey
```

ERRORS

KUVAUS

Asettaa CoolBasicin virheilmoitukset päälle/pois. On lähinnä makuasia, haluatko ohjelman ilmoittavan *Memory Access Violation* vai jonkun kuvaavamman virheilmoituksen. Oletuksena virheilmoitukset on kytketty päälle. Se helpottaa virheen etsimistä.

Tyypillisin ilmoitus on ***komento failed***, missä *komento* on epäonnistuneen komennon nimi.



Kokeile esimerkkiä ensimmäisen rivin kanssa ja sitä ilman.

KÄYTTÖ

ERRORS päällä

- **päällä**

ON = virheet päällä (oletus)

OFF = virheet pois päältä (Memory Access Violation)

ESIMERKKI

```
'Try to remove this line
```

```
Errors OFF
```

```
DrawImage LoadImage("THISIMAGECANTBEFOUND.TGA"),0,0
```

ESCAPEKEY

KUVAUS

Palauttaa 1, jos ESC-näppäintä painetaan. Muutoin palautetaan nolla.

Muistathan kytkeä ensin SAFEEXIT-tominnon pois komentamalla ohjelmasi alussa **SAFEEXIT OFF**. Muuten ohjelmasi loppuu välittömästi, kun Esciä painetaan.

KÄYTTÖ

ESCAPEKEY ()

Katso myös: [SAFEEXIT](#)

ESIMERKKI

```
SAFEEXIT OFF 'disable escape auto-exit

AddText "Press escape..."

Repeat

    Text 30,30,"ESC: "+EscapeKey()

    DrawScreen

Until KeyHit(cbkeyreturn)
```

EXECUTE

KUVAUS

Tämä komento suorittaa toisen ohjelman.

Voit myös komentaa suorittamaan esim. **.txt** tai **.doc** -tiedostoja, jolloin ne avautuvat niiden omiin oletusohjelmiin.

Voit myös avata selaimen tai sähköpostiohjelman käyttämällä Internet- tai sähköposti-osoitetta:

Execute "http://www.coolbasic.com"

Execute "mailto:my.name@domain.com"

Jos tiedostopolussa on välilyöntejä, polku on esitettävä lainauksessa:

Execute Chr(34) + "C:\Program Files\CoolBasic\versions.html" + Chr(34)

Voit käynnistää Windowsin mukana tulleita ohjelmia useimmiten pelkällä tiedostonimellä, kuten esimerkissä.

KÄYTTÖ

EXECUTE tiedosto

- **tiedosto** = Merkkijono, joka määrittää suoritettavan tiedoston nimen (polkuineen).

ESIMERKKI

Execute "mspaint"

EXIT

KUVAUS

Rikkoutuu ulos mistä tahansa silmukasta, vaikka silmukka ei olisikaan vielä "valmis":

- For..Next
- For (each) ..Next
- Repeat..Until
- While..Wend

(Else)If ja Select -rakenteista ei voida poistua Exit:llä. Lisäksi funktioista ja aliohjelmista poistutaan **Return**-komennolla.

KÄYTTÖ

EXIT

ESIMERKKI

```
i=1  
  
Repeat  
  
    Print "Press RETURN to QUIT."  
  
    'If 10 000 iterations have passed break free  
    i=i+1  
    If i>10000 Then Exit  
  
Until KeyHit(cbkeyreturn)
```

TYPE ja FIELD

KUVAUS

Tämä koodilohko esittelee uuden TYYPIN ohjelmaasi. Siksi se kannattaa kirjoittaa ohjelman alkuun. Tyypit ovat kokoelmia, joihin voidaan luoda ja joista voidaan poistaa jäseniä. Kullakin jäsenellä on omat muuttujansa. Ne ovat ikään kuin itsenäisiä objekteja, jotka muodostavat listan.

Tyyppien avulla voit hallita suuria määriä vihollisia, ammuksia, efektejä ja objekteja helposti ja vaivattomasti.

TYYPPI-lohko lopetetaan komenttoon **END TYPE**. Älä unohda sitä! Tämän lohkon sisään taas kirjoitetaan **FIELD**-sanaa käyttäen kaikki muuttujat, jotka siis monistetaan automaattisesti jokaiselle jäsenelle.

Esimerkki kertoo kaiken tarpeellisen, mutta suosittelemme ehdottomasti, että luet myös [tämän](#).

KÄYTTÖ

```
TYPE nimi
  FIELD muuttuja1
  FIELD muuttuja2
  ...
END TYPE
```

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
Type BULLETS
  Field obj
End Type

' _ _ _ _ _
'MAIN PROGRAM
' _ _ _ _ _

FrameLimit 40

'load the soldier, turning detail 72
guy=LoadObject("Media\soldier.bmp",72)

'load the bullet (master-object) and hide it
bullet=LoadObject("Media\bullet.bmp")
ShowObject bullet,OFF

ClsColor cbdark

AddText "Arrows to turn, SPACE to shoot"

'MAIN LOOP
Repeat

  'Update turning
  If LeftKey() Then TurnObject guy,5
  If RightKey() Then TurnObject guy, -5

  'Shoot?
  If KeyDown(cbkeyspace) And reload=0

    newBullet.BULLETS= New(BULLETS)
    newBullet\obj=CloneObject(bullet)
    CloneObjectPosition newBullet\obj,guy
    CloneObjectOrientation newBullet\obj,guy
    'put the bullet at the end of the barrel
    MoveObject newBullet\obj,24

    reload=4
  EndIf

  'Update all BULLETS (move them forward)
  i=0
  For iBullet.BULLETS= Each BULLETS

    MoveObject iBullet\obj,6 'speed=6

    'delete bullet from collection if it
    'gets too far away
    If Distance2(iBullet\obj,guy)>200 Then
```



```
        DeleteObject iBullet\obj
        Delete iBullet
    EndIf
```

```
    i=i+1
```

```
Next iBullet
```

```
'Update reloading
```

```
If reload>0 Then reload=reload-1
```

```
Text 30,30,"Bullets to update: "+i
```

```
'Don't forget this :)
```

```
DrawScreen
```

```
Until EscapeKey()
```

FILEEXISTS

KUVAUS

Kertoo, onko tiedosto tai kansio olemassa. Yleensä kannattaa vielä tarkistaa onko kyseessä kansio vai tiedosto.

Palauttaa 1 jos on, muutoin nolla.

KÄYTTÖ

FILEEXISTS (tiedosto)

- **tiedosto** = Tiedostonimi (merkkijono).

Katso myös: [ISDIRECTORY](#)

ESIMERKKI

```
AddText "Input a filename"

Repeat
    k$=Input("::")
    DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
ClearText

isfile=FileExists(k$)

Text 10,30,"FILE EXISTS: "+isfile
DrawScreen

WaitKey
```

FILEOFFSET

KUVAUS

Kertoo tiedosto-osoittimen sijainnin.

Joka kerta kun tiedostosta luetaan tai tiedostoon kirjoitetaan, sen sisäinen tiedosto-osoitin muuttaa paikkaa (riippuen montako tavua tiedostoon kirjoitetaan). Kun tiedosto avataan, sen tiedosto-osoitin on aina alussa, nollassa (huom, ei yksi).

SeekFile -komennolla voit siirtää tiedosto-osoittimen paikkaa tiedostossa, ja asettaa seuraavan kohdan, johon tietoa kirjoitetaan tai tietoa luetaan. Huomaa, että laskuri lähtee nolasta, mikä on tiedoston ensimmäinen paikka.

Tiedosto on avattava **OpenToEdit**-funktioilla, jotta FileOffsetia voisi käyttää. Tulos on kokonaisluku.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

```
FILEOFFSET (tiedostomuuttuja)
```

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToEdit-funktioilla.

Katso myös: [SEEKFILE](#)

ESIMERKKI

```
f=OpenToWrite("Media\temp.dat")

Print "File position: "+FileOffset(f)

WriteInt f,1234

Print "File position: "+FileOffset(f)

WriteShort f,55

Print "File position: "+FileOffset(f)

WriteFloat f,PI

Print "File position: "+FileOffset(f)

CloseFile f

WaitKey
```

FILESIZE

KUVAUS

Kertoo tiedoston koon tavuina. Kansioista ei voi ottaa kokoa.

Tulos on kokonaisluku.

KÄYTTÖ

```
FILEEXISTS (tiedosto)
```

- **tiedosto** = Tiedostonimi (merkkijono).

ESIMERKKI

```
'Scan all files in folder 'Media'
ChDir "Media"

StartSearch

  Repeat
    file$=FindFile()

    If file$="" Then Exit 'search has ended

    fsize=FileSize(file$) 'figure out the file size

    Print file$+", "+fsize+" bytes" 'print info
  Forever

EndSearch

WaitKey
```

FINDFILE

KUVAUS

Etsii tiedostoja nykyisestä kansioista. Käytä tätä funktiota koko kansion läpikäyntiin. FindFile palauttaa seuraavan tiedoston nimen. Jos etsintä on suoritettu loppuun, palautetaan tyhjä merkkijono. Myös kansiot palautetaan. Voit halutessasi tarkistaa onko haettu nimi kansio vai tiedosto **IsDirectory**-kutsulla.

Et voi selata tiedostolistaa taaksepäin. FindFile palauttaa aina *seuraavan* tiedoston.

Etsintä täytyy aloittaa komennolla **StartSearch**, ja lopettaa komennolla **EndSearch**.

KÄYTTÖ

```
FINDFILE ()
```

Katso myös: [STARTSEARCH](#), [ENDSEARCH](#)

ESIMERKKI

```
'Change directory to Media
ChDir "Media"

'Prepare search
StartSearch

'Scan all files and print all .BMP images
count=0
Repeat

    file$=FindFile()

    If file$="" Then Exit 'no more files

    If Lower(Right(file$,4))=".bmp" Then
        Print file$

        count=count+1
    EndIf

Forever

'End search
EndSearch

Print "Found "+count+" files"

WaitKey
```

FIRST

KUVAUS

Tämä funktio kertoo kokoelman ensimmäisen jäsenen.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

FIRST (tyyppi_nimi)

- tyyppi_nimi = TYPE-kokoelman nimi.

Katso myös: [LAST](#), [BEFORE](#), [AFTER](#)

ESIMERKKI

```
Type test
  Field value As Float
End Type

'First create 10 elements
For i=1 To 10
  t.test = New(test)

  t\value=Rnd(1)
Next i

'Print them all
Repeat

  t.test = First(test)
  i=10
  While t<> NULL
    Text 10,i,t\value
    i=i+16
    t = After(t)
  Wend

  DrawScreen

Until EscapeKey()
```

FLIP

KUVAUS

Kääntää merkkijonon ylösalaisin (lopusta alkuun kirjoitetuksi).

KÄYTTÖ

FLIP (jono)

- **jono** = Merkkijono, joka käännetään.

ESIMERKKI

```
Repeat
  k$=Input("Type someting: ")

  Text 0,50,"Inverse text: "+Flip(k$)
  DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
```

FLOAT

KUVAUS

Muuntaa minkä tahansa arvon desimaaliluvuksi.

Jotkin komennot ja funktiot vaativat desimaaliluvun. Käytä tätä funktiota muuntamaan esim. kokonaislukuja desimaalimuotoon, jonka jälkeen se kelpaa parametriksi.

Voit käyttää Float-funktiota muuttamaan merkkijonon desimaaliluvuksi. Muunnos kuitenkin pysähtyy epäkelvollisiin merkkeihin:

- `Int ( "57" ) -> 57.0`
- `Int ( "-6" ) -> -6.0`
- `Int ( "22yo" ) -> 22.0`
- `Int ( "Hello World" ) -> 0.0`

Tämän funktion yleisin käyttötarkoitus lienee pakottaa luvut desimaalimuotoon laskutoimituksen onnistumiseksi:

- `18 / 5 = 3`
- `Float (18) / 5 = 3.6`

KÄYTTÖ

FLOAT (arvo)

- **arvo** = Kokonaisluku, desimaaliluku tai merkkijono.

Katso myös: [INT](#), [STR](#)

ESIMERKKI

```
For i=1 To 5

    value=Rand(-10,10)

    Text 0, i*12, "original: "+value
    Text 200, i*12, "-> FLOAT -> "+Float(value)

Next i

DrawScreen
WaitKey
```


FOR

KUVAUS

Muodostaa silmukan, jota toistetaan tietty määrä kertoja. Silmukkaan liitetään muuttuja, joka käy läpi kaikki arvot alkuarvon ja lopetusarvon väliltä.

Rakenne on ihanteellinen, kun halutaan lisätä monen muuttujan arvoa tai hallita esim. kaikkia taulukon elementtejä. Jos haluat selata TYYPIN jäsenet läpi, käytä erikoismuotoitua For-lauseketta (**Each**-komento). Lisätietoja myöhemmin tässä oppaassa.

Voit valinnaisesti määrittää myös hyppyvälin käyttämällä **STEP**-sanaa. Tämän arvon on oltava vakio, eli positiivinen tai negatiivinen luku. Voit hyppiä esimerkiksi 2 askelta kerralla, tai selata lukuja alaspäin kohti nollaa miinusmerkkisellä hyppyvälillä.

Silmukasta voidaan rikkoutua kesken kaiken pois komennolla **Exit**.

Silmukka päätetään komenttoon NEXT. Pääsilmukaksi For...Next -rakenteesta ei ole.

Esimerkki valkaisee asiaa.

KÄYTTÖ

```
FOR muuttuja = alkuarvo TO loppuarvo [STEP hyppyväli]
    'koodia
NEXT sama_muuttuja
```

Step-arvo on valinnainen, ja se on oletuksellisesti yksi.

Katso myös: [TOINEN SELITYS](#), [FOR ... EACH](#)

ESIMERKKI

```
SCREEN 300,500

Print "ITERATE 1 -> 10"
Print

Print "-- STEP BY 1 -----"

' Tulosta arvot 1-10
For t= 1 To 10
    Print t
Next t

Print "-- STEP BY 2 -----"

' Tulosta arvot 1,3,5,7,9
For t= 1 To 10 Step 2
    Print t
Next t

Print "-- VALUES 1 -> Random -----"

' Tulosta arvot lähtien 1- joku satunnainen luku väliltä (2,10)
For t= 1 To Rand(2,10) Step 1
    Print t
Next t

Print "-- VALUES 1 -> 4 -----"

' Tulostaa arvot 1-4
For t= 1 To 6
    Print t
    If t> 3 Then Exit
Next t

WaitKey
```

FPS

KUVAUS

Kertoo kuinka monta kertaa ruutu ehdittiin piirtää sekunnissa. Eli oikeastaan tämä kertoo pelin toimintanopeuden.

Palautettu luku on kokonaisluku.

KÄYTTÖ

```
FPS ()
```

ESIMERKKI

'Try this example with and without the line below

```
FrameLimit 30
```

```
one=1
```

```
Repeat
```

```
    'move the ball
```

```
    x=x+one
```

```
    If x<0 Or x>400 Then one= -one
```

```
    'draw the ball
```

```
    Color cbred
```

```
    Circle x-20,180,40
```

```
    Color cbwhite
```

```
    Text 0,0,FPS() 'print the current FPS
```

```
    DrawScreen
```

```
Until EscapeKey()
```

FRAMELIMIT

KUVAUS

Tämä komento rajoittaa pelisi suorituskyyä niin, että se ei pyöri liian nopeasti tehokkailla koneilla. Arvo kannattaa asettaa niin alas kuin mahdollista ilman, että peli nykii häiritsevästi. Sopivia arvoja ovat mm. 30,35,40 ja 60.

Suorituskyy mitataan peleissä FPS:llä (*frames per second*). Se mittaa suoraan kuinka monta kertaa näyttö piirretään yhden sekunnin aikana. Suurempi FPS mahdollistaa pelin pyöriä sulavasti, mutta heikommät laitteistot eivät välttämättä pysy perässä, mikä taas ilmenee pelin hidastumisena.

Kokeile esimerkkiä komennon kanssa ja ilman.

KÄYTTÖ

FRAMELIMIT nopeus

- **nopeus** = Kokonaisluku, joka määrittää kuinka monta kertaa ruutu piirretään sekunnissa.

ESIMERKKI

'Try this example with and without the line below

FrameLimit 30

one=1

Repeat

 'move the ball

 x=x+one

If x<0 **Or** x>400 **Then** one= -one

 'draw the ball

Color cbred

Circle x-20,180,40

Color cbwhite

Text 0,0,**FPS**() 'print the current FPS

DrawScreen

Until **EscapeKey**()

FUNCTION

KUVAUS

Funktio on koodilohko, joka sisältää tehtävän, jota tarvitaan toistuvasti tai useassa kohtaa koodiasi. Funktio on kuin aliohjelma, mutta se palauttaa arvon.

Funktiot ovat itsenäisiä pääohjelmasta, ja ne suoritetaan vain ja ainoastaan kun niitä kutsutaan. Niillä on omat muuttujat (vaikka pääohjelmassasi olisi samanniminen, se on silti eri) ja omat kirjanmerkit (**Goto**). Globaalit muuttujat, taulukot sekä vakiot kuitenkin näkyvät myös funktioissa. Määrittääksesi muuttujan globaaliksi, katso **GLOBAL**-komento. Gosub-aliohjelmat eivät ole sallittuja funktioiden sisällä.

Voit välittää muuttujia ja arvoja funktiolle. Funktio palauttaa arvoja **RETURN-käskyn** avulla. Jotta voisit palauttaa desimaalilukuja tai merkkijonoja, varmista että funktion nimen perässä on '#'- tai '\$'-merkki.

Ideaalinen käyttötarkoitus olisi jakaa koodisi helposti ylläpidettäviin lohkoihin. Näin sinun ei esimerkiksi tarvitse muuttaa koodia kuin yhdestä paikasta, jos olet lisäämässä siihen ominaisuuksia tai korjaamassa bugeja. Voit esimerkiksi luoda funktion, joka järjestää jonkun taulukon solut suuruusjärjestykseen.

Lohko lopetetaan **END FUNCTION** -komennolla. Kun tämä komento saavutetaan, palaa ohjelma välittömästi siihen kohtaan, mistä funktiota kutsuttiin.

Vaikka funktiot voivat aluksi hieman hämmentää, kannattaa ne silti opetella tarkoin. Jossain vaiheessa kuitenkin ymmärrät, että funktiot eivät ole muuta kuin pikkuohjelmia oman ohjelmasi sisällä. Katso esimerkki.

Voi myös olla näppärää korvata jonkun CoolBasicin omista funktioista lisäten siihen ylimääräisiä virheentarkistuksia tms. Omat funktiot eivät saa olla samannimisiä kuin CoolBasicin jo olemassa olevat funktiot.

Rekursio onnistuu CoolBasicissa. Rekursio tarkoittaa sitä, että funktio kutsuu itseään ja käyttää hyväkseen omaa paluuarvoaan jatkaakseen. Lisätietoja sanakirjasta.

Parametriluettelo voi sisältää myös valinnaisia parametreja samaan tapaan kuin eräät CoolBasicin funktiot. Valinnainen parametri esitetään kirjoittamalla muuttujan nimen perään sijoitus, esim. "kalat=1" (oletuksellisesti kalat=1). Valinnaisten parametrien jälkeen ei voi esiintyä normaaleja parametreja.

KÄYTTÖ

```
FUNCTION nimi ([parametriluettelo])
```

Parametriluettelo on valinnainen ja ne erotetaan pilkulla. Parametriluettelo sisältää listan muuttujista, joihin arvot asetetaan funktiota kutsuttaessa.

Katso myös: [RETURN](#)

ESIMERKKI

Repeat

```
c=Input("Give tempereture in Celcius degrees? ")
```

DrawScreen

Until KeyHit(28) 'loop until RETURN-key

CloseInput

'-----

Repeat

```
Text 0,0,c+" Celcius equals to "+Fahrenheit(c)+" Fahrenheit"
```

DrawScreen

Until EscapeKey()

'-----

'FUNCTIONS

'-----

'This function converts Celcius tempereature into

'Fahrenheit temperature

Function Fahrenheit(degrees)

```
fah=(degrees-32)/1.8
```

Return fah

End Function

GETANGLE

KUVAUS

Kertoo kahden pisteen välisen kulman. Käytä tätä komentoa laskemaan missä kulmassa piste 2 on pisteestä 1 katsottuna.

Palautettu kulma on desimaaliluku väliltä 0-360.

Esimerkki selventää asiaa.

KÄYTTÖ

```
GETANGLE (x1, y1, x2, y2)
```

- **x1** = Ensimmäisen pisteen X-koordinaatti.
- **y1** = Ensimmäisen pisteen Y-koordinaatti.
- **x2** = Toisen pisteen X-koordinaatti.
- **y2** = Toisen pisteen Y-koordinaatti.

Katso myös: [GETANGLE2](#)

ESIMERKKI

Repeat

```
Line 200,150,350,150
```

```
Line 200,150,MouseX(),MouseY()
```

```
Text 0,0,GetAngle(200,150,MouseX(),MouseY())
```

```
DrawScreen
```

```
Until EscapeKey()
```

Objektit : funktiot

GETANGLE2

KUVAUS

Kertoo missä kulmassa objekti 2 on objektiin 1 nähden. Eli miten objekti 1 näkee objekti 2:en.

Tulos on kokonaisluku väliltä 0-360.

KÄYTTÖ

```
GETANGLE2 (obj1, obj2)
```

- **obj1** = Objekti, jonka näkökulmasta tarkastellaan.
- **obj2** = Kohdeobjekti.

Katso myös: [GETANGLE](#), [DISTANCE2](#)

ESIMERKKI

```
DrawToWorld ON
FrameLimit 40

obj1=LoadObject("Media\guy.bmp")
obj2=LoadObject("Media\guy.bmp")

AddText "The code updates every 5 seconds..."

Repeat

    'Perform this every 5 seconds
    If TIMER()>moment+5000 Then

        PositionObject obj1,Rand( -180,180),Rand( -130,130)
        PositionObject obj2,Rand( -180,180),Rand( -130,130)

        'find out the angle between obj1->obj2
        angle=GetAngle2(obj1,obj2)

        moment=TIMER()
    EndIf

    Line ObjectX(obj1),ObjectY(obj1),ObjectX(obj1)+Cos(angle)*40,ObjectY(obj1)+Sin(angle)*40
    Text 0,30,angle

    DrawScreen

Until EscapeKey()
```

GETCOLLISION

KUVAUS

Palauttaa tietyn törmäyksen toisen osapuolen. Pääset jokaiseen törmäykseen erikseen käsiksi. (1 -> **CountCollisions**).

Käytä tätä tarkistamaan KENEN kanssa törmättiin.

KÄYTTÖ

GETCOLLISION (objektimuuttuja, indeksi)

- **objektimuuttuja** = Objekti, jonka törmäykset tarkistetaan.
- **indeksi** = Kokonaisluku ykkösestä eteenpäin. Törmäyksen järjestysnumero.

Katso myös: [COLLISIONX](#), [COLLISIONY](#), [COUNTCOLLISIONS](#), [COLLISIONANGLE](#)

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

'Load the map
map=LoadMap("Media\cdm2.til", "Media\tileset.bmp")

'Load player
soldier=LoadObject("Media\soldier.bmp",72)
ObjectRange soldier,32

'Load dumb immobile bot
bot=LoadObject("Media\soldier.bmp")
ObjectRange bot,32
PositionObject bot,50,200

'setup some collisions
SetupCollision soldier,bot,2,2,2
SetupCollision soldier,map,1,4,2

Repeat

    'Update controls
    If LeftKey() Then TurnObject soldier,5
    If RightKey() Then TurnObject soldier, -5
    If UpKey() Then MoveObject soldier,3.5
    If DownKey() Then MoveObject soldier, -2

    'Camera follow
    CloneCameraPosition soldier

    'Enable additional graphics
    DrawGame

    collisions=CountCollisions(soldier)

    'iterate through all collisions
    For i=1 To collisions
        Color cbwhite
        Text 0,(i-1)*12,"angle: "+CollisionAngle(soldier,i)+" object: "+GetCollision(soldier,i)

        'draw the collision-ball
        Color cbred
        Circle CollisionX(soldier,i) -5,CollisionY(soldier,i)+5,10
    Next i

    'render SCREEN
    DrawScreen

Until EscapeKey()
```

GETEXENAME

KUVAUS

Kertoo ohjelman suoritushakemiston ja nimen.

Palauttaa merkkijonon.

KÄYTTÖ

GETEXENAME ()

ESIMERKKI

Print GetEXEName ()

WaitKey

GETKEY

KUVAUS

Tämä funktio kertoo, mitä näppäintä on viimeksi painettu. Toisin kuin KeyUp ja KeyHit, GetKey palauttaa painetun näppäimen [ASCII-arvon](#). Lisäksi GetKey lukee näppäimistöä aktiivisesti, siihen ei tarvita UpdateGamea väliin.

Katso esimerkki.

KÄYTTÖ

```
GETKEY ()
```

Katso myös: [KEYDOWN](#), [KEYHIT](#)

ESIMERKKI

```
AddText "Press keyboard"
```

```
Repeat
```

```
    a=GetKey()
```

```
    If a Then b=a
```

```
    Text 20,20,"Pressed ASCII: "+b
```

```
    Text 20,35,"Pressed character: "+Chr(b)
```

```
    DrawScreen
```

```
Until EscapeKey()
```

GETMAP

KUVAUS

Käytä tätä funktiota lukeaksesi karttaa. Koordinaatit annetaan pelimaailman pikseleinä, joten (0,0) ei välttämättä ole kartan vasen yläkulma.

Jos haluat esittää koordinaatit tileinä, käytä mieluummin **GetMap2**:sta.

Palautettu luku on kokonaisluku, ja kahdessa ensimmäisessä kerroksessa se kertoo siinä kohdassa olevan tilen järjestysnumeron. Törmäyskerroksessa 1 (seinä) tai nolla (tyhjä).

KÄYTTÖ

GETMAP (kerros, x, y)

- **kerros**

0 = taustakerros

1 = päälliskerros

2 = törmäyskerros

3 = datakerros

- **x** = Kokonaisluku. X-koordinatti pikseleinä.

- **y** = Kokonaisluku. Y-koordinatti pikseleinä.

Katso myös: [GETMAP2](#)

ESIMERKKI

```
FrameLimit 40 'limit game performance

'Load map and animate it
map= LoadMap("Media\cdm2.til","Media\tileset.bmp")
PlayObject map,0,0,0.5

cursor=LoadImage("Media\cursor.bmp")
ShowMouse cursor

AddText "Arrows to scroll..."
AddText "Pinpoint tiles with the mouse"

Repeat

  'Move the camera
  If UpKey() Then MoveCamera 0,2
  If DownKey() Then MoveCamera 0, -2
  If LeftKey() Then MoveCamera-2
  If RightKey() Then MoveCamera 2

  'Otherwise we wouldn't be able to see the text
  DrawGame

  Color cbyellow
  'Calculate the tile under mouse
  '(200 is half the SCREENWIDTH, 150 is half the SCREENHEIGHT)
  tile=GetMap(0,MouseWX(), MouseWY())
  Text 60,60,"Tile under mouse: "+tile

  DrawScreen

Forever
```

GETMAP2

KUVAUS

Käytä tätä funktiota lukeaksesi karttaa. Koordinaatit annetaan (1,1) on kartan vasen yläkulma. Ota kartan kokosuhteet (tileinä) selville funktioilla **MapWidth** ja **MapHeight**.

Tämä funktio on ihanteellinen, kun halutaan skannata vaikka koko kartta läpi, ja etsiä siitä tiettyjä tilejä. Esim. data-kerrokseen koodattuja vihollisten sijainteja tai teleportteja.

Jos haluat esittää koordinaatit pikseleissä, käytä mieluummin funktiota **GetMap**.

Palautettu luku on kokonaisluku, ja kahdessa ensimmäisessä kerroksessa se kertoo siinä kohdassa olevan tilen järjestysnumeron. Törmäyskerroksessa 1 (seinä) tai nolla (tyhjä).

KÄYTTÖ

```
GETMAP2 (kerros, x, y)
```

- **kerros**

- **0** = taustakerros

- **1** = päälliskerros

- **2** = törmäyskerros

- **3** = datakerros

- **x** = Kokonaisluku. X-koordinatti tileinä.

- **y** = Kokonaisluku. Y-koordinatti tileinä.

Katso myös: [GETMAP](#)

ESIMERKKI

```
FrameLimit 40 'limit game performance
```

```
'Load map and animate it  
map= LoadMap ("Media\cdm2.til", "Media\tileset.bmp")
```

```
AddText "Scanning map..."
```

```
DrawScreen
```

```
i=0  
For y=1 To MapWidth()  
For x=1 To MapWidth()  
If GetMap2 (0,x,y)=137 Then i=i+1  
Next x  
Next y
```

```
'Remove map from memory  
DeleteObject map
```

```
AddText "DONE"
```

```
AddText ""
```

```
AddText "Tile 137 occurred "+i+" times."
```

```
DrawScreen
```

```
WaitKey
```

GETMOUSE

KUVAUS

Palauttaa alhaalla olevan hiiren nappulan.

Toisin kuin *MouseHit* ja *MouseUp* - tämä funktio lukee hiirtä aktiivisesti, ja tila päivitetään joka kerta kun sitä kutsutaan. Jos sinun täytyy odottaa pääloopissa jokin aika ja tarkastaa hiiren tila sen jälkeen, käytä mielummin tätä funktiota.

Kokonaisluku palautetaan seuraavasti:

- **1** = vasen nappula
- **2** = oikea nappula
- **3** = keskimäinen nappula

KÄYTTÖ

```
GETMOUSE ()
```

Katso myös: [MOUSEDOWN](#), [MOUSEUP](#), [MOUSEHIT](#)

ESIMERKKI

```
AddText "Press mouse buttons"

Repeat

    button=GetMouse()

    If button Then record=button

    Text 20,20,"GetMouse: "+record

    DrawScreen

Until EscapeKey()
```

GETPIXEL

KUVAUS

Lukee (32-bittisen) pikselin nykyisestä ei-lukitusta puskurista.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

```
GETPIXEL (x, y, [puskuri])
```

- **x** = Kokonaisluku, pikselin X-sijainti.
- **y** = Kokonaisluku, pikselin Y-sijainti.
- **puskuri** (**valinnainen**) = Puskuriosoitin. Kts. funktiot SCREEN() ja IMAGE(). Oletuksena nykyinen puskuri.

Katso myös: [PUTPIXEL](#), [SCREEN](#), [IMAGE](#)

ESIMERKKI

```
SCREEN 640,480
```

```
AddText "Press any key to copy..."
```

```
'First make a gradient
```

```
For i=0 To 240
```

```
    Color 0,i+15,0
```

```
    Line 0,i,640,i
```

```
Next i
```

```
DrawScreen OFF
```

```
WaitKey
```

```
'make 154 481 iterations, copy each pixel
```

```
For y=0 To 240
```

```
For x=0 To 640
```

```
    PutPixel x,480-y,GetPixel(x, y)
```

```
Next x
```

```
Next y
```

```
DrawScreen
```

```
WaitKey
```

GETPIXEL2

KUVAUS

Lukee (32-bittisen) pikselin nykyisestä puskurista. Suorituskyky on nopeampi kuin **GetPixel**-funktiossa.

Puskurin täytyy olla lukittu. Tämä tapahtuu komennolla **Lock**. Pikselikomennot lukittuun puskuriin ovat erittäin nopeita.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

```
GETPIXEL2 (x, y, [puskuri])
```

- **x** = Kokonaisluku, pikselin X-sijainti.
- **y** = Kokonaisluku, pikselin Y-sijainti.
- **puskuri** (**valinnainen**) = Puskuriosoitin. Kts. funktiot **SCREEN()** ja **IMAGE()**. Oletuksena nykyinen puskuri.

Katso myös: [LOCK](#), [PUTPIXEL2](#), [SCREEN](#), [IMAGE](#)

ESIMERKKI

```
SCREEN 640,480

AddText "Press any key to copy..."

'First make a gradient
For i=0 To 240
    Color 0,i+15,0
    Line 0,i,640,i
Next i

DrawScreen OFF

WaitKey

'Lock screen
Lock

'make 154 481 iterations, copy each pixel
For y=0 To 240
For x=0 To 640
    PutPixel2 x,480-y,GetPixel2(x, y)
Next x
Next y

'Unlock screen
Unlock

DrawScreen

WaitKey
```

GETRGB

KUVAUS

Palauttaa nykyisen piirtovärin punaisen, vihreän tai sinisen komponentin. Eli RGB-arvot. Jos et tiedä mitä tämä tarkoittaa, et tarvitse tätä funktiota :)

KÄYTTÖ

GETRGB (kanava)

- kanava = RED, GREEN tai BLUE.

Katso myös: [PICKCOLOR](#)

ESIMERKKI

```
'THIS PROGRAM WILL CHALLENGE MS PAINT

FrameLimit 40

'load palette
palette= LoadImage("Media\palette.bmp")

'Make canvas
canvas= MakeImage(300,300)
ClsColor cbwhite
DrawToImage canvas
  Cls
DrawToScreen

ClsColor cbwhite
Color cbbblack

'add info
Locate 110,10
AddText "Pick a color via palette right click"
AddText "Draw using the mouse"

ClsColor cbwhite

Repeat

  'draw palette
  DrawImage palette,0,0,0,OFF

  'draw preview
  Box 20,220,20,20

  Text 20,260,GetRGB(RED)
  Text 20,270,GetRGB(GREEN)
  Text 20,280,GetRGB(BLUE)

  'pick colour
  If MouseDown(2)
    If MouseX()<100 And MouseY()<200 Then
      PickColor MouseX(),MouseY()
    EndIf
  EndIf

  'draw
  If MouseDown(1)
    If MouseX()>100 Then
      DrawToImage canvas
        Line oldx-100,oldy,MouseX() -100,MouseY()
        oldx=MouseX()
        oldy=MouseY()
      DrawToScreen
    EndIf
  EndIf

  'draw canvas
  DrawImage canvas,100,0,0,OFF

DrawScreen
```

Until **EscapeKey**()

GETWORD

KUVAUS

Kaivaa tekstistä tietyn sanan. Voit esimerkiksi kaivaa kaikki sanat erikseen.

Token-ominaisuuksien vuoksi myös "tyhjät" sanat poimitaan. Sana on tyhjä, jos törmätään kahteen peräkkäiseen välilyöntiin.

Välimerkin ei ole pakko olla välilyönti. Voit määrittää myös toisen erottimen valinnaisella parametrilla (kts. alas).

KÄYTTÖ

```
GETWORD (jono, monesko, [erotin])
```

- **jono** = Merkkijono, josta sanat lasketaan.
- **monesko** = Monesko sana kaivetaan. Kokonaisluku.
- **erotin** (**valinnainen**) = Yksi merkki. Esim. **"*"**. Oletuksena välilyönti. Tätä parametria ei ole pakko kirjoittaa.

Katso myös: [COUNTWORDS](#)

ESIMERKKI

```
mytext$="one two three four five six"

'original text
Text 0,0,mytext$

'grab all words
For i=1 To CountWords(mytext$)
    Text 10, 20+i*10, GetWord(mytext$,i)
Next i

DrawScreen

WaitKey
```

GFXMODEEXISTS

KUVAUS

Kertoo onko haluttu kokonäytön tila saatavilla. Kaikki näytönohjaimet eivät tue kaikkia resoluutiota, ja ohjelmoija voi tarkastaa tukeeko pelaajan näytönohjain haluttua näyttötilaa. Jos ei tue, voidaan kokeilla toista. Näin estetään helposti pelin toimimattomuus toisilla koneilla. Esimerkiksi 15 tuuman littunäytöt eivät yleensä pysty parempaan resoluutioon kuin 1024*768.

Palauttaa ykkösen, jos näyttötila toimii, ja nollan jos näyttötila ei ole saatavilla.

KÄYTTÖ

GFXMODEEXISTS (leveys, korkeus, värisyvyys)

- **leveys** = Kokonaisluku. Näyttötilan leveys pikseleinä.
- **korkeus** = Kokonaisluku. Näyttötilan korkeus pikseleinä.
- **värisyvyys** = Kokonaisluku. Näyttötilan värisyvyys, yleensä 16 tai 32.

ESIMERKKI

```
If GFXModeExists(640,480,16) Then
    Print "640*480 (16-bit) AVAILABLE"
Else
    Print "640*480 (16-bit) NOT AVAILABLE"
EndIf

If GFXModeExists(1024,768,32) Then
    Print "1024*768 (32-bit) AVAILABLE"
Else
    Print "1024*768 (32-bit) NOT AVAILABLE"
EndIf

If GFXModeExists(2000,2000,8) Then
    Print "2000*2000 (8-bit) AVAILABLE"
Else
    Print "2000*2000 (8-bit) NOT AVAILABLE"
EndIf

WaitKey
```

Objektit : **komennot**

GHOSTOBJECT

KUVAUS

Voit asettaa objektin (ei koske lattiaobjekteja tai tilekarttoja) läpinäkyväksi. Toimii partikkelilähteiden ja normaalien objektien kanssa. Tekniikka kutsutaan myös nimellä alpha-blending eli alfa-blendaus, blend. Objektin mustaa (RGB 0,0,0) väriä ei piirretä. Kaikki muut värit lasketaan alla olevan taustan mukaan läpinäkyviksi. Arvo nolla piilottaa objektin kokonaan (silti objekti osallistuu törmäyksen tunnistukseen). Arvo 100 on yhtä kuin täysin näkyvä objekti. Arvo 50 taas on puolinäkyvä. Voit itse päättää läpinäkyvyyden tason asteikolla 1-99.

VAROITUS! läpinäkyvien objektien piirtäminen on melkoisen raskas urakka nykylaitteillekin ilman laitteistokiihdytystä (3D). Sen liiallinen käyttö saattaa vaikuttaa dramaattisesti pelin suorituskykyyn. Käytä ominaisuutta säästeliäästi ja mieluiten lähinnä pienikokoisille objekteille, kuten ammuksille tai liekeille ym.

KÄYTTÖ

GHOSTOBJECT objektimuuttuja, läpinäkyvyys

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **läpinäkyvyys** = Arvo 1-99. Pienet arvot tekevät objektista vaikeasti nähtävän, ja suuremmat näkyvämmän.

Katso myös: [DRAWGHOSTIMAGE](#)

ESIMERKKI

```
'load a map
map= LoadObject("Media\map.bmp")
cow= LoadObject("Media\cow.bmp",72)

alpha=100

AddText "Move the cow with arrows"
AddText "A/Z to alter transparency"

Repeat

    'update controls
    TranslateObject cow,RightKey()-LeftKey(),UpKey()-DownKey()

    TurnObject cow,5

    'alter transparency
    alpha= alpha+(KeyDown(cbkeya) -KeyDown(cbkeyz))
    GhostObject cow,alpha

    DrawGame 'draw all objects so the text can be rendered visible
    Text 0,60,FPS()+" "+alpha 'show how fast the game is able to render
    DrawScreen()

Forever
```

GLOBAL

KUVAUS

Tekee muuttujasta globaalin. Globaalit muuttujat säilyttävät arvonsa kaikkialla ohjelmassa, myös funktioissa. Niinpä globaalien muuttujien arvoja voidaan muuttaa tai käyttää hyväkseen kaikkialla ohjelmassa. Aloittelijoille hyvä valinta.

Kun lataat esimerkiksi kuvia, ääniä ja tilekarttoja, haluat luultavasti säilyttää pääsyn niihin. Jos olet vaikka ladannut pääohjelmassa kuvan muuttujalle "logo" (logo=LoadImage("kuva.bmp")) ja haluat, että funktio PiirraKaikki() piirtää kaikki kuvat ohjelmassasi, täytyy sinun määrittää muuttuja "logo" globaaliksi, koska muutoin se nollaantuu funktioon mentäessä. Voit kokeilla asettamalla esimerkissä rivin "Global logo" kommentiksi.

Voit esitellä useamman muuttujan samassa käskyssä. Voit myös käyttää **As**-sanaa esittelemään muuttujan tyyppin, esim. **Global i As String**. Tyyppeinä voivat olla Integer, Float, String, Byte ja Short. Lisää manuaalin kohdasta "Tietotyytit".

KÄYTTÖ

```
GLOBAL muuttujal, [muuttuja2], [muuttuja3], [muuttuja4], ...
```

ESIMERKKI

```
'Comment this and see what happens...
```

```
Global logo
```

```
'load the image
```

```
logo=LoadImage ("media/pacman.bmp")
```

```
'Call function
```

```
drawimages()
```

```
'UpdateScreen
```

```
DrawScreen
```

```
WaitKey
```

```
'This function draws all images
```

```
Function drawimages()
```

```
    DrawImage logo,0,0
```

```
End Function
```

GOSUB

KUVAUS

Hyppää toiseen kohtaan koodissa. Toisin kuin Goto, pääset myöhemmin palaamaan tähän hyppykohtaan **RETURN**-kutsulla.

Gosub-kirjanmerkkejä kutsutaan aliohjelmiksi. Aliohjelmat päättyvät aina RETURN-komentoon, joka palauttaa ohjelman sinne, mistä aliohjelmaan hypättiin. Aliohjelmat ovat siis "*käy-tekemässä-jotain-ja-palaa-takaisin*" -rakenteita. **ÄLÄ UNOHDA RETURN-KOMENTOA!** Sen puuttuminen voi kaataa ohjelman tai aiheuttaa outoja toimintahäiriöitä itse pelissä.

KÄYTTÖ

```
GOSUB kirjanmerkki
```

Katso myös: [RETURN](#), [GOTO](#)

ESIMERKKI

```
x=200: y=150
```

```
AddText "Guide the cube via arrows"
```

```
'This is the main program. See how short it is
```

```
Repeat
```

```
    GoSub UpdateControls
```

```
    GoSub DrawAll
```

```
    DrawScreen
```

```
Until EscapeKey()
```

```
'End program so we can't accidentally
```

```
'access the sub-programs
```

```
End
```

```
'-----
```

```
UpdateControls:
```

```
    If RightKey() Then x=x+1
```

```
    If LeftKey() Then x=x-1
```

```
    If DownKey() Then y=y+1
```

```
    If UpKey() Then y=y-1
```

```
Return
```

```
'-----
```

```
DrawAll:
```

```
    Box x-20,y-20,40,40,OFF
```

```
Return
```

GOTO

KUVAUS

Hyppää toiseen kohtaan koodissa. Pysyvästi. Et pääse takaisin samaan paikkaan ilman uutta Goto-käskyä.

Goto siirtää ohjelman suorituksen kirjanmerkiksi (*label*) sanottuun paikkaan, ja jatkaa sieltä. Voit esimerkiksi hypätä silmukasta ulos tai siirtyä tekstiseikkailussa toiseen paikkaan (kts. Select-rakenteen esimerkki).

Kirjanmerkki on sana lähdekoodissa, jonka perään on kirjoitettu kaksoispiste. Kirjanmerkki tulee esittää yksin omalla rivillään. Kirjanmerkin nimeen pätevät samat säännöt kuin muuttujien nimiin.

Kirjanmerkki voi olla ylempänä tai alempana kuin Goto-kutsu, toisin sanoen missä vain. Paitsi funktiossa. Et voi hypätä Goton avulla funktioon tai funktiosta pois. Funktiot ovat pieniä ohjelmia ohjelmasi sisällä, ja niillä on omat kirjanmerkkinsä, jotka eivät ole millään tavoin sidoksissa toisiin funktioihin tai pääohjelmaan. Siksi ohjelmassasi voi olla saman nimisiä kirjanmerkkejä, kunhan ne eivät ole samassa funktiossa / pääohjelmassa.

Goto ei hirveästi ole joidenkin ohjelmajien suosiossa, koska se kuulemma sekoittaa ohjelmarakenteen ja sekavoittaa koodin lukemista. Joissakin tilanteissa Goto taas on erittäin kätevä, ja paljon mukavampi ratkaisu, kun silmukan suunnittelu niin, että siitä voidaan lähteä jonnekin.

KÄYTTÖ

```
GOTO kirjanmerkki
```

Katso myös: [GOSUB](#)

ESIMERKKI

```
Print "MOI"
GoTo yli

Print "Tätä riviä ei ikinä tulostu."

yli:
Print "Tänne hypättiin!"

WaitKey
```

GOTOSAVEDLOCATION

KUVAUS

Tämä komento siirtää pelin suorituksen täsmälleen samaan paikkaan, mihin se jäi kun peli tallennettiin. Käytä tätä komentoa LoadProgramin **jälkeen**. Yleensä vielä senkin jälkeen, kun kaikki kuvat sun muut on ladattu uudelleen.
Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

GOTOSAVEDLOCATION

Katso myös: [LOADPROGRAM](#)

ESIMERKKI

```
'Save game
foo()

Print "Program continues..."

'LoadGame
LoadProgram "savegame.sav"
GotoSavedLocation

Print "Program ended"

WaitKey

Function foo()
    SaveProgram "savegame.sav"
    'will continue from here after loading...
    Print "Game Saved..."
End Function
```

HEX

KUVAUS

Palauttaa lukuarvosta HEX-muotoisen merkkijonon.

Jos et tiedä mikä HEX on, et tarvitse tätä komentoa :).

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

HEX (luku)

- **luku** = Kokonaisluku.

ESIMERKKI

```
Repeat
  num=Input("Give a number: ")

  Text 0,50,"HEX: "+Hex(num)
  DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
```


HOTSPOT

KUVAUS

Kun kuva piirretään komennolla **DrawImage**, sen vasen yläkulma piirretään haluttuihin koordinaatteihin ruudulla. Tämä "vasen yläkulma" on kuvan omien koordinaattien mukaan (0,0). Voit kuitenkin vaihtaa tätä pistettä.

Tyypillisin esimerkki on haluta siirtää kuvan hallintapiste sen keskelle niin, että kuvan keskipiste piirtyy haluttuihin koordinaatteihin. Komenna silloin: **"HotSpot kuvamuuttuja"** sen jälkeen kun kuva on ladattu tai luotu. Tämä siirtää hallintapisteen keskelle. Voit myös määrittää pisteen itse, esim. **"HotSpot kuvamuuttuja, 50, 60"** siirtää hallintapisteen kohtaan (50,60) kuvassa. Oletus kaikilla kuvilla on vasen yläkulma, eli (0,0).

Komento sisältää myös automaattisen hallintapisteen muokkaimen. Tällöin CoolBasic siirtää hallintapisteen haluttuun kohtaan aina kun uusi kuva ladataan. Komento on tällöin **"HotSpot ON"**. Automaatin saa pois päältä komentamalla **"HotSpot OFF"**. Voit myös halutessasi määrittää automaatille X- ja Y-koordinaatit samaan tapaan kuin yksittäiselle kuvalle yllä mainitulla tavalla. Automaatti määritellään ennen kuvien lataamista.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

HOTSPOT päällä/kuvamuuttuja, [x, y]

- **päällä/kuvamuuttuja** = Se muuttuja, joka sisältää kuvan TAI 0=automaatti pois päältä TAI 1=automaatti päällä
- **x ja y** = Kokonaisluku, joka määrittää hallintapisteen (piirtokohdan) koordinaatit. (0,0) on kuvilla oletus.

ESIMERKKI

```
map=LoadImage("Media\map.bmp")

// Set the drawing point at 200,20
HotSpot map,200,20

// Notice that the image is introduced to be drawn to (0,0)
DrawImage map,0,0

// The image isn't really drawn at the top left corner
DrawScreen
WaitKey
```

IF

KUVAUS

JOS TÄMÄ SELITYS ON PITKÄ JA VAIVALLOINEN, KATSO TÄMÄ.

Tämä komento tarkastaa, onko annettu ehto totta. Kaikki muut paitsi arvo 0 tulkitaan todeksi. Mikäli ehto on totta, suoritetaan THEN-avainsanan jälkeen esitetyt komennot. Muussa tapauksessa hypätään IF-rakenteen yli tai seuraavaan kohtaan (ELSEIF, ELSE). Jos kirjoitat suoritettavat komennot THEN-sanan jälkeen yhdelle riville, erota ne toisistaan kaksoispisteellä.

If-lause voidaan suomentaa suoraan JOS-lauseeksi. Eli **JOS {jotain} NIIN {TEE_JOTAIN}**

Tässä on yksinkertainen IF-lause:
IF 1+1=2 THEN PRINT "Oikein"

Mutta entä jos meidän pitäisi tulostaa vastaus, myös silloin, kun lasku on väärin? Se menisi seuraavasti:
IF 1+1=2 THEN PRINT "Oikein" ELSE PRINT "Väärin"

Voimme myös suorittaa useita tehtäviä erottamalla ne toisistaan kaksoispisteellä:
IF 1+1=2 THEN PRINT "Oikein": PRINT "Hyvä!" ELSE PRINT "Väärin"

Jos ehto on totta (1+1=2), tulostetaan **kaksi** viestiä. Tosin tämä on aika sekavan näköinen rivi. Se voidaan jakaa myös osiin:

```
IF 1+1=2 THEN
    PRINT "Oikein"
    PRINT "Hyvä!"
ELSE
    PRINT "Väärin"
    PRINT "Yritä uudestaan"
ENDIF
```

Huomaa, että tällä tavalla esitettynä loppuun pitää pistää **ENDIF**, joka kertoo CoolBasicille, että rakenne loppuu tuossa kohtaa. Se toimii myös kohtana, johon CoolBasic hyppää, jos ehto on epätosi ja ELSE-lohko puuttuu:

```
IF 1+1=2 THEN
    PRINT "Oikein"
    PRINT "Hyvä!"
ENDIF
PRINT "Tämä ei kuulu ehtoon, ja näkyy, vaikka ehto onkin epätosi."
```

If-lauseeseen voidaan liittää myös toisia ehtoja, jos ensimmäinen ehto olikin epätosi. Tämä tapahtuu **ElseIF**-lausekkeella. Lisätietoja kohdasta "Toinen selitys" (linkki alapuolella).

Tulet törmäämään IF-rakenteeseen kaikkialla, mihin katsot. Sinun ei tarvitse sisäistää koko asiaa kerralla, vaan opit kaiken varmasti ajan myötä.

KÄYTTÖ

```
IF {ehdot} THEN {komennot1} ELSE {komennot2}

    TAI

IF {ehdot} THEN
    {komennot1}
ELSE
    {komennot2}
ENDIF
```

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
AddText "Use arrows to move..."

x=200:y=150

Repeat

    'update controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1

    'draw "player"
    Box x,y,10,10

DrawScreen
```

Until **EscapeKey**()

IMAGE

KUVAUS

Osoittaa kuvapuskuriin. Tarvitset tätä funktiota mm. **CopyBox**- ja pikselikäskyjen yhteydessä osoittamaan kuvaan.

KÄYTTÖ

IMAGE (kuvamuuttuja)

- **kuvamuuttuja** = Se muuttuja, johon kuva on luotu tai ladattu.

Katso myös: [SCREEN](#)

ESIMERKKI

```
'Load the map and cow
themap=LoadImage("Media/map.bmp")
cow=LoadImage("Media/cow.bmp")

'Copy the cow to the map - permanently
CopyBox 0,0,30,25,100,80,Image(cow),Image(themap)

While Not EscapeKey()

    Cls

    'draw the map
    DrawImage themap,MouseX()-200,MouseY()-150

    DrawScreen

Wend

End
```

IMAGEHEIGHT

KUVAUS

Kertoo kuvan korkeuden pikseleinä.

Palautettu arvo on kokonaisluku.

KÄYTTÖ

IMAGEHEIGHT (kuvamuuttuja)

- **kuvamuuttuja** = Se muuttuja, mihin kuva ladattiin.

Katso myös: [IMAGEWIDTH](#)

ESIMERKKI

```
img= LoadImage("Media\grass.bmp")

Color cbblack

AddText "This image is aligned to bottom"
AddText "The position is calculed by subtracting"
AddText "image height from screen bottom"

ClsColor cbwhite

Repeat

    'vertical align image to bottom
    DrawImage img,100,ScreenHeight()-ImageHeight(img)

    DrawScreen

Until EscapeKey()
```

IMAGESCOLLIDE

KUVAUS

Kertoo ovatko kaksi kuvaa päällekkäin. Tarkistus on pikselintarkka, mutta aavistuksen hitaampi kuin **ImagesOverlap**.

Funktio soveltuu erinomaisesti törmäyksen tunnistukseen esimerkiksi (pelaaja -> kenttä), (ammus -> kenttä) tai (kursori -> esine).

Vertaile esimerkin overlap-collide tarkkuutta.

Funktio palauttaa nollan, jos kuvat eivät ole päällekkäin, muutoin 1.

KÄYTTÖ

```
IMAGESCOLLIDE (kuva1, x1, y1, frame1, kuva2, x2, y2, frame2)
```

- **kuva1** = Ensimmäinen kuva, muuttuja.
- **x1** = Ensimmäisen kuvan X-koordinaatti.
- **y1** = Ensimmäisen kuvan Y-koordinaatti.
- **frame1** = Ensimmäisen kuvan animaatio-frame. Yleensä nolla.
- **kuva2** = Toinen kuva, muuttuja.
- **x2** = Toisen kuvan X-koordinaatti.
- **y2** = Toisen kuvan Y-koordinaatti.
- **frame2** = Toisen kuvan animaatio-frame. Yleensä nolla.

Katso myös: [IMAGESOVERLAP](#)

ESIMERKKI

```
'First make two images
img1= MakeImage(50,50)
img2= MakeImage(100,100)
DrawToImage img1
    Color cbred
    Circle 0,0,50
DrawToImage img2
    Color cbyellow
    Circle 0,0,100
DrawToScreen

'Initialize...
Color cbwhite
AddText "Arrows to move..."

x1=20
y1=20

x2=200
y2=150

'The game
Repeat

    'Update controls
    If LeftKey() Then x1=x1-1
    If RightKey() Then x1=x1+1
    If UpKey() Then y1=y1-1
    If DownKey() Then y1=y1+1

    'Draw both images
    DrawImage img1,x1,y1
    DrawImage img2,x2,y2

    'Check for collisions
    Text 10,280,"Overlap: "+ImagesOverlap(img1,x1,y1,img2,x2,y2)
    Text 290,280,"Collision: "+ImagesCollide(img1,x1,y1,0,img2,x2,y2,0)

    DrawScreen

Until EscapeKey()
```

IMAGESOVERLAP

KUVAUS

Kertoo ovatko kaksi kuvaa päällekkäin. Tarkistus on verrattain nopea, mutta ei välttämättä kovin tarkka. Tässä verrataan nimittäin kuvien rajoja. Kuvat muodostavat kaksi suorakaidetta, joiden päällekkäisyys tarkastetaan. Pikselintarkkaa törmäystä voi hakea funktiolla **ImagesCollide**.

Peleissä usein suorakaide-tunnistus riittää, joten pikselintarkka tunnistus ei ole välttämätön. Käytä tätä komentoa aina kun mahdollista. Esimerkki valaisee ImagesOverlapin huonot puolet.

Funktio palauttaa nollan, jos kuvat eivät ole päällekkäin, muutoin 1.

KÄYTTÖ

IMAGESOVERLAP (kuva1, x1, y1, kuva2, x2, y2)

- **kuva1** = Ensimmäinen kuva, muuttuja.
- **x1** = Ensimmäisen kuvan X-koordinaatti.
- **y1** = Ensimmäisen kuvan Y-koordinaatti.
- **kuva2** = Toinen kuva, muuttuja.
- **x2** = Toinen kuvan X-koordinaatti.
- **y2** = Toinen kuvan Y-koordinaatti.

Katso myös: [IMAGESCOLLIDE](#), [BOXOVERLAP](#)

ESIMERKKI

```
'First make two images
img1= MakeImage(50,50)
img2= MakeImage(100,100)
DrawToImage img1
  Color cbred
  Circle 0,0,50
DrawToImage img2
  Color cbyellow
  Circle 0,0,100
DrawToScreen

'Initialize...
Color cbwhite
AddText "Arrows to move..."

x1=20
y1=20

x2=200
y2=150

'The game
Repeat

  'Update controls
  If LeftKey() Then x1=x1-1
  If RightKey() Then x1=x1+1
  If UpKey() Then y1=y1-1
  If DownKey() Then y1=y1+1

  'Draw both images
  DrawImage img1,x1,y1
  DrawImage img2,x2,y2

  'Check for collisions
  Text 10,280,"Overlap: "+ImagesOverlap(img1,x1,y1,img2,x2,y2)
  Text 290,280,"Collision: "+ImagesCollide(img1,x1,y1,0,img2,x2,y2,0)

  DrawScreen

Until EscapeKey()
```

IMAGEWIDTH

KUVAUS

Kertoo kuvan leveyden pikseleinä.

Palautettu arvo on kokonaisluku.

KÄYTTÖ

IMAGEWIDTH (kuvamuuttuja)

- **kuvamuuttuja** = Se muuttuja, mihin kuva ladattiin.

Katso myös: [IMAGEHEIGHT](#)

ESIMERKKI

```
img= LoadImage("Media\grass.bmp")

Color cbblack

AddText "This image is aligned to right."
AddText "The position is calculed by subtracting"
AddText "image width from screen right side."

ClsColor cbwhite

Repeat

    'align image to right
    DrawImage img,ScreenWidth()-ImageWidth(img),0

    DrawScreen

Until EscapeKey()
```


INCLUDE

KUVAUS

Tämän käskyn tarkoituksena ei ole mikään muu kuin lätkästä sille ilmoitetun tiedoston sisältö osaksi lähdekoodia. Näin voit koostaa varsinaisen lähdekoodin useista tiedostoista. Termi on modulaarisuus. Modulaarisuuden ansiosta ohjelmankehitys on selkeämpää. Voit jäsenellä tietyn tyyppiset funktiot omiin moduuleihinsa. Kun moduuli on valmis, sen olemassaolon voi unohtaa, kunhan muistat linkittää sen lähdekoodiin. Tämä pienentää ja sitä kautta myös selventää lähdekoodia, kun rivimäärä pienenee. Käytä moduuleja, moduulit ovat hyviä.

Kukin "includeoitu" tiedosto ilmestyy koodiin vain kerran. Yleensä nämä linkitettävät tiedostot sisältävät globaalien muuttujien, vakioiden ja taulukoiden esittelyjä sekä tietenkin funktioita. Yleensä Include-rivit kirjoitetaan pääohjelmamoduulin alkuun, mutta luonteensa vuoksi sitä voidaan myös käyttää keskellä koodia liittämään osia lähdekoodiin.

KÄYTTÖ

```
INCLUDE "tiedosto"
```

- **"tiedosto"** = Merkkijono. Tiedostonimi tai polku linkitettävään tiedostoon.

ESIMERKKI

```
'Put these "finished" files here...
Include "include_globals.cb"
Include "include_menu.cb"
Include "include_game.cb"
Include "include_music.cb"
Include "include_sound.cb"
Include "include_ai.cb"

'the actual code begins...
```

Komennot

Täältä löydät ohjeen jokaiseen komenttoon. Kaikista komennoista on myös esimerkki. Kun kopioit esimerkin editoriin, muista avata aina tyhjä sivu sitä varten, tai muuten mahdolliset kuvat ja äänet eivät lataudu oikein.

Valitse komentosarja, josta haluat lisätietoa...

0-9 A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

KOMENTOSARJAT		
	PERUSKOMENNOT	Nämä tulisi kaikkien hallita. Ne kuuluvat ohjelmoinnin perusteisiin.
	MATEMATIIKKA	Sisältää muutaman <u>erittäin</u> hyödyllisen komennon.
	MERKKIJONOT (teksti)	Sisältää muutaman <u>erittäin</u> hyödyllisen komennon.
	TEKSTI (tulostus)	Fontit ja tekstin näyttäminen.
	TIEDOSTOT	Tiedostojen luku/kirjoitus/luominen ja kansiot.
	MUISTI	Muistin varaus, luku ja kirjoitus 
	LAITTEET JA SYÖTTEET	Lue näppäimistö- ja hiirisyötteitä.
	JÄRJESTELMÄ	Ajastin, tallennus, aika, virheet ja DLL. 
	ÄÄNET JA MUSIIKKI	Äänen tuotto ja CD-playback.
	ANIMAATIO	Katsele elokuvia!
	KUVAT	Piirrä kuvia ruudulle! Peliohjelmoinnin perusta.
	PIIRTOKOMENNOT	-
	OBJEKTIT	Peliohjelmointia!
	KAMERA	Peliohjelmointia!
	KARTAT	Peliohjelmointia!
	EFEKTIT	CoolBasicin oma, nopea partikkeliengine. Kipinät, savu, lähteet, räjähdykset jne... 

- **Lihavoidut** komentosarjat kannattaa käydä läpi, koska niitä tarvitaan usein ja ne tulisi jokaisen ohjelmoijan hallita.
-  -merkintä kertoo sellaisista komennoista, joita aloittelijan ei tarvitse ymmärtää eikä osata käyttää.

Uusimman version tästä oppaasta löydät [netistä](#).

INITOBJECTLIST

KUVAUS

CoolBasic tarjoaa mahdollisuuden käydä kaikki objektit läpi. Samaan tapaan kuin For..Each läpikäy kaikki tyyppin jäsenet tai For..Next läpikäy esimerkiksi kaikki taulukon solut. Lista alustetaan tällä komennolla, joka siirtää sisäisen osoittimen listan alkuun, eli ensiksi luotuun objektiin.

Pääset käsiksi itse objekteihin funktiolla **NextObject**. Kyseinen funktio palauttaa listassa seuraavan objektin. Kun lista on käyty kokonaan läpi, palauttaa NextObject nollan.

Tämä on kätevä esimerkiksi silloin, kun pitäisi käydä kaikki viholliset tai ammuksset läpi (ne tunnistetaan esim. ObjectInteger-funktiolla).

HUOM! Tämä komento ei toimi 3D-tilassa.

KÄYTTÖ

INITOBJECTLIST

Katso myös: [NEXTOBJECT](#)

ESIMERKKI

```
'Create 9 cows and position them into a row
For i=0 To 7
  obj=LoadObject("Media\cow.bmp")
  PositionObject obj,(-4+i)*40,0
Next i

'enable shapes to be rendered into world coordinates
DrawToWorld ON

ClsColor cbdarkred

InitObjectList 'reset the object iteration
Repeat

  If TIMER()>interv+1000 Then
    current=NextObject()
    If current
      drawx=ObjectX(current)
      drawy=ObjectY(current)
    Else
      'all objects have been iterated, start all over
      InitObjectList
    EndIf

    interv=TIMER()
  EndIf

  Color cbpink
  Circle drawx-20,drawy+20,40,OFF

  Text 0,0,current
  DrawScreen

Until EscapeKey()
```

INPUT

KUVAUS

Tämä funktio avaa konsolisytön ruudulle. Toisin kuin muissa ohjelmointikielissä, konsolisyttö toimii itsenäisesti pelin taustalla ilman, että pelin suoritus keskeytyy. Voit tietysti itse kirjoittaa pelin "pausettamisen" syötön ajaksi. Konsolia on ylläpidettävä kutsumalla sitä joka kerta pelin pääsilmutkassa.

Voit halutessasi määrittää *salasanamerkin*. Kirjoittamalla ylimääräisen parametrin, saat käyttöösi salasana kentän, joka ei näytä kirjoitettua tekstiä sellaisenaan, vaan sarjana salamerkkiä. Esimerkki selventää asiaa.

Voit itse määrittää konsolin paikan ruudulla käyttämällä ensin **Locate**-käskyä. Huomaa myös, että **AddText** asettaa konsolin kohdan automaattisesti seuraavalle riville. Konsoli ilmestyy siis automaattisesti siihen kohtaan, mihin tekstiosoitin on jäänyt.

Lopeta konsolisyttö kirjoittamalla koodi, joka reagoi esim. ENTER-painallukseen. Konsoli on suljettava **CloseInput**-komennolla, jolloin teksti nollautuu ja on valmis seuraavaa käyttökertaa varten.

Lisää esimerkkejä konsolin käytöstä saat selailemalla [merkkijonokomentojen](#) ohjeita.

KÄYTTÖ

INPUT (viesti, [salasanamerkki])

- **viesti** = Merkkijono. Teksti, joka näkyy juuri ennen syötettä. Sopivia tekstejä ovat esimerkiksi "", "? " ja ">>".
- **salasanamerkki** (*valinnainen*) = Merkkijono. Haluttu salasana merkki, yleensä "*". Jätä pois, jos haluat normaalin konsolin. Oletuksena pois päältä.

Katso myös: [CLOSEINPUT](#), [LOCATE](#)

ESIMERKKI

```
movex=1
movey=1

Repeat
  'UPDATE  CONSOLE

  Color cbwhite

  If console=OFF Then
    Text 0,0,"Press SPACE to open console"

    If KeyHit(cbkeyspace) Then
      console=ON
      Locate 10,30
      ClearKeys
    EndIf
  Else
    command$=Input("? ")
    'to enable password field, comment the line above
    'and uncomment the line below
    'command$=Input("? ", "*")

    Text 0,0,"Close console by pressing RETURN"

    If KeyHit(cbkeyreturn)
      console=OFF
      CloseInput
      ClearKeys
    EndIf
  EndIf

  'UPDATE  "GAME"

  Color cbred

  Circle x-20,y-20,40

  x=x+movex
  y=y+movey

  If x<0 Then movex= -movex
  If y<0 Then movey= -movey
  If x>400 Then movex= -movex
  If y>300 Then movey= -movey
```

DrawScreen

Until EscapeKey()

INSERT

KUVAUS

Siirtää listan jäsentä toiseen paikkaan. Voit esimerkiksi siirtää kokoelman viimeisen jäsenen ensimmäiseksi tai päinvastoin.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

INSERT tyyppi_muuttuja, kohde

- **tyyppi_muuttuja** = Jo valmis tyyplitetty muuttuja, joka toimii osoittimena listaan.
- **kohde** = Toinen tyyppimuuttuja, joka osoittaa uuden paikan.

Katso myös: [FIRST](#), [LAST](#), [BEFORE](#), [AFTER](#)

ESIMERKKI

```
Type test
  Field value
End Type

'First create 10 elements
For i=1 To 10
  t.test= New(test)

  t\value=i
Next i

'change the order a little bit
t.test= Last(test)
Insert t, First(test)

'Print them all
Repeat

  i=10
  For t.test= Each test
    Text 10,i,t\value
    i=i+16
  Next t

  DrawScreen

Until EscapeKey()
```

INSTR

KUVAUS

Etsii merkkijonosta esiintymän, ja kertoo sijainnin mistä se löytyi.

Isoilla ja pienillä kirjaimilla ON väliä.

Etsintä suoritetaan halutessasi vasta tietystä paikasta lähtien.

Funktio palauttaa siis esiintymän sijainnin. Jos etsittävää ei löydy, palautetaan nolla.

KÄYTTÖ

```
INSTR (jono, etsittävä, [lähtien])
```

- **jono** = Merkkijono.
- **etsittävä** = Merkkijonoesiintymä, jonka haluat korvattavan.
- **lähtien** (**valinnainen**) = Mistä lähtien merkkejä luetaan. Oletuksena 1. Tätä parametria ei ole pakko kirjoittaa.

ESIMERKKI

```
theline$="Don't you worry, babe. I'm an expert."
```

```
Text 0,0,theline$
```

```
Text 20,20,"The location of 'babe' is: "+Instr(theline$,"babe")
```

```
Text 20,33,"The location of 'hasta la vista' is: "+Instr(theline$,"hasta la vista")
```

```
'Try to search substring "you" after the 10th character...
```

```
Text 20,46,"The location of 'you' from character 10 is: "+Instr(theline$,"you",10)
```

```
' (of course it can't be found because "you" is located at 7)
```

```
DrawScreen
```

```
WaitKey
```

INT

KUVAUS

Muuntaa minkä tahansa arvon kokonaisluvuksi.

Jotkin komennot ja funktiot vaativat kokonaisluvun. Käytä tätä funktiota muuntamaan esim. desimaalimuuttuja kokonaisluvuksi, jonka jälkeen se kelpaa parametriksi.

Kun muunnetaan desimaaliluku, luku pyöristetään lähimpään kokonaislukuun. Siten esim. **2.3** -> **2** ja **4.7** -> **5**. Täsmälleen puolikas luku pyöristetään matemaattisten sääntöjen mukaan aina ylös päin.

Voit käyttää Int:tä muuttamaan merkkijonon kokonaisluvuksi. Muunnos kuitenkin pysähtyy epäkelvollisiin merkkeihin:

- `Int ( "57" ) -> 57`
- `Int ( "-6" ) -> -6`
- `Int ( "22.4" ) -> 22`
- `Int ( "Hello World" ) -> 0`

Katso myös RoundUp ja RoundDown erityyppisiä pyöristyksiä varten.

KÄYTTÖ

INT (arvo)

- **arvo** = Kokonaisluku, desimaaliluku tai merkkijono.

Katso myös: [ROUNDDOWN](#), [ROUNDUP](#), [FLOAT](#), [STR](#)

ESIMERKKI

```
For i=1 To 10

    value# = Rnd(-10,10)

    Text 0, i*12, "original: "+value#
    Text 200, i*12, "-> INT -> "+Int(value#)

Next i

DrawScreen
WaitKey
```


ISDIRECTORY

KUVAUS

Kertoo, onko annettu tiedostonimi kansio.

Palauttaa 1 jos on, muutoin nolla. Esimerkiksi jos tiedostonimi paljastuukin normaaliksi tiedostoksi, palautetaan nolla.

KÄYTTÖ

ISDIRECTORY (tiedosto)

- **tiedosto** = Merkkijono, tiedostonimi.

Katso myös: [FILEEXISTS](#)

ESIMERKKI

```
AddText "Input a filename (try also folders)"

Repeat
    k$=Input("::")
    DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
ClearText

isfile=FileExists(k$)

Text 10,30,"FILE EXISTS: "+isfile
Text 10,50,"IS FOLDER? "+IsDirectory(k$)
DrawScreen

WaitKey
```

KEYDOWN

KUVAUS

Tätä funktiota käytetään kertomaan onko jokin tietty näppäin painettuna alhaalla.

Funktiolle pitää kertoa halutun näppäimen [SCAN-koodi](#).

Scan-koodeja ei ole pakko muistaa ulkoa, vaan tilalle voi myös kirjoittaa jonkun CoolBasicin sisäänrakennetuista näppäinvakioista. Esim. **KeyDown(cbKeyJ)** tarkastaa, painetaanko J-näppäintä. Listan näppäinvakioista näet SCAN-taulukosta ([linkki yllä](#)).

Nuolinappuloille on olemassa omat funktionsa, joskaan mikään ei estä sinua kutsumasta niitä tämän kautta.

Palauttaa 1, jos on ja nolla, jos ei ole.

KÄYTTÖ

KEYDOWN (koodi)

- **koodi** = Halutun näppäimen [SCAN-koodi](#) tai näppäinvakio.

Katso myös: [KEYHIT](#), [KEYHIT](#), [GETKEY](#), [LEFTKEY](#), [RIGHTKEY](#), [UPKEY](#), [DOWNKEY](#), [ESCAPEKEY](#)

ESIMERKKI

```
AddText "Use W,A,S,D to move around"
AddText "Press SPACE to quit"
```

```
x=200:y=150
```

```
Repeat
```

```
  If KeyDown(cbkeya) Then x=x-1
  If KeyDown(cbkeyd) Then x=x+1
  If KeyDown(cbkeyw) Then y=y-1
  If KeyDown(cbkeys) Then y=y+1
```

```
  Box x,y,10,10
```

```
  DrawScreen
```

```
Until KeyDown(57) '57=space
```

KEYHIT

KUVAUS

Tämä funktio kertoo, onko haluttu näppäin "näpäytetty" alas, mutta se EI kerro onko se koko ajan painettuna. Toisin sanoen: tapahtuma käsitellään vain sillä hetkellä, kun näppäintä ollaan painamassa pohjaan.

Funktiolle pitää kertoa halutun näppäimen [SCAN-koodi](#).

Scan-koodeja ei ole pakko muistaa ulkoa, vaan tilalle voi myös kirjoittaa jonkun CoolBasicin sisäänrakennetuista näppäinvakioista. Esim. **KeyHit(cbKeyA)** tarkastaa, painetaanko J-näppäintä. Listan näppäinvakioista näet SCAN-taulukosta (linkki yllä).

Näppäinten tila päivitetään UpdateGame-kutsun aikana (tai DrawScreenissä, jos UpdateGamea ei kutsuta). Tämä tarkoittaa sitä, että et voi odottaa pääloopissa tiettyä aikaa ja tarkastaa onko näppäimen tila vieläkin ennallaan.

KÄYTTÖ

KEYHIT (koodi)

- **koodi** = Halutun näppäimen [SCAN-koodi](#) tai näppäinvakio.

Katso myös: [KEYDOWN](#), [KEYUP](#), [GETKEY](#)

ESIMERKKI

```
SCREEN 600,200,0,cwindowed

AddText "Hit space to increase the counter"
AddText "Then alter this example: KeyHit->KeyDown and see how it affects"

Repeat

    If KeyHit(cbkeyspace) Then counter+1

    Text 0,40,"Counter: "+counter

    DrawScreen

Until EscapeKey()
```

KEYUP

KUVAUS

Tämä funktio kertoo, onko haluttu näppäin "päästetty ylös", eli vapautettu. Funktiota voi verrata hiirellä klikkaukseen (kts. MouseUp)

Funktiolle pitää kertoa halutun näppäimen [SCAN-koodi](#).

Scan-koodeja ei ole pakko muistaa ulkoa, vaan tilalle voi myös kirjoittaa jonkun CoolBasicin sisäänrakennetuista näppäinvakioista. Esim. **KeyHit(cbKeyA)** tarkastaa, painetaanko J-näppäintä. Listan näppäinvakioista näet SCAN-taulukosta (linkki yllä).

Näppäinten tila päivitetään UpdateGame-kutsun aikana (tai DrawScreenissä, jos UpdateGamea ei kutsuta). Tämä tarkoittaa sitä, että et voi odottaa pääloopissa tiettyä aikaa ja tarkastaa onko näppäimen tila vieläkin ennallaan.

KÄYTTÖ

KEYUP (koodi)

- **koodi** = Halutun näppäimen [SCAN-koodi](#) tai näppäinvakio.

Katso myös: [KEYDOWN](#), [KEYHIT](#), [GETKEY](#)

ESIMERKKI

```
SCREEN 600,200,0,cwindowed

AddText "Hit space to increase the counter. Hold it down at first."
AddText "Then alter this example: KeyUp->KeyHit and see how it affects"

Repeat

    If KeyUp(cbkeyspace) Then counter+1

    Text 0,40,"Counter: "+counter

    DrawScreen

Until EscapeKey()
```

LAST

KUVAUS

Tämä funktio kertoo kokoelman viimeisen jäsenen.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

LAST (tyyppi_nimi)

- tyyppi_nimi = TYPE-kokoelman nimi.

Katso myös: [FIRST](#), [BEFORE](#), [AFTER](#)

ESIMERKKI

```
Type test
  Field value As Float
End Type

'First create 10 elements
For i=1 To 10
  t.test = New(test)

  t\value=Rnd(1)
Next i

'Print them all from end to start
Repeat

  t.test = Last(test)
  i=10
  While t<> NULL
    Text 10,i,t\value
    i=i+16
    t = Before(t)
  Wend

  DrawScreen

Until EscapeKey()
```

LEFT

KUVAUS

Lukee merkkijonon alusta tietyn määrän merkkejä.

Soveltuu esimerkiksi merkkijonojen pituuden rajoittamiseen.

Katso esimerkki.

KÄYTTÖ

```
LEFT (jono, montako)
```

- **jono** = Merkkijono, josta luetaan.
- **montako** = Montako merkkiä vasemmalta luetaan. Positiivinen kokonaisluku.

Katso myös: [RIGHT](#), [MID](#)

ESIMERKKI

```
mystring$="Hello World"

extracted$=Left(mystring$,4)

Text 10,10,"Original string: "+mystring$

Text 10,30,"First 4 characters of it: "+extracted$

DrawScreen

WaitKey
```

LEFTKEY

KUVAUS

Palauttaa 1, jos vasenta nuolinäppäintä painetaan. Muutoin palautetaan nolla.

KÄYTTÖ

```
LEFTKEY ()
```

Katso myös: [RIGHTKEY](#), [UPKEY](#), [DOWNKEY](#)

ESIMERKKI

```
AddText "Use arrows to move..."
```

```
x=200:y=150
```

```
Repeat
```

```
    'update controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1
```

```
    'draw "player"
    Box x,y,10,10
```

```
    DrawScreen
```

```
Until EscapeKey()
```

LEN

KUVAUS

Kertoo merkkijonon pituuden kirjaimina.

Tätä kannattaa käyttää esimerkiksi laskiessasi tekstin tasausta tai kun läpikäyt kaikki merkkijonon merkit **Mid**-funktiota apuna käyttäen.

Palautettu luku on kokonaisluku.

Jos haluat laskea merkkijonon pituuden *pikseleinä*, käytä **TextWidth**-funktiota.

KÄYTTÖ

LEN (jono)

- **jono** = Merkkijono, jonka pituus lasketaan.

ESIMERKKI

```
Repeat
  k$=Input("Give a sentence: ")

  'Print the length
  Text 0,50,"Text length: "+Len(k$)
  DrawScreen
Until KeyHit(cbkeyreturn)

CloseInput
```


LINE

KUVAUS
Piirtää viivan kahden pisteen välille. Käyttää nykyistä edusväriä (**Color**-käsky).

KÄYTTÖ

```
LINE x1, y1, x2, y2
```

- **x1** = Kokonaisluku tai desimaaliluku. Alkupisteen vaakakoordinaatti.
- **y1** = Kokonaisluku tai desimaaliluku. Alkupisteen pystykoordinaatti.
- **x2** = Kokonaisluku tai desimaaliluku. Loppupisteen vaakakoordinaatti.
- **y2** = Kokonaisluku tai desimaaliluku. Loppupisteen pystykoordinaatti.

Katso myös: [DOT](#), [BOX](#), [CIRCLE](#), [ELLIPSE](#)

ESIMERKKI

```
Repeat
    Color cbyellow

    Line 200,150,MouseX(),MouseY()

    DrawScreen
Until EscapeKey()
```

LOADANIMIMAGE

KUVAUS

Lataa kuvan kiintolevyiltä. Tämän jälkeen kuva voidaan piirtää **DrawImage**-komennolla.

Toisin kuin **LoadImage**, tällä komennolla voit ladata animoituja kuvasarjoja. Sen sijaan, että lataisit kaikki animaation framet eri tiedostoista, voit ladata kaikki yhdestä tiedostosta. Tämä yksinkertaistaa kuvien piirtämistä huomattavasti. Animoitu kuva piirretään normaalisti **DrawImage**-komennolla, mutta sen perään kirjoitetaan ylimääräinen parametri, joka kertoo CoolBasicille piirrettävän framen numeron. Ensimmäinen frame on nolla.

Jotta kuvasarja voidaan ladata oikein, täytyy CoolBasicille kertoa kaikkien framejen koko, sekä kuvasarjan pituus. Oikeaoppinen kuvasarja sisältää tasakokoisia kuvia, jotka on sijoitettu vierekkäin ilman erottavia pikseleitä. Esimerkki:



Tuetut formaatit ovat: **bmp, jpg, png ja tga**

Kaikki ladatut kuvat tuhoutuvat, kun SCREEN-komentoa käytetään ikkunakoon tai resoluution vaihtamiseen. Lataa kuvat uudelleen, ennen kuin yrität käyttää niitä. Muutoin ohjelma kaatuu.

Funktio palauttaa nollan, jos kuvaa ei voitu ladata.

Älä koskaan lataa kovalevyiltä mitään silmukan sisällä! Tätä komentoa tulisi käyttää vain ohjelman alussa.

KÄYTTÖ

```
LOADANIMIMAGE (tiedostonimi, frame_leveys, frame_korkeus, aloitus, pituus)
```

- **tiedostonimi** = Merkkijono. Ladattavan kuvan nimi tai polku.
- **frame_leveys** = Kokonaisluku. Framen leveys pikseleinä.
- **frame_korkeus** = Kokonaisluku. Framen korkeus pikseleinä.
- **aloitus** = Kokonaisluku. Monesko on ensimmäinen frame kuvatiedostossa. Yleensä nolla.
- **pituus** = Kokonaisluku. Montako framea kuvasarjassa on (esimerkkikuvassa on 8).

Katso myös: [LOADANIMIMAGE](#)

ESIMERKKI

```
FrameLimit 40

'Background white
ClsColor cbwhite

anim=LoadAnimImage("Media/note.bmp",32,32,0,8)
MaskImage anim,cbmagenta

While Not EscapeKey()

    'Draw the image
    DrawImage anim,184,134,f

    'animate
    If k=0 Then
        f=f+1
        If f>7 Then f=0
    EndIf
    k=k+1
    If k>2 Then k=0 'change frame every 3. time

    DrawScreen

Wend
```

LOADANIMOBJECT

KUVAUS

Tämä on vähän niin kuin LoadAnimImagen ja LoadObjectin välimuoto. Jos et vielä ole katsonut **LoadObjectia**, tee se nyt. Funktio lataa kuvan ja tekee siitä objektin. Sallitut kuvamuodot ovat **bmp, jpg, png ja tga**. Ero **LoadObjectiin** on se, että voit ladata kuvasarjan, joka muodostaa animoitavan objektin.

Voit animoida objektin komennoilla **PlayObject** ja **LoopObject**.

Jos lataat objektin, jota on tarkoitus voida pyörittää, kirjoita koko hoidon perään ylimääräinen parametri, joka määrittää kuinka sulavasti objekti pyörii ruudulla. Oikeastaan **tämä määrittää kuinka moneen eri suuntaan objekti voi osoittaa**. Arvo 8 saa objektin katsomaan 8 eri suuntaan sen kulumasta riippuen. Arvon tulee olla väliltä 1-360. **Mitä isompi sitä sulavampi kääntyvyys**.

VAROITUS! Isojen objektien lataaminen korkeilla pyörityslaaduilla saattaa olla erittäin hidasta. Joudut usein tekemään kompromissin laadun ja latausaikojen välillä. On ainakin suositeltavaa pistää peliin latausteksti tai -taustakuva, että käyttäjä tietää peliä ladattavan.

Jos lataat paljon samaa objektia, kannattaa mieluummin kopioida se funktiolla **CloneObject**. Se on PALJON nopeampaa. **Animoidut ja raskaat objektit hyötyvät tästä TODELLA paljon**. Alkuperäistä objektia ei silloin kuitenkaan voi poistaa tai käsitellä millään tavalla. Lisätietoja komennosta **CloneObject**.

Funktio palauttaa nollan, jos objektia ei voitu ladata.

Älä koskaan lataa kovalevyiltä mitään silmukan sisällä! Tätä komentoa tulisi käyttää vain ohjelman alussa.

KÄYTTÖ

```
LOADANIMOBJECT (tiedostonimi, frame_leveys, frame_korkeus, aloitus, pituus, [pyörityslaatu])
```

- tiedostonimi** = Merkkijono. Määrittää ladattavan kuvan tiedostonimen tai polun.
- frame_leveys** = Kokonaisluku. Framen leveys pikseleinä.
- frame_korkeus** = Kokonaisluku. Framen korkeus pikseleinä.
- aloitus** = Kokonaisluku. Monesko on ensimmäinen frame kuvatiedostossa. Yleensä nolla.
- pituus** = Kokonaisluku. Kuvasarjan framejen määrä.
- pyörityslaatu** (**valinnainen**) = Kokonaisluku väliltä 1-360. Kuinka moneen suuntaan objekti voi katsoa. Oletuksena 1 (ei käännä).

Katso myös: [LOADOBJECT](#), [PLAYOBJECT](#), [LOOPOBJECT](#)

ESIMERKKI

```
'make screen a bit zoomed
SCREEN 250,188,0,cbsizable

FrameLimit 40 'limit game speed

'load animation for both directions
zeroright=LoadAnimObject("Media\zerorun1.bmp",47,51,0,12)
MaskObject zeroright,cbmagenta

zeroleft=LoadAnimObject("Media\zerorun2.bmp",47,51,0,12)
MaskObject zeroleft,cbmagenta
ShowObject zeroleft,OFF

'default direction
zero=zeroright

Color cbblack
AddText "Try arrows"
ClsColor cbwhite

'MAIN LOOP
Repeat

    'run right!
    If RightKey() Then
        ShowObject zeroright,ON
        ShowObject zeroleft,OFF
        zero=zeroright
        PlayObject zero,2,11,0.25 'animate at speed 0.25
    'run left!
    ElseIf LeftKey() Then
        ShowObject zeroleft,ON
        ShowObject zeroright,OFF
        zero=zeroleft
        PlayObject zero,2,11,0.25 'animate at speed 0.25
    Else
        'nothing is pressed so freeze
```

```
    PlayObject zero,0  
EndIf
```

```
DrawScreen
```

```
Until EscapeKey()
```

LOADFONT

KUVAUS

Lataa fontin muistiin. Fontti haetaan automaattisesti Windowsin omasta fonttihakemistosta, joten sinun ei tarvitse kirjoittaa tiedostopolkua tai edes tiedostopäätettä.

Fontti voi olla TrueType-fontti tai rasterifontti, mutta symbolisia fontteja, kuten WingDings tai WebDings ei tueta.

Voit vaikuttaa fontin ulkonäköön määrittämällä jo latausvaiheessa sen erikoismuotoilut, kuten lihavointi, kursivointi ja alleviivaus. Näitä ei voi jälkikäteen muuttaa, vaan sinun on ladattava sama fontti erikseen eri muotoiluvalitsimilla.

Fontti pitää vielä aktivoida tekstikomentojen käyttöön komennolla **SetFont**.

Älä koskaan lataa kovalevyltä mitään silmukan sisällä! Tätä komentoa tulisi käyttää vain ohjelman alussa.

KÄYTTÖ

```
LOADFONT (nimi, [koko], [lihavointi], [kursivointi], [alleviivaus])
```

- **nimi** = Merkkijono, joka määrittää ladattavan fontin nimen ilman tiedostopäätettä.
- **koko** (*valinnainen*) = Fontin koko kokonaislukuna. Oletuksena 13.
- **lihavointi** (*valinnainen*) = **ON** päällä tai **OFF** pois. Oletuksena pois päältä.
- **kursivointi** (*valinnainen*) = **ON** päällä tai **OFF** pois. Oletuksena pois päältä.
- **alleviivaus** (*valinnainen*) = **ON** päällä tai **OFF** pois. Oletuksena pois päältä.

Katso myös: [SETFONT](#), [DELETEFONT](#)

ESIMERKKI

```
arial=LoadFont("arial",20)
impact=LoadFont("impact",28)
system=LoadFont("fixedsys")
coolb=LoadFont("coolb")
verdana=LoadFont("verdana",18,ON,OFF,ON) 'bold and underlined
times=LoadFont("times new roman",20,OFF,ON) 'italic
courier=LoadFont("courier new",40)
```

Repeat

```
Color cborange
SetFont arial
Text 40,20,"I'll look into it"

Color cblightgreen
SetFont impact
Text 100,100,"A sound plan!"

Color cbyellow
SetFont system
Text 260,270,"Dead man walking"

Color cbcyan
SetFont system
Text 20,200,"This..is..my BOOMSTICK!"

Color cbsilver
SetFont verdana
Text 70,220,"What do you want?!"

Color cblightred
SetFont times
Text 150,180,"Night beckons"

Color cbgreen
SetFont courier
CenterText 200,130,"Don't worry, be happy!"

DrawScreen
```

```
Until EscapeKey()
```

Kuvat : funktiot

LOADIMAGE

KUVAUS

Lataa kuvan kiintolevyiltä. Tämän jälkeen kuva voidaan piirtää **DrawImage**-komennolla.

Huomaa, että laitteistokohtaisesti et voi ehkä ladata suurempia kuvia kuin työpöydän koko. Lataa silloin kuva pienemmissä osissa.

Tuetut formaatit ovat: **bmp, jpg, png ja tga**

Kaikki ladatut kuvat tuhoutuvat, kun SCREEN-komentoa käytetään ikkunakoon tai resoluution vaihtamiseen. Lataa kuvat uudelleen, ennen kuin yrität käyttää niitä. Muutoin ohjelma kaatuu.

Funktio palauttaa nollan, jos kuvaa ei voitu ladata.

Älä koskaan lataa kovalevyiltä mitään silmukan sisällä! Tätä komentoa tulisi käyttää vain ohjelman alussa.

KÄYTTÖ

```
LOADIMAGE (tiedostonimi)
```

- **tiedostonimi** = Merkkijono. Ladattavan kuvan nimi tai polku.

Katso myös: [LOADANIMIMAGE](#)

ESIMERKKI

```
grass=LoadImage ("Media/grass.bmp")

cow=LoadImage ("Media/cow.bmp")

While Not EscapeKey ()

    'Draw the ground
    DrawImage grass,150,100

    'Draw the cow
    DrawImage cow,200,150

    DrawScreen

Wend
```

LOADMAP

KUVAUS

Tämä funktio lataa kartan levyltä muistiin. Kartta ilmestyy saman tien ruudulle oletuspaikkaan (0,0). Tilekartat piirretään aina keskitettynä ja ne käyttävät maailman koordinaatteja. Huomaa, että vain yksi kartta kerrallaan voi olla ladattuna. Jos teet tai lataat uuden kartan, entinen tuhotaan ensin.

Tilekarttoja tehdään **Tilester**-ohjelmalla, joka tulee CoolBasicin mukana. Löydät sen CoolBasicin valikoista. Tilesterillä on myös suomenkielinen manuaali, jonka löydät Tilesterin kansioista (Ohjelmatiedostot tai Program Files ja Tilester).

Tilekartoissa on aina neljä kerrosta: tausta, päälllys, törmäys ja data. Vain kaksi ensimmäistä piirretään. Objektit sijoitetaan näiden kahden väliin, joten päälliskerrokseen on mahdollista luoda piiloja, joiden alle pelaajat ja muut esineet voivat kätkeytyä. Voit kytkeä tarpeettomien kerroksien piirtämisen pois päältä lisänopeuden saavuttamiseksi. Kts. komento **SetMap**.

Tilekarttojen piirtämiseen käytetään **tilesettiä**. Se on iso kuva, joka sisältää paljon pienempiä kuvia. Ne on sijoitettu saumattomasti viereen samaan tapaan kuin animoiduissa kuvasarjoissa (kts. komento LoadAnimImage). Tätä settiä voi jopa vaihtaa lennossa, ja muuttaa kartan sävytystä tai ulkonäköä dramaattisesti. Tilesetti voi olla **BMP/JPG/PNG/TGA-kuva**. Voit myös merkkijonon sijasta antaa kuvamuuttujan, jos setti on jo ladattu kuvana muistiin.

Törmäyskerros sisältää tiedon niistä kohdista, joiden päältä ei voi kävellä. Tähän kerrokseen on siis tallennettu kaikki kentän "seinät". Käytä Tilesterin HIT-kerrosta luodaksesi nämä tiedot.

Data-kerrokseen voidaan tallentaa lisätietoa kartasta. Esimerkiksi ns. trigger-zonet, joiden päälle astuessaan pelaaja laukaisee jonkin tapahtuman. Myös hidastavaa mutaa tai polttavaa laavaa tai säteilyä voidaan koodata kartan data-kerrokseen. Data-kerros sisältää erilaisia lukuja. Data-kerrokseen pääsee käsiksi funktioilla **GetMap** ja **GetMap2**. **Datakerros on tarkoitettu edistyneille käyttäjille, ja se on yleensä aloittelijoille turha.**

Tilekarttaan käytetään normaaleita objekti-komentoja: Voit poistaa kartan komennolla **DeleteObject** tai siirtää sen toiseen paikkaan komennolla **PositionObject**.

Kartta niin ikään animoidaan komennolla **PlayObject** tai **LoopObject**. Ei ole väliä, kumpaa käytät, kartta pysyy animoituna kunnes se pysäytetään komennolla **StopObject**. Poikkeuksena fremejen aloitus- ja lopetusarvoilla ei karttojen kanssa ole merkitystä, vaan kartat animoidaan sen omien määritysten mukaan. Sen sijaan voit vaikuttaa animointinopeuteen. Tosin sekin poikkeaa hieman normaalien objektien animoinnista. Animaation nopeus määritetään sekunteina, joten numero 1 animoi karttaa joka sekunti, numero 2 vain joka toinen sekunti. 0.1 kymmenen kertaa sekunnissa jne.

Karttaan asetetaan törmäys- ja poimintaominaisuudet normaaleilla objektikomennoilla.

Tilekartat kannattaa yleensä ladata ennen muita objekteja, jotta kartta piirtyy kaikkien alle.

Funktio palauttaa nollan, jos kuvaa ei voitu ladata.

Älä koskaan lataa kovalevyltä mitään silmukan sisällä! Tätä kommentoa tulisi käyttää vain ohjelman alussa.

KÄYTTÖ

```
LOADMAP (tiedostonimi, tilesetti)
```

- tiedostonimi** = Merkkijono: Ladattavan kartan nimi tai polku.
- tilesetti** = Merkkijono: Kartan tilesetti-kuvan nimi tai polku. / Kokonaisluku: Kuvamuuttuja, jos setti on jo ladattu kuvana muistiin

Katso myös: [MAKEMAP](#)

ESIMERKKI

```
FrameLimit 40 'limit game performance

'Load map and animate it (2 times a second)
map= LoadMap("Media\cdm2.til","Media\tileset.bmp")
PlayObject map,0,0,0.5

'Load the character and setup a collision for it
guy= LoadObject("Media\soldier.bmp",72)
SetupCollision guy,map,2,4,2

'The game
Repeat

    'Control the guy
    If LeftKey() Then TurnObject guy,5
    If RightKey() Then TurnObject guy, -5
    If UpKey() Then MoveObject guy,4
    If DownKey() Then MoveObject guy, -4

    'Stick camera to the guy
    CloneCameraPosition guy
```

DrawScreen

Until EscapeKey()

LOADOBJECT

KUVAUS

Lataa kuvan ja tekee siitä objektin. Sallitut kuvamuodot ovat: **bmp, jpg, png ja tga.**

Objektit ovat pelimaailmaan ladattavia hahmoja, esineitä, vihollisia tai kerättäviä power-uppeja. Niitä voi erikseen liikuttaa, sijoittaa, pyörittää, osoittaa, animoida, teksturoida ja tappaa. Lisäksi ne voivat reagoida keskenään mm. törmäten toisiinsa. Myös kartat ja partikkelilähteet ovat objekteja.

Objektit ovat siitä kivoja, että kaikkia voi ohjata samoilla objektikomennoilla. Voit esimerkiksi animoida tilekartan samaan tapaan kuin normaalin objektin.

Jokaisella objektilla on oma kulma. Niitä voi pyörittää kaikkiin suuntiin, ja ne liikkuvat aina siihen suuntaan mihin osoittavat. Samalla tavalla kuin ihminen liikkuu sinne minne nenä osoittaa.

Jos lataat objektin, jota on tarkoitus voida pyörittää, anna sille ylimääräinen parametri, joka määrittää kuinka sulavasti objekti pyörii ruudulla. Oikeastaan **tämä määrittää kuinka moneen eri suuntaan objekti voi osoittaa**. Arvo 8 saa objektin katsomaan 8 eri suuntaan sen kulmasta riippuen. Arvon tulee olla väliltä 1-360. **Mitä isompi sitä sulavampi kääntyvyys**. Isojen objektien lataaminen parhaalla pyörityslaadulla saattaa kuitenkin olla hidasta. Parhaan kääntyvyyssarvon saat, kun jaat 360/käännösnopeus, missä käännösnopeus on se arvo millä kulmaa kasvatetaan tai vähennetään kun pelaaja haluaa kääntää objektia. Esimerkissä ihannearvo on 72, koska objekti kääntyy 5 astetta kerralla, ja $360/5=72$.

Jos lataat paljon samaa objektia, kannattaa mieluummin kopioida se funktiolla **CloneObject**. Se on PALJON nopeampaa. Alkuperäistä objektia ei silloin kuitenkaan voi poistaa tai käsitellä millään tavalla. Lisätietoja komennosta **CloneObject**.

Funktio palauttaa nollan, jos objektia ei voitu ladata.

Älä koskaan lataa kovalevyiltä mitään silmukan sisällä! Tätä komentoa tulisi käyttää vain ohjelman alussa.

KÄYTTÖ

```
LOADOBJECT (tiedostonimi, [pyörityslaatu])
```

- tiedostonimi** = Merkkijono. Määrittää ladattavan kuvan tiedostonimen tai polun.
- pyörityslaatu** (**valinnainen**) = Kokonaisluku väliltä 1-360. Kuinka moneen suuntaan objekti voi katsoa. Oletuksena 1 (ei käännä).

Katso myös: [LOADANIMOBJECT](#), [MAKEOBJECTFLOOR](#), [LOADMAP](#), [CLONEOBJECT](#)

ESIMERKKI

```
'limit speed for fast machines
FrameLimit 40

'load player
guy= LoadObject("Media\guy.bmp", 72)

Color cbblack
AddText "Move via arrows..."
ClsColor cbwhite

Repeat 'THE MAIN LOOP

    'Controls
    If LeftKey() Then TurnObject guy, 5
    If RightKey() Then TurnObject guy, -5

    If UpKey() Then MoveObject guy, 2
    If DownKey() Then MoveObject guy, -2

    DrawScreen

Forever
```

LOADPROGRAM

KUVAUS

Tämä komento lataa pelin kaikki muuttujat ja taulukot tallennustiedosta palauttaen tallennetun pelitilanteen.

Huomaa, että komento ei lataa tietoa kuvista, äänistä eikä mistään muustakaan ladatusta mediasta. Latauksen jälkeen kaikki kuvat ja muu media on ladattava takaisin ennen kuin peli voi jatkaa. Komento ei siis sellaisenaan sovi suuren pelin lataukseen.

Latauksen jälkeen voit vaihtoehtoisesti siirtyä juuri siihen kohtaan ohjelmassa, josta tallennus tapahtui komennolla **GotoSavedLocation**.

VAROITUS! Pelin tallennus on erittäin herkkä virhetilanteille. Tätä ominaisuutta tulisi käyttää vasta kun peli on valmis. Yhdenkin muuttujan lisääminen ohjelmaan tallennustiedoston tekemisen jälkeen voi sotkea niiden arvot latausvaiheessa. Siksi on hyvä mainita, että tallennustiedostot eivät ole yhteensopivia päivitettyjen peliversioiden kanssa.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

LOADPROGRAM tiedostonimi

- **tiedostonimi** = Merkkijono, tallennustiedoston nimi tai polku.

Katso myös: [SAVEPROGRAM](#), [GOTOSAVEDLOCATION](#)

ESIMERKKI

```
'Prevent the game from running too fast
FrameLimit 40

AddText "Move around using the arrows"
AddText "F5 = Save game"
AddText "F7 = Load game"

'Default Position
x=200
y=150

Repeat

    'Upadte controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1

    'draw the ball
    Color cbred
    Circle x-16,y-16,32,32

    'Save/load
    If KeyHit(cbkeyf5) Then SaveProgram "savegame.sav"
    If KeyHit(cbkeyf7) Then LoadProgram "savegame.sav"

    s$=Input("?")

    Text 300,0,x
    Text 300,20,y

    DrawScreen

Until EscapeKey()
```

LOADSOUND

KUVAUS

Lataa äänitiedoston muistiin, jotta sitä voitaisiin soittaa. Yleensä isot äänitiedostot (musiikit) kannattaa soittaa suoraan kovalevyiltä muistin säästämiseksi (PlaySound-komento), mutta pienet efektit, joita käytetään paljon, kannattaa ladata muistiin. Niinpä tällä funktiolla ladataan yleensä vain pelin ääniefekteiksi tarkoitettuja **Wav**-tiedostoja.

Tuetut formaatit ovat: **raw, wav, mp3 ja ogg**.

Älä koskaan lataa kovalevyiltä mitään silmukan sisältä! Tätä komentoa tulisi käyttää vain ohjelman alussa.

KÄYTTÖ

LOADSOUND (nimi)

- **nimi** = Merkkijono, joka määrittää ladattavan äänen nimen.

Katso myös: [PLAYSOUND](#), [DELETESOUND](#)

ESIMERKKI

```
weaponsound=LoadSound("Media\Blaster.wav")
```

```
Repeat
```

```
    If KeyHit(cbkeyspace) Then PlaySound weaponsound
```

```
Until EscapeKey()
```

```
DeleteSound weaponsound
```

LOCATE

KUVAUS

Asettaa tekstin aloituskohdan komennoille **Print**, **Write**, **Input** ja **AddText**.

Koordinaatit ilmoitetaan pikseleinä. Huomaa myös, että järjestys on (x,y) toisin kuin QBasicissa.

KÄYTTÖ

```
LOCATE x, y
```

- **x** = Tekstin vaakasuora sijainti. Kokonaisluku.
- **y** = Tekstin pystysuora sijainti. Kokonaisluku.

ESIMERKKI

```
'This is for AddText
```

```
Locate 300,60
```

```
AddText "HEY!"
```

```
'This is for Input
```

```
Locate 20,260
```

```
Repeat
```

```
    k$=Input("Write something:")
```

```
    DrawScreen
```

```
Until KeyHit(cbkeyreturn)
```

```
CloseInput
```

LOCK

KUVAUS

Lukitsee puskurin nopeita pikselikomentoja varten (**GetPixel2**, **PutPixel2**). Puskuriksi annetaan osoitin joko ruudulle tai kuvaan. Jätä valinnainen puskuripos, jos komennot suoritetaan nykyiseen puskuriiin (yleensä näyttöön).

Puskuriin osoitetaan funktioilla **Screen** ja **Image**.

Muista AINA vapauttaa puskuriopeeraation jälkeen komennolla **UnLock**.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

LOCK [puskuri]

- **puskuri** (*valinnainen*) = Puskuriosoitin. Kts. funktiot SCREEN() ja IMAGE(). Oletuksena nykyinen puskuripos.

Katso myös: [UNLOCK](#), [GETPIXEL2](#), [PUTPIXEL2](#), [SCREEN](#), [IMAGE](#)

ESIMERKKI

```
SCREEN 640,480

AddText "Press any key to copy..."

'First make a gradient
For i=0 To 240
    Color 0,i+15,0
    Line 0,i,640,i
Next i

DrawScreen OFF

WaitKey

'Lock screen
Lock

'make 154 481 iterations, copy each pixel
For y=0 To 240
For x=0 To 640
    PutPixel2 x,480-y,GetPixel2(x, y)
Next x
Next y

'Unlock screen
Unlock

DrawScreen

WaitKey
```

LOG

KUVAUS

Laskee annetun luvun luonnollisen logaritmin.

Logaritmia ei voi ottaa negatiivisista luvuista.

Tulos on desimaaliluku.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

LOG (luku)

- **luku** = Jokin kokonais- tai desimaaliluku.

Katso myös: [LOG10](#)

ESIMERKKI

'You can easily make new Log functions like this:

```
Function log5#(x#)  
  Return Log(x)/Log(5)  
End Function
```

LOG10

KUVAUS

Laskee annetun luvun 10-kantaisen logaritmin.

Tämä on päinvastainen laskutoimitus sille, että luku korotettaisiin kymmenenteen potenssiin.

Logaritmia ei voi ottaa negatiivisista luvuista.

Tulos on desimaaliluku.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

LOG10 (luku)

- **luku** = Jokin kokonais- tai desimaaliluku.

Katso myös: [LOG](#)

ESIMERKKI

```
i=10^5
```

```
Text 0,0,Log10(i)
```

```
DrawScreen
```

```
WaitKey
```

Loogiset operaattorit AND, OR, XOR ja NOT

KUVAUS

Näitä operaattoreita käytetään yhdistelemään ehtoja, kuten:

JOS luku > 5 JA luku < 10 NIIN tulosta "luku oli 6-9"
JOS nimi\$ = "Simo" TAI nimi\$ = "Jukka" NIIN tulosta "luku oli 6-9"

Näitä käytetään lähinnä IF-lausekkeiden kanssa! Katso lisätietoja **IF**-komennon ohjeesta.

Eli:

- AND = "ja"
- OR = "tai"
- XOR = "joko tai"
- NOT = "ei"

AND

Käytä tätä operaattoria yhdistääksesi kaksi väittämää ja tutkimalla ovatko ne *kummatkin* tosia. Tosi on mikä tahansa luku paitsi nolla. Epätosi taas on aina nolla. AND käyttäytyy tällä tavalla:

Vasen	Oikea	Tulos
TOSI	TOSI	TOSI
TOSI	EPÄTOSI	EPÄTOSI
EPÄTOSI	TOSI	EPÄTOSI
EPÄTOSI	EPÄTOSI	EPÄTOSI

Huomaa, että vain jos kummatkin puolet olivat tosia (ei nolla), niin ehto menee läpi.

OR

Käytä tätä operaattoria yhdistääksesi kaksi väittämää ja tutkimalla onko yksikään niistä tosi (eli ei nolla). OR käyttäytyy tällä tavalla:

Vasen	Oikea	Tulos
TOSI	TOSI	TOSI
TOSI	EPÄTOSI	TOSI
EPÄTOSI	TOSI	TOSI
EPÄTOSI	EPÄTOSI	EPÄTOSI

Huomaa, että jos yksikin on tosi (ei nolla), menee koko ehto läpi.

XOR

Tämä on poissulkeva "TAI". Käytä tätä operaattoria yhdistääksesi kaksi väittämää ja tutkimalla onko jompikumpi tosi (eli ei nolla). XOR käyttäytyy tällä tavalla:

Vasen	Oikea	Tulos
TOSI	TOSI	EPÄTOSI
TOSI	EPÄTOSI	TOSI
EPÄTOSI	TOSI	TOSI
EPÄTOSI	EPÄTOSI	EPÄTOSI

Huomaa, että vain jomman kumman on oltava tosi (ei nolla), jotta ehto menee läpi.

NOT

Käytä tätä operaattoria kääntääksesi toden epätodeksi ja päinvastoin. Not ottaa vain yhden luvun. NOT käyttäytyy tällä tavalla:

Oikea	Tulos
TOSI	EPÄTOSI
EPÄTOSI	TOSI

KÄYTTÖ
a AND b
a OR b
a XOR b
NOT a

- **a** = lause 1
- **b** = lause 2

ESIMERKKI: AND

```
'Psst.. The login name is "zero" and password is "secret"
```

```
'Ask for name
```

```
Repeat
```

```
    name$=Input("Login ID: ")
```

```
    DrawScreen
```

```
Until KeyHit(cbkeyreturn)
```

```
CloseInput
```

```
AddText name
```

```
Locate 0,20
```

```
'Ask for password
```

```
Repeat
```

```
    pass$=Input("Password: ", "*")
```

```
    DrawScreen
```

```
Until KeyHit(28)
```

```
ClearText
```

```
'Check if we have the access..
```

```
If name$="Zero" And pass$="secret" Then
```

```
    Locate 0,0
```

```
    AddText "Welcome, Zero"
```

```
    Repeat
```

```
        DrawScreen
```

```
    Until EscapeKey()
```

```
    End
```

```
Else
```

```
    Color cbred
```

```
    Locate 0,0
```

```
    AddText "Access denied."
```

```
    Repeat
```

```
        DrawScreen
```

```
    Until EscapeKey()
```

```
    End
```

```
EndIf
```

ESIMERKKI: OR

```
AddText "Input number between 1 and 5"
```

```
'ask the number
```

```
Repeat
```

```
    num=Input("number: ")
```

```
    DrawScreen
```

```
Until KeyHit(cbkeyreturn)
```

```
CloseInput
```

```
ClearKeys
```

```
'check the number
```

```
If num=1 Or num=2 Then
```

```
    Text 30,30,"You entered 1 or 2"
```

```
ElseIf num=3 Or num=4 Or num=5
```

```
    Text 30,30,"You entered 3 or 4 or 5"
```

```
ElseIf num= -20
```

```
    Text 30,30,"Secret cheat activated!"
```

```
Else
```

```
    Text 30,30,"You entered invalid number"
```

```
EndIf
```

```
DrawScreen
```

```
WaitKey
```

ESIMERKKI: XOR

ESIMERKKI: NOT

```
AddText "Press Return to quit..."
```

```
While Not KeyHit(cbkeyreturn)
```

```
    ClsColor Rand(255),Rand(255),Rand(255)
```

```
    DrawScreen
```

```
Wend
```

LOOPOBJECT

KUVAUS

Tämä komento animoi objektin. Objekti pitää olla ladattuna funktiolla **LoadAnimObject**. Aloitus- ja lopetuskohdat määrittävät mikä osa animoidusta kuvasarjasta soitetaan. Kaikki loput parametrit ovat valinnaisia. Huomaa, että ensimmäinen animaatioframe on nolla, joten jos haluat soittaa animaation alusta alkaen, ensimmäiseksi parametriksi on annettava nolla.

Voit käyttää miinusmerkkistä nopeutta, jos haluat toistaa animaation takaperin. Silloin alkuarvon pitää olla suurempi kuin lopetusarvon. Esim. **10->0**, nopeus **-5**. Kaikkia frameja käsitellään desimaalilukuina, joten saat halutessasi melko tarkan animaatiokeston. Nykyisen framen saat funktiolla **ObjectFrame**.

Valinnainen jatkumoparametri määrittää, jatketaanko animaation soittamista samasta kohdasta mihin se pysäytettiin komennolla **StopObject**. Tällä tavalla voit *tauottaa* animaatioita. Yleensä aloittelijat eivät tarvitse jatkumoa, joten sen voi jättää kirjoittamatta.

Toisin kuin **PlayObject**, nyt objektia toistetaan loputtomiin kunnes se pysäytetään erilaisella animointikäskyllä tai komennolla **StopObject**.

Tilekarttoja animoidaan eri tavalla. Alku- ja loppuarvoilla ei silloin ole merkistystä, pistä vaikka nollat kumpaankin. Sen sijaan animaationopeus on erilainen kuin normaaleissa objekteissa. Siinä se ilmoitetaan monenko sekunnin välien animaatio päivitetään. Esimerkiksi arvo 0.5 on päivittää kartan animaation 2 kertaa sekuntissa, ja arvo 4 joka 4 sekunnin välein.

KÄYTTÖ

```
LOOPOBJECT objektimuuttuja, [alku], [loppu], [nopeus], [jatkumo]
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **alku** (valinnainen) = Mistä kohtaa animaatiota aletaan soittamaan. Alku on nolla.
- **loppu** (valinnainen) = Animaation loppukohta.
- **nopeus** (valinnainen) = Animaation nopeus (suurempi on nopeampi). Oletus 0.1.
- **jatkumo** (valinnainen)
OFF = aloita aina alusta (oletus)
ON = jatka siitä mihin animaatio on pysäytetty

Katso myös: [PLAYOBJECT](#), [STOPOBJECT](#), [LOADANIMOBJECT](#)

ESIMERKKI

```
'make screen a bit zoomed
SCREEN 250,188,0,cbsizable

FrameLimit 40 'limit game speed

'load animation for both directions
zeroright=LoadAnimObject("Media\zerorun1.bmp",47,51,0,12)
MaskObject zeroright,cbmagenta

zeroleft=LoadAnimObject("Media\zerorun2.bmp",47,51,0,12)
MaskObject zeroleft,cbmagenta
ShowObject zeroleft,OFF

'default direction
zero=zeroright

Color cbblack
AddText "Try arrows"
ClsColor cbwhite

'MAIN LOOP
Repeat

    'run right!
    If RightKey() Then
        ShowObject zeroright,ON
        ShowObject zeroleft,OFF
        zero=zeroright
        LoopObject zero,2,11,0.25,ON 'animate at speed 0.25
    'run left!
    ElseIf LeftKey() Then
        ShowObject zeroleft,ON
        ShowObject zeroright,OFF
        zero=zeroleft
        LoopObject zero,2,11,0.25,ON 'animate at speed 0.25
    Else
        'nothing is pressed so freeze
        StopObject zero
    EndIf
```

DrawScreen

Until EscapeKey ()

LOWER

KUVAUS

Muuttaa tekstin pienellä kirjoitetuksi.

Toistaiseksi kaikkien maiden merkistöt eivät ole täysin tuettuja.

Komennon hyöty lienee muotoilun lisäksi kahden sanan tai lauseen yhtäläisyyden tarkastamisessa. Katso myös komennon **Upper** esimerkki.

KÄYTTÖ

```
LOWER (jono)
```

- **jono** = Merkkijono.

Katso myös: [UPPER](#)

ESIMERKKI

```
'This example will ask you to input a sentence  
'It will then be stripped from capital letters
```

```
AddText "Write a sentence."
```

```
AddText "Please remember the capitals."
```

```
Repeat
```

```
    sentence$=Input("? ")
```

```
    DrawScreen
```

```
Until KeyHit(cbkeyreturn)
```

```
CloseInput
```

```
'Print the results
```

```
AddText "You wrote:"
```

```
AddText Lower(sentence$)
```

```
Repeat
```

```
    DrawScreen
```

```
Until EscapeKey()
```

LSET

KUVAUS

Asettaa merkkijonon absoluuttisen pituuden. Lisää ylimääräisiä välilyöntejä merkkijonon loppuun niin, että merkkijonosta tulee halutun pituinen. Teksti pysyy vasemmalle tasattuna.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

```
LSET (jono, pituus)
```

- **jono** = Tasattava merkkijono.
- **pituus** = Uusi pituus välilyöntien lisäyksen jälkeen.

Katso myös: [RSET](#)

ESIMERKKI

```
st$=Lset("Hastalavista, baby!",28)
```

```
Text 10,10,"#" + st$ + "#"
```

```
DrawScreen
```

```
WaitKey
```

MAKEDIR

KUVAUS

Luo uuden kansion.

Sinun tulisi aina tarkastaa hakemiston luonnin onnistuminen funktiolla **isDirectory**.

KÄYTTÖ

MAKEDIR polku

- **polku** = Merkkijono, joka määrittää uuden hakemiston.

ESIMERKKI

```
Text 10,10,"Look at drive C:"
Text 10,25,"Then press any key..."
DrawScreen

WaitKey

MakeDir "C:\cbrules"

Text 10,10,"Now look again drive C:"
Text 10,25,"A new directory should be emerged."
DrawScreen

WaitKey
```

MAKEEMITTER

KUVAUS

Luo partikkelilähteen koordinaatteihin (0,0). Partikkelilähteitä käytetään koristamaan peliä luomalla savua, kipinöitä, tulta tai säteilyä. Partikkelit ovat pieniä objekteja, joita tarvitaan paljon. Suurien määrien hallitseminen on paitsi hidasta myös sekavaa ja työlästä. Tämän takia CoolBasic hoitaa tämän puolestasi. Täysin konekieliset ja erittäin optimoidut rutiinit takaavat mahdollisimman nopean partikkelipäivityksen. Ja mikä parasta - ohjelmoijan ei tarvitse huolehtia juuri mistään sen jälkeen kun lähde on luotu! CoolBasic tekee kaiken puolestasi.

Näin voit tehdä raketin jälkeensä jättämän hienon savu-efektin:

Liimaa lähde objektiin (pitämällä sitä jatkuvasti samalla kohdalla). Aseta partikkelien nopeudeksi nolla, aseta animoitu savukuva ja anna painovoiman arvoksi nolla. Kun ammus räjähtää seinään, voidaan lähdekin tuhota. Samalla tapaa voit luoda esimerkiksi avaruusaluksen jälkipolttimen.

Partikkelilähde käyttäytyy kuin mikä tahansa objekti. Sitä voidaan liikuttaa ja kääntää normaaleilla objektikomennoilla. Sille voidaan lisäksi asettaa elämä, jonka jälkeen se tuhoetaan automaattisesti. Jopa törmäys- ja poiminta-komennot toimivat lähteiden kanssa.

Voit poistaa partikkelilähteen komennolla **DeleteObject**.

Tälle funktiolle välitetään kaksi parametria: kuva ja elo aika. Elo aika on kokonaisluku, joka määrittää kuinka monta kertaa ruutu piirretään ennen kuin partikkeli häviää. Ei siis lähde, vaan lähteestä tullut partikkeli. Jos olet esimerkiksi rajoittanut pelisi toimimaan 40 kertaa sekunnissa (FrameLimit-komento), arvo 80 pitää partikkeleja elossa 2 sekuntia.

Partikkeleiden ominaisuuksia voi monipuolisesti säätää komennoilla **ParticleEmission**, **ParticleMovement** ja **ParticleAnimation**.

KÄYTTÖ

MAKEEMITTER (kuvamuuttuja, elo aika)

- **kuvamuuttuja** = Se muuttuja, mihin kuva ladattiin. Lähde suoltaa näitä kuvia.
- **elo aika** = Kokonaisluku, kuinka monen ruudun piirron verran partikkelit ovat olemassa.

Katso myös: [PARTICLEMOVEMENT](#), [PARTICLEANIMATION](#), [PARTICLEEMISSION](#)

ESIMERKKI

```
FrameLimit 40

img=LoadImage("Media\particle.bmp")

par=MakeEmitter(img,120) 'lifetime for 3 seconds

Repeat
    DrawScreen

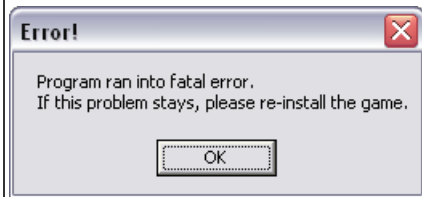
Forever
```


MAKEERROR

KUVAUS

Tämä komento luo keinotekoisien virheilmoituksen, jonka jälkeen ohjelma lopetetaan välittömästi.

Käyttäjälle näytetään jokin kuvaava teksti, jonka ohjelmoija saa itse määrittää.



VINKKI: Käyttämällä +Chr(10)+Chr(13)+, kuten esimerkissä, saat virheilmoitukseen rivinvaihdon.

KÄYTTÖ

MAKEERROR viesti

- **viesti** = Merkkijono, joka sisältää virheilmoituksen.

ESIMERKKI

```
Dim msg As String
Dim linechange As String

linechange=Chr(13)+Chr(10)
msg="An awfully fatal system error has occurred, and this program is terminated by force."
msg=msg+linechange
msg=msg+linechange
msg=msg+"Just kidding.. this was only a test. This error was launched on purpose :)"

Print "Hit any key to fire an error..."
WaitKey
MakeError msg
```

MAKEIMAGE

KUVAUS

Tekee uuden kuvan tietokoneen muistiin. Tämä on suhteellisen nopeaa, koska kuvaa ei tarvitse ladata mistään. Yleensä tyhjiä kuvia tarvitaan, kun niihin pitää piirtää. Aloittelijat eivät välttämättä tarvitse tätä komentoa.

Funktio palauttaa nollan, jos kuvaa ei voitu luoda.

KÄYTTÖ

```
MAKEIMAGE (leveys, korkeus, [framet])
```

- **leveys** = Kokonaisluku. Kuvan leveys pikseleinä.
- **korkeus** = Kokonaisluku. Kuvan korkeus pikseleinä.
- **framet** (**valinnainen**) = Kokonaisluku. Montako framea kuva sisältää. Oletuksena nolla (yksittäinen kuva).

Katso myös: [LOADIMAGE](#), [DRAWTOIMAGE](#)

ESIMERKKI

```
'Create an image
img= MakeImage(256, 256)

'Draw to it
DrawToImage img

  'Draw 100 random colour circles
  For i=1 To 100
    Color Rand(255), Rand(255), Rand(255)
    Box Rand(255),Rand(255),Rand(20,100),Rand(20,100)
  Next i

DrawToScreen

'Put our master-piece onto screen
DrawImage img, 0, 0

Color cbwhite
Text 10,270,"Press any key quit..."

'Refresh screen
DrawScreen

WaitKey
```

MAKEMAP

KUVAUS

Jos et ole vielä tutustunut funktioon LoadMap, tee se ensin.

Tämä funktio luo tyhjän kartan lennossa. Edellinen kartta tuhotaan automaattisesti. Koska kartta on tyhjä, on sinun "maalattava" sille tilesetti ennen kuin sitä voidaan käyttää. Teet tämän lataamalla tilesetin kuvana, ja käyttämällä **PaintObject**-komentoa. Katso esimerkki.

Rakenna sitten karttaa **EditMap**- ja **SetTile**-komennoilla.

Tilekartat kannattaa yleensä luoda ennen muita objekteja, jotta kartta piirtyy kaikkien alle.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

```
MAKEMAP (leveys, korkeus, tile_leveys, tile_korkeus)
```

- **leveys** = Kokonaisluku. Kartan leveys tileinä.
- **korkeus** = Kokonaisluku. Kartan korkeus tileinä.
- **tile_leveys** = Kokonaisluku. Yhden tilen leveys pikseleinä.
- **tile_korkeus** = Kokonaisluku. Yhden tilen korkeus pikseleinä.

Katso myös: [EDITMAP](#), [SETTILE](#)

ESIMERKKI

```
FrameLimit 40

'Create a map from Blank
map=MakeMap (32,32,32,32)
u=LoadObject ("Media\guy.bmp",72)

'Apply some graphics on it
tileset=LoadImage ("Media\tileset.bmp")
PaintObject map,tileset

'Randomly fill the map
For y=1 To MapHeight()
For x=1 To MapWidth()
  If Rand(1,5)=5 Then
    EditMap map,0,x,y,2 'drawing
    EditMap map,2,x,y,1 'collision
  Else
    EditMap map,0,x,y,145
  EndIf
Next x
Next y

'The game loop-itself
Repeat

  If UpKey() Then MoveObject u,2
  If DownKey() Then MoveObject u, -2
  If LeftKey() Then TurnObject u,5
  If RightKey() Then TurnObject u, -5

  'Glue camera to object position
  CloneCameraPosition u

  DrawScreen

Forever
```

MAKEMEMBLOCK

KUVAUS

Varaa muistia ja luo siitä muistipalan, johon käyttäjä voi kirjoittaa ja lukea tietoa.

Muistipaloja käytetään nopeisiin operaatioihin (esim. tiedontiivistys) ja parametrien välittämiseen DLL-tiedostoille.

- Tietokoneen muisti on pitkä pötkö bittejä. Ne ovat muistin perusyksiköitä ja pienimpiä mahdollisia tietolokeroita.
- Bitti voi olla joko nolla tai yksi.
- Yksi tavu koostuu kahdeksasta bitistä. Kahdeksan bitin jonosta voidaan muodostaa 256 erilaista yhdistelmää, joten erilaisia tavuja on olemassa 256 kappaletta. Niiden numerointi menee 0-255.
- Yksi merkki on yksi tavu. Siten esim. ASCII-merkistössä on 255 erilaista merkkiä/kirjainta + ns. nollamerkki.
- Jokaisella tavulla on ns. muistiosoite. Niiden avulla muuttujien arvoja tallennetaan ja luetaan muistista. Muuttuja on siis ikään kuin muistilappu, joka sisältää muuttujalle varatun muistipaikan.

CoolBasicin avulla varattuun muistiin voidaan kirjoittaa tavuja, kahdenlaisia kokonaislukuja sekä desimaalilukuja. Huomaa, että kun kirjoitat/luet tietoa, ensimmäinen paikka on nolla ja viimeinen KOKO-1.

On myös tärkeää pitää huoli siitä, että et kirjoita varatun muistialueen ulkopuolelle. Se saattaa kaataa ohjelman tai jopa koko järjestelmän.

KÄYTTÖ

MAKEMEMBLOCK (koko)

- **koko** = Muistipalan koko tavuina. Positiivinen kokonaisluku.

Katso myös: [POKEBYTE](#), [POKEINT](#), [POKEFLOAT](#), [PEEKBYTE](#), [PEEKINT](#), [PEEKFLOAT](#)

ESIMERKKI

```
'Reserve memory for 8 bytes.
'It's big enough to store two integer numbers.
mem=MakeMEMblock (8)

PokeInt mem,0,8 'write number 8 at the beginning
PokeInt mem,4, -67 'write another number after the first one

Text 10,10,PeekInt(mem,0) 'read the first integer and print it
Text 10,30,PeekInt(mem,4) 'read the second integer and print it

DrawScreen

WaitKey

DeleteMEMblock mem 'free the memory we used
```

MAKEOBJECT

KUVAUS

Tekee näkymättömän objektin. Sille voidaan kuitenkin suorittaa normaaleita törmäystestejä, poimintoja, liikutusta ja kääntämistä.

Täydellinen liikkuviin, mutta näkymättömiin ammuksiin tms.

Voit kuitenkin maalata tämän objektin haluamallasi toisella objektilla tai kuvalla ja antaa sille muodon komennolla **PaintObject**.

KÄYTTÖ

```
MAKEOBJECT ( [pyörityslaatu] )
```

- **pyörityslaatu** (**valinnainen**) = Kokonaisluku väliltä 1-360. Kuinka moneen suuntaan objekti voi katsoa. Oletuksena 1 (ei käänny).

Katso myös: [PAINTOBJECT](#), [MAKEOBJECTFLOOR](#)

ESIMERKKI

```
'limit speed for fast machines
FrameLimit 40

'load player
guy= LoadObject("Media\guy.bmp", 72)

Color cbblack
AddText "Move via arrows..."
AddText "Watch out the invisible walls"
ClsColor cbwhite

For y= -128 To 128 Step 64
For x= -192 To 192 Step 64
    temp=MakeObject() 'make a new object
    PositionObject temp,x,y 'locate it
    ObjectRange temp,32 'setup a collision
    SetupCollision guy,temp,2,2,2
Next x
Next y

Repeat 'THE MAIN LOOP

    'Controls
    If LeftKey() Then TurnObject guy, 5
    If RightKey() Then TurnObject guy, -5

    If UpKey() Then MoveObject guy, 2
    If DownKey() Then MoveObject guy, -2

    DrawScreen

Forever
```

MAKEOBJECTFLOOR

KUVAUS

Tekee aluksi näkymättömän "lattiaobjektin". Se on äärettömän suuri, ja monistuu kaikkialle pelimaailmaan. Sen ihanteellinen käyttötarkoitus on luoda esimerkiksi nurmikkoja muodostamaan loppumattoman suuren pelimaailman.

Huomaa, että lattiaobjektit piirtyvä KAIKKIEN muiden objektien alle. Kun luot uuden lattiaobjektin, se piirtyy jopa muiden lattiaobjektien alle. Piirtojärjestys on siis vastakkainen normaaleihin objekteihin verrattuna. Lattiaobjekteille ei voida määrittää poimintaa eikä törmäystä.

Käytä **PaintObject**ia maalaamaan lattiaobjekti jollain kuvalla ennen sen käyttämistä. Muista kuitenkin, että et voi poistaa kyseistä kuvaa (DeleteImage) ennen kuin lattiaobjekti on poistettu tai ohjelma kaatuu.

Voit luoda esimerkiksi taustoja pelimaailmoille ja kartoille. Käytä Z-sijaintia syvyysvaikutelmaan. Lattiaobjekteja voidaan jopa animoida.

KÄYTTÖ

MAKEOBJECTFLOOR ()

Katso myös: [PAINTOBJECT](#)

ESIMERKKI

```
floor=MakeObjectFloor()
grid=LoadImage("Media\whitegrid.bmp")
PaintObject floor,grid

grass=MakeObjectFloor()
lawn=LoadImage("Media\grass.bmp")
PaintObject grass,lawn
PositionObject grass,0,0, -2 'apply z-depth

'Load the map
map=LoadMap("Media\testmap.til","Media\tileset.bmp")

AddText "Use Arrows to move..."

SetMap map,OFF,ON 'make back layer transparent

PositionCamera 0, -90
RotateCamera 0,90 'make the camera face upwards

Repeat

    'Controls
    If LeftKey() Then MoveCamera -1,0
    If RightKey() Then MoveCamera 1,0
    If UpKey() Then MoveCamera 0,1
    If DownKey() Then MoveCamera 0, -1

    DrawScreen

Until EscapeKey()
```

MAPHEIGHT

KUVAUS

Kertoo ladatun kartan korkeuden tileinä. Jos haluat tietää korkeuden pikseleissä, kokeile komentoa **ObjectSizeY**.

Palautettu luku on kokonaisluku.

KÄYTTÖ

MAPHEIGHT ()

Katso myös: [MAPWIDTH](#), [OBJECTSIZEY](#)

ESIMERKKI

```
FrameLimit 40 'limit game performance

'Load map and animate it
map= LoadMap("Media\cdm2.til","Media\tileset.bmp")

AddText "Scanning map..."
DrawScreen

i=0
For y=1 To MapWidth()
For x=1 To MapWidth()
    If GetMap2(0,x,y)=137 Then i=i+1
Next x
Next y

'Remove map from memory
DeleteObject map

AddText "DONE"
AddText ""
AddText "Tile 137 occurred "+i+" times."
DrawScreen

WaitKey
```

MAPWIDTH

KUVAUS

Kertoo ladatun kartan leveyden tileinä. Jos haluat tietää leveyden pikseleissä, kokeile komentoa **ObjectSizeX**.

Palautettu luku on kokonaisluku.

KÄYTTÖ

MAPWIDTH ()

Katso myös: [MAPHEIGHT](#), [OBJECTSIZEX](#)

ESIMERKKI

```
FrameLimit 40 'limit game performance

'Load map and animate it
map= LoadMap("Media\cdm2.til","Media\tileset.bmp")

AddText "Scanning map..."
DrawScreen

i=0
For y=1 To MapWidth()
For x=1 To MapWidth()
    If GetMap2(0,x,y)=137 Then i=i+1
Next x
Next y

'Remove map from memory
DeleteObject map

AddText "DONE"
AddText ""
AddText "Tile 137 occurred "+i+" times."
DrawScreen

WaitKey
```


MASKIMAGE

KUVAUS

Tällä komennolla voit asettaa kuvan läpinäkyvän värin (oletuksena musta) toiseksi. Tämä on erityisen kätevää silloin, kun kuvasi käyttää selvästi eri maskiväriä ja haluat mustatkin osat näkyviin.

Väri ilmoitetaan RGB-järjestelmän mukaisesti. Väri muodostuu kolmesta komponentista, joista kukin saa arvon 0-255. (0,0,0) on musta ja (255,255,255) on valkoinen. Lisätietoa RGB-koodauksesta oppitunnista [Värit](#), [värikartasta](#) ja sanakirjasta.

Yleinen vaihtoehtoinen väri on puhdas magenta (255,0,255 tai *cbMagenta*), koska se on niin harvinainen.



KÄYTTÖ

MASKIMAGE kuvamuuttuja, punainen, vihreä, sininen

- **kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.
- **punainen** = Kokonaisluku. Värin punainen komponentti.
- **vihreä** = Kokonaisluku. Värin vihreä komponentti.
- **sininen** = Kokonaisluku. Värin sininen komponentti.

Katso myös: [DEFAULTMASK](#), [MASKOBJECT](#)

ESIMERKKI

```
'BEFORE YOU START, TAKE A LOOK AT THE IMAGE
'ALSO, TRY TO REMOVE THE MASKIMAGE-COMMAND AND SEE WHAT HAPPENS
FrameLimit 40

'Load an image and mask it
img=LoadImage("Media/mask.bmp")
MaskImage img,255,0,255

'Apply the image as Mouse Pointer
ShowMouse img

'Add some text to screen
Color cbblack
AddText "Use mouse to move around"

'Background to white
ClsColor cbwhite

While Not EscapeKey()

    'Draw green background pattern
    Color cbgreen
    For i=0 To 400 Step 20
        Line i,0,i+300,300
    Next i
    For i=20 To 300 Step 20
        Line 0,i,400,i+400
    Next i

    'Update screen
    DrawScreen

Wend
```

MASKOBJECT

KUVAUS

ÄLÄ KÄYTÄ TÄTÄ SELLAISIIN OBJEKTEIHIN, JOITA PYÖRITETÄÄN.

Voit asettaa objektin läpinäkyvän värin (oletuksena pikimusta) tällä komennolla. Arvot annetaan RGB-standardin mukaisesti. Lisätietoa sanakirjasta ja Grafiikka-sarjan värikomennosta. Tämä komento sopii mm. valikko-objekteihin, jotka sisältävät mustalla reunustettua tai mustaa tekstiä. Kunhan objektia ei pyöritetä missään vaiheessa.

Värin voi ilmoittaa myös [värivakioilla](#), tai pikseliarvolla (esimerkki COLOR-käskyn oppaasta).

KÄYTTÖ

MASKOBJECT objektimuuttuja, punainen, vihreä, sininen

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **punainen** = Kokonaisluku väliltä 0-255. Maskivärin punainen komponentti.
- **vihreä** = Kokonaisluku väliltä 0-255. Maskivärin vihreä komponentti.
- **sininen** = Kokonaisluku väliltä 0-255. Maskivärin sininen komponentti.

ESIMERKKI

MAX

KUVAUS

Palauttaa kahdesta annetusta arvosta suuremman.

Voit vertailla kokonaislukuja, desimaalilukuja ja jopa merkkijonoja. Merkkijonojen vertailussa palautetaan se, joka on aakkosissa suurempi. Huomaa tällöin, että saman aakkosen iso ja pieni kirjain ovat eriarvoisia. Kunnollista vertailua varten kummatkin merkkijonot kannattaa ensin käsitellä joko Lower- tai Upper -komennoilla.

KÄYTTÖ

```
MAX (arvo1, arvo2)
```

- **arvo1** = Ensimmäinen vertailtava. Kaikki kolme perustietotyyppiä käyvät.
- **arvo2** = Toinen vertailtava. Kaikki kolme perustietotyyppiä käyvät.

Katso myös: [MIN](#)

ESIMERKKI

```
Text 10,10,Max(-6,8)
```

```
Text 10,30,Max(7.6,66.9)
```

```
Text 10,50,Max("beta","alfa")
```

```
DrawScreen
```

```
WaitKey
```

MEMBLOCKSIZE

KUVAUS

Kertoo muistipalan koon, eli sen luvun, jolla muistipala varattiin (MakeMemBlock).

KÄYTTÖ

MEMBLOCKSIZE (muistipala)

- muistipala** = Se muuttuja, mihin muistipala luotiin.

ESIMERKKI

```
'create a memory block with 64 bytes
mem=MakeMemBlock(64)

memsize=MEMblockSize(mem)
Text 0,0,memsize

DrawScreen

WaitKey

DeleteMemBlock mem 'free the memory we used
```

MEMCOPY

KUVAUS

Kopioi osan muistipalasta toiseen muistipalaan.
Kopiointi on paljon nopeampi kuin sama operaatio tavu tavulta.

KÄYTTÖ

MEMCOPY lähde, sijainti1, kohde, sijainti2, määrä

- **lähde** = Se muuttuja, mihin muistipala luotiin. Muistipalikka **josta** kopioidaan.
- **sijainti1** = Sijainti lähdepalassa. 0=ensimmäinen, (KOKO-1)=viimeinen. Kokonaisluku.
- **kohde** = Se muuttuja, mihin muistipala luotiin. Muistipalikka **johon** kopioidaan.
- **sijainti2** = Sijainti kohdepalassa. 0=ensimmäinen, (KOKO-1)=viimeinen. Kokonaisluku.
- **määrä** = Kuinka monta tavua kopioidaan. Kokonaisluku.

ESIMERKKI

```
'create a memory block with 64K bytes
mem=MakeMEMblock(64000)

'fill half of the memblock with crap
For i=0 To 32000
    PokeByte mem,i,Rand(255)
Next i

Text 0,0,"Press any key to copy..."
DrawScreen
WaitKey

'duplicate the crap in the memblock
MemCopy mem,0,mem,32000,32000

Text 0,0,"Copying ready."
DrawScreen

WaitKey

DeleteMEMblock mem
```

MID

KUVAUS

Lukee merkkijonon keskeltä tietyn määrän merkkejä.

Katso esimerkki.

KÄYTTÖ

```
MID (jono, paikka, [montako])
```

- **jono** = Merkkijono, josta luetaan.
- **paikka** = Monennestako merkistä lukeminen alkaa. Positiivinen kokonaisluku.
- **montako** (**valinnainen**) = Montako merkkiä luetaan. Jätä pois, jos haluat lukea kaikki loput merkit. Tämä on oletusarvo.

Katso myös: [LEFT](#), [RIGHT](#)

ESIMERKKI

```
mystring$="Hello World"
```

```
'extract some characters from the middle
```

```
Text 0,0,Mid(mystring$,7,4)
```

```
'extract all characters starting from 3
```

```
Text 0,20,Mid(mystring$,3)
```

```
DrawScreen
```

```
WaitKey
```

MIN

KUVAUS

Palauttaa kahdesta annetusta arvosta pienemmän.

Voit vertailla kokonaislukuja, desimaalilukuja ja jopa merkkijonoja. Merkkijonojen vertailussa palautetaan se, joka on aakkosissa pienempi. Huomaa tällöin, että saman aakkosen iso ja pieni kirjain ovat eriarvoisia. Kunnollista vertailua varten kummatkin merkkijonot kannattaa ensin käsitellä joko Lower- tai Upper -komennoilla.

KÄYTTÖ

```
MIN (arvo1, arvo2)
```

- **arvo1** = Ensimmäinen vertailtava. Kaikki kolme perustietotyyppiä käyvät.
- **arvo2** = Toinen vertailtava. Kaikki kolme perustietotyyppiä käyvät.

Katso myös: [MAX](#)

ESIMERKKI

```
Text 10,10,Min(-6,8)
Text 10,30,Min(7.6,66.9)
Text 10,50,Min("beta","alfa")
```

```
DrawScreen
```

```
WaitKey
```

MIRROROBJECT

KUVAUS

Tämä komento peilaa objektin vaaka- tai pystysuunnassa. Halutessa vaikka molemmiin päin. Tämä on erityisen kätevää esimerkiksi sivulta kuvatuissa tasoloikissa, kun päähahmolle pitäisi saada animaatiot sekä vasemmalle että oikealle. Sen sijaan, että tekisit käännetyt kuvasarjat jo valmiiksi peliäsi varten, CoolBasic voi kääntää objektit täysin automaattisesti, mikä säästää sinut vaivalta.

Partikkelilähteitä tai tilekarttoja ei toistaiseksi voi kääntää.

Reaaliaikainen peilaus on suhteellisen raskas operaatio, ja sitä tulisi välttää. Laske mieluummin peilatut objektit valmiiksi ohjelmasi alussa.

KÄYTTÖ

MIRROROBJECT objektimuuttuja, tapa

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **tapa** (valinnainen)
 - 0** = peilaus vaakasuunnassa (oletus)
 - 1** = peilaus pystysuunnassa
 - 2** = peilaus vaaka- ja pystysuorassa

ESIMERKKI

```
'load both (animated) objects
run=LoadAnimObject("Media\zerorun1.bmp",47,51,0,12)
MaskObject run,cbmagenta
run2=CloneObject(run)

'mirror the second one (mirror&flip)
mirrorobject run2,2

'move them a bit further from each other
PositionObject run,0,50
PositionObject run2,0, -50

'animate them (continuous)
LoopObject run,2,11
LoopObject run2,2,11

'the game loop itself
Repeat
    DrawScreen
Forever
```


MOD

KUVAUS

Laskee kahden luvun jakojäännöksen. Tätä matemaattista operaattoria käytetään samaan tapaan kuin mitä tahansa muutakin operaattoria (+ - * / ...). Luvulle suoritetaan jakolasku. Joskus jako ei kuitenkaan mene tasan, ja siitä jää ns. jakojäännös. Eli käytännössä luku jaetaan niin monta kertaa kuin mahdollista. Kun jakoa ei voi jatkaa, palautetaan "ylijäämä" eli jakojäännös.

KÄYTTÖ

a MOD b

- **a** = jaettava
- **b** = jakaja

ESIMERKKI

Text 0,0,(10 MOD 3)
DrawScreen
WaitKey
'Prints "1" because number 3 goes to 10 totally 3 times and the surplus is 1

Syötteet : funktiot

MOUSEDOWN

KUVAUS

Kertoo, onko tietty hiiren nappula alhaalla; palauttaa 1, jos on ja nolla, jos ei ole.

KÄYTTÖ

MOUSEDOWN (painike)

- **painike**

1 = vasen painike

2 = oikea painike

3 = keskimmäinen painike

Katso myös: [MOUSEHIT](#), [MOUSEUP](#), [GETMOUSE](#)

ESIMERKKI

```
AddText "Press mouse"
```

```
AddText "Press mouse middle button to quit..."
```

```
Repeat
```

```
    Text 20,40,"LEFT BUTTON: "+MouseDown(1)
```

```
    Text 20,55,"RIGHT BUTTON: "+MouseDown(2)
```

```
    DrawScreen
```

```
Until EscapeKey() Or MouseDown(3)
```

MOUSEHIT

KUVAUS

Käytä tätä funktiota tarkastaaksesi painettiinko hiiren painike juuri alas. Tämä on siis se tapahtuma, kun nappulaa painetaan. Jos haluat tarkastaa onko painike alhaalla, käytä mieluummin funktiota *MouseDown*.

Hiiren tila päivitetään painikkeiden osalta UpdateGame-kutsun aikana (tai DrawScreenissä, jos UpdateGame ei kutsuta). Tämä tarkoittaa sitä, että et voi odottaa pääloopissa tiettyä aikaa ja tarkastaa onko painikkeen tila vieläkin ennallaan.

KÄYTTÖ

MOUSEHIT (painike)

- **painike**

1 = vasen painike

2 = oikea painike

3 = keskimmäinen painike

Katso myös: [MOUSEDOWN](#), [MOUSEUP](#), [GETMOUSE](#)

ESIMERKKI

```
AddText "Click the box"
```

```
AddText "Try to hold mouse down on the box"
```

```
Repeat
```

```
    Color cbsilver
```

```
    Box 100,100,200,100
```

```
    'check if the mouse is inside the box
```

```
    If MouseX()>100 And MouseX()<300 And MouseY()>100 And MouseY()<200
```

```
        If MouseHit(1)
```

```
            'draw another (red) box
```

```
            Color cbred
```

```
            Box 100,100,200,100
```

```
        EndIf
```

```
    Else
```

```
        'outside... clear the click state.
```

```
        'You can comment this and see what happens
```

```
        ClearMouse
```

```
    EndIf
```

```
    DrawScreen
```

```
Until EscapeKey()
```

MOUSEMOVEX

KUVAUS

Tämä funktio kertoo hiiren nopeuden vaakasuuntaan. Kätevä silloin, kun haluat esimerkiksi oman hiirikursorin tai ohjaussysteemin. Silloin ruudun koko ei rajoita sijaintia.

Tulos on kokonaisluku.

KÄYTTÖ

MOUSEMOVEX ()

Katso myös: [MOUSEMOVEY](#), [MOUSEMOVEZ](#)

ESIMERKKI

```
Repeat

    Text 0, 0, "X: "+MouseX()
    Text 0,10, "Y: "+MouseY()
    Text 0,20, "wheel: "+MouseZ()

    Text 0,40, "XSpeed: "+MouseMoveX()
    Text 0,50, "YSpeed: "+MouseMoveY()
    Text 0,60, "wheelSpeed: "+MouseMoveZ()

    DrawScreen

Until EscapeKey()
```

MOUSEMOVEY

KUVAUS

Tämä funktio kertoo hiiren nopeuden pystysuuntaan. Kätevä silloin, kun haluat esimerkiksi oman hiirikursorin tai ohjaussysteemin. Silloin ruudun koko ei rajoita sijaintia.

Tulos on kokonaisluku.

KÄYTTÖ

MOUSEMOVEY ()

Katso myös: [MOUSEMOVEX](#), [MOUSEMOVEZ](#)

ESIMERKKI

```
Repeat

  Text 0, 0, "X: "+MouseX()
  Text 0,10, "Y: "+MouseY()
  Text 0,20, "wheel: "+MouseZ()

  Text 0,40, "XSpeed: "+MouseMoveX()
  Text 0,50, "YSpeed: "+MouseMoveY()
  Text 0,60, "wheelSpeed: "+MouseMoveZ()

  DrawScreen

Until EscapeKey()
```

MOUSEMOVEZ

KUVAUS

Tämä funktio on kätevä esimerkiksi aseiden vaihtoon peleissä. Funktio palauttaa -1, jos rullaa on pyöritetty alas, ja 1 jos rullaa on pyöritetty ylös. Jos rullaa ei ole pyöritetty ollenkaan, palautetaan nolla.

Tulos on kokonaisluku.

KÄYTTÖ

MOUSEMOVEZ ()

Katso myös: [MOUSEMOVEX](#), [MOUSEMOVEY](#)

ESIMERKKI

```
Repeat

    Text 0, 0, "X: "+MouseX()
    Text 0,10, "Y: "+MouseY()
    Text 0,20, "wheel: "+MouseZ()

    Text 0,40, "XSpeed: "+MouseMoveX()
    Text 0,50, "YSpeed: "+MouseMoveY()
    Text 0,60, "wheelSpeed: "+MouseMoveZ()

    DrawScreen

Until EscapeKey()
```

MOUSEUP

KUVAUS

Tarkastaa onko KLIKKAUS tapahtunut, jolloin palautetaan 1 (ON, True). Muutoin nolla (OFF, False). Vertaa MouseUp:ia ja MouseHit:iä keskenään.

Hiiren tila päivitetään painikkeiden osalta UpdateGame-kutsun aikana (tai DrawScreenissä, jos UpdateGame ei kutsuta). Tämä tarkoittaa sitä, että et voi odottaa pääloopissa tiettyä aikaa ja tarkastaa onko painikkeen tila vieläkin ennallaan.

KÄYTTÖ

MOUSEUP (painike)

- **painike**

1 = vasen painike

2 = oikea painike

3 = keskimäinen painike

Katso myös: [MOUSEDOWN](#), [MOUSEHIT](#), [GETMOUSE](#)

ESIMERKKI

```
AddText "Click the box"
```

```
AddText "Try to hold mouse down on the box, then release"
```

```
Repeat
```

```
    Color cbsilver
```

```
    Box 100,100,200,100
```

```
    'check if the mouse is inside the box
```

```
    If MouseX()>100 And MouseX()<300 And MouseY()>100 And MouseY()<200
```

```
        If MouseUp(1)
```

```
            'draw another (red) box
```

```
            Color cbred
```

```
            Box 100,100,200,100
```

```
        EndIf
```

```
    Else
```

```
        'outside... clear the click state.
```

```
        'You can comment this and see what happens
```

```
        ClearMouse
```

```
    EndIf
```

```
    DrawScreen
```

```
Until EscapeKey()
```

MOUSEWX

KUVAUS

Kertoo hiiren vaakasijainnin pelimaailmassa, eli tämä funktio on objektiyhteensopiva.

Tulos on kokonaisluku.

KÄYTTÖ

MOUSEWX ()

Katso myös: [MOUSEX](#), [MOUSEWY](#)

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

ShowMouse LoadImage("Media\cursor.bmp")

hero= LoadObject("Media\guy.bmp",18)

Repeat

    'move camera with arrows
    TranslateCamera KeyDown(205) -KeyDown(203), KeyDown(208) -KeyDown(200)

    'rotate the game character
    TurnObject hero, -40

    'Lue hiiri
    mx=MouseX()
    my=MouseY()

    Color cbwhite
    Text 10,10,"RuutuX:"+ mx
    Text 10,30,"RuutuY:"+ my

    worldx= MouseWX()
    worldy= MouseWY()

    Color cborange
    Text 10,50,"Mouse World X:"+ worldx
    Text 10,70,"Mouse World Y:"+ worldy

    'check if the coordinates overlap the object
    If worldx>ObjectX(hero) -16 And worldx<ObjectX(hero)+16
        If worldy<ObjectY(hero)+16 And worldy>ObjectY(hero) -16
            'draw a box around the object
            'Remember DRAWTOWORLD!
            Color 0,255,0
            Box ObjectX(hero) -16,ObjectY(hero)+16,32,32,OFF
        EndIf
    EndIf

    DrawScreen

Forever
```


MOUSEWY

KUVAUS

Kertoo hiiren pystysijainnin pelimaailmassa, eli tämä funktio on objektiyhteensopiva.

Tulos on kokonaisluku.

KÄYTTÖ

MOUSEWY ()

Katso myös: [MOUSEY](#), [MOUSEWX](#)

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

ShowMouse LoadImage("Media\cursor.bmp")

hero= LoadObject("Media\guy.bmp",18)

Repeat

    'move camera with arrows
    TranslateCamera KeyDown(205) -KeyDown(203), KeyDown(208) -KeyDown(200)

    'rotate the game character
    TurnObject hero, -40

    'Lue hiiri
    mx=MouseX()
    my=MouseY()

    Color cbwhite
    Text 10,10,"RuutuX:"+ mx
    Text 10,30,"RuutuY:"+ my

    worldx= MouseWX()
    worldy= MouseWY()

    Color cborange
    Text 10,50,"Mouse World X:"+ worldx
    Text 10,70,"Mouse World Y:"+ worldy

    'check if the coordinates overlap the object
    If worldx>ObjectX(hero) -16 And worldx<ObjectX(hero)+16
        If worldy<ObjectY(hero)+16 And worldy>ObjectY(hero) -16
            'draw a box around the object
            'Remember DRAWTOWORLD!
            Color 0,255,0
            Box ObjectX(hero) -16,ObjectY(hero)+16,32,32,OFF
        EndIf
    EndIf

    DrawScreen

Forever
```

Syötteet : funktiot

MOUSEX

KUVAUS

Kertoo hiiren sijainnin vaakasuunnassa.

Tulos ilmoitetaan pikseleinä, kun vasen yläkulma on (0,0).

Tulos on kokonaisluku.

KÄYTTÖ

MOUSEX ()

Katso myös: [MOUSEY](#), [MOUSEWX](#)

ESIMERKKI

Repeat

```
Text 0, 0, "X: "+MouseX()
Text 0,10, "Y: "+MouseY()
Text 0,20, "wheel: "+MouseZ()

Text 0,40, "XSpeed: "+MouseMoveX()
Text 0,50, "YSpeed: "+MouseMoveY()
Text 0,60, "wheelSpeed: "+MouseMoveZ()
```

DrawScreen

Until EscapeKey()

MOUSEY

KUVAUS

Kertoo hiiren sijainnin pystysuunnassa.

Tulos ilmoitetaan pikseleinä, kun vasen yläkulma on (0,0).

Tulos on kokonaisluku.

KÄYTTÖ

MOUSEY ()

Katso myös: [MOUSEX](#), [MOUSEWY](#)

ESIMERKKI

Repeat

```
Text 0, 0, "X: "+MouseX()
Text 0,10, "Y: "+MouseY()
Text 0,20, "wheel: "+MouseZ()

Text 0,40, "XSpeed: "+MouseMoveX()
Text 0,50, "YSpeed: "+MouseMoveY()
Text 0,60, "wheelSpeed: "+MouseMoveZ()
```

DrawScreen

Until EscapeKey()

MOUSEZ

KUVAUS

Kertoo, paljonko hiirullaa on pyöritetty siitä asti, kun ohjelma käynnistettiin. Negatiiviset arvot ovat itseesi päin ja positiiviset arvot itsestäsi pois päin.

KÄYTTÖ

```
MOUSEZ ()
```

Katso myös: [MOUSEMOVEZ](#), [MOUSEX](#), [MOUSEZ](#)

ESIMERKKI

Repeat

```
Text 0, 0, "X: "+MouseX()
Text 0,10, "Y: "+MouseY()
Text 0,20, "wheel: "+MouseZ()

Text 0,40, "XSpeed: "+MouseMoveX()
Text 0,50, "YSpeed: "+MouseMoveY()
Text 0,60, "wheelSpeed: "+MouseMoveZ()
```

DrawScreen

Until EscapeKey()

kamera : **komennot**

MOVECAMERA

KUVAUS

Liikuttaa kameraa eteenpäin sen kulman mukaisesti. Negatiiviset arvot liikuttavat kameraa taaksepäin.

KÄYTTÖ

MOVECAMERA eteen, [sivulle]

- **eteen** = Kokonaisluku tai desimaaliluku. Paljonko kameraa liikutetaan.
- **sivulle** (**valinnainen**) = Kokonaisluku tai desimaaliluku. Paljonko kameraa liikutetaan sivusuunnassa. Oletuksena nolla.

Katso myös: [TRANSLATECAMERA](#)

ESIMERKKI

```
FrameLimit 40

floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

guy=LoadObject("Media\guy.bmp")

Repeat

    'Make the camera run circle
    TurnCamera 2
    MoveCamera 1

    DrawScreen

Until EscapeKey()
```

MOVEOBJECT

KUVAUS

Liikuttaa objektia eteenpäin ja/tai sivulle. Liikutus tapahtuu niin, että CoolBasic huomioi objektin kulman. Siten kallistunut objekti liikkuu nimenomaan viistosti, eikä suoraan vaaka/pystysuunnassa.

Ylhäältä kuvatuissa räiskintäpeleissä voit antaa valinnaisen sivulleastumisarvon.

Negatiivisella arvolla saat objektin liikkumaan taaksepäin tai toiseen sivusuuntaan.

Z-arvon muuttaminen on edistyneemmille käyttäjille. Sen avulla voidaan teeskennellä syvyysvaikutelmaa, mikä saa objektin piirtymään sijaintinsa puolesta perspektiivissä taustaa vasten.

KÄYTTÖ

```
MOVEOBJECT objektimuuttuja, eteen, [sivulle], [Z]
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin
- **eteen** = Lukuarvo, paljonko objekti liikkuu eteenpäin
- **sivulle** (*valinnainen*) = Lukuarvo, paljonko objekti liikkuu sivuttain. Oletuksena nolla
- **Z** (*valinnainen, edistyneemmille*) = Lukuarvo, paljonko objekti liikkuu syvyyssuunnassa. Oletuksena nolla

Katso myös: [TRANSLATEOBJECT](#), [POSITIONOBJECT](#)

ESIMERKKI

```
FrameLimit 40

soldier=LoadObject("Media\soldier.bmp",72)

AddText "Arrows to move"

Repeat

    'Update controls
    If LeftKey() Then TurnObject soldier,5
    If RightKey() Then TurnObject soldier, -5
    If UpKey() Then MoveObject soldier, 2
    If DownKey() Then MoveObject soldier, -2

    DrawScreen

Until EscapeKey()
```

NEW

KUVAUS

Tämä funktio luo kokoelmaan uuden jäsenen. Oletuksellisesti se lisätään listan loppuun. Funktio palauttaa osoittimen tähän uuteen jäseneseen. Osoittimen avulla pääset käsiksi jäsenen muuttujiin \-operaattorilla.

Katso esimerkki. Suosittelemme sinua myös katsomaan [tämän](#).

KÄYTTÖ

NEW (tyyppi_nimi)

- **tyyppi_nimi** = TYPE-kokoelman nimi.

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
Type BULLETS
  Field obj
End Type

'-----
'MAIN PROGRAM
'-----

FrameLimit 40

'load the soldier, turning detail 72
guy=LoadObject("Media\soldier.bmp",72)

'load the bullet (master-object) and hide it
bullet=LoadObject("Media\bullet.bmp")
ShowObject bullet,OFF

ClsColor cbdark

AddText "Arrows to turn, SPACE to shoot"

'MAIN LOOP
Repeat

  'Update turning
  If LeftKey() Then TurnObject guy,5
  If RightKey() Then TurnObject guy, -5

  'Shoot?
  If KeyDown(cbkeyspace) And reload=0

    newBullet.BULLETS= New(BULLETS)
    newBullet\obj=CloneObject(bullet)
    CloneObjectPosition newBullet\obj,guy
    CloneObjectOrientation newBullet\obj,guy
    'put the bullet at the end of the barrel
    MoveObject newBullet\obj,24

    reload=4
  EndIf

  'Update all BULLETS (move them forward)
  i=0
  For iBullet.BULLETS= Each BULLETS

    MoveObject iBullet\obj,6 'speed=6

    'delete bullet from collection if it
    'gets too far away
    If Distance2(iBullet\obj,guy)>200 Then
      DeleteObject iBullet\obj
      Delete iBullet
    EndIf

    i=i+1

  Next iBullet
```

```
'Update reloading  
If reload>0 Then reload=reload-1  
  
Text 30,30,"Bulets to update: "+i  
  
'Don't forget this :)  
DrawScreen
```

```
Until EscapeKey()
```


NEXTOBJECT

KUVAUS

CoolBasic tarjoaa mahdollisuuden käydä kaikki objektit läpi. Samaan tapaan kuin For..Each läpikäy kaikki tyyppin jäsenet tai For..Next läpikäy esimerkiksi kaikki taulukon solut. Lista alustetaan komennolla **InitObjectList**, joka siirtää sisäisen osoittimen listan alkuun, eli ensiksi luotuun objektiin.

Pääset käsiksi itse objekteihin tällä funktiolla. Se palauttaa listassa seuraavan objektin. Kun lista on käyty kokonaan läpi, palauttaa NextObject nollan.

Tämä on kätevä esimerkiksi silloin, kun pitäisi käydä kaikki viholliset tai ammukset läpi (ne tunnistetaan esim. ObjectInteger-funktiolla).

HUOM! Tämä komento ei toimi 3D-tilassa.

KÄYTTÖ

```
NEXTOBJECT ()
```

Katso myös: [INITOBJECTLIST](#)

ESIMERKKI

```
'Create 9 cows and position them into a row
For i=0 To 7
    obj=LoadObject("Media\cow.bmp")
    PositionObject obj, (-4+i)*40,0
Next i

'enable shapes to be rendered into world coordinates
DrawToWorld ON

ClsColor cbdarkred

InitObjectList 'reset the object iteration
Repeat

    If TIMER()>interv+1000 Then
        current=NextObject()
        If current
            drawx=ObjectX(current)
            drawy=ObjectY(current)
        Else
            'all objects have been iterated, start all over
            InitObjectList
        EndIf

        interv=TIMER()
    EndIf

    Color cbpink
    Circle drawx-20,drawy+20,40,OFF

    Text 0,0,current
    DrawScreen

Until EscapeKey()
```

OBJECTANGLE

KUVAUS

Kertoo objektin kulman, eli sen suunnan mihin se osoittaa.

Tulos on desimaaliluku väliltä 0-360.

KÄYTTÖ

OBJECTANGLE (objektimuuttuja)

- **objektimuuttuja** = Se muuttuja, joka sisältää objektin.

Katso myös: [OBJECTX](#), [OBJECTY](#)

ESIMERKKI

```
'ALSO SEE BEGINNER'S LESSON 8

FrameLimit 40

'create an infinite grass scene
lawn=LoadImage("Media\grass.bmp")
grass=MakeObjectFloor()
PaintObject grass,lawn

'load the player
guy=LoadObject("Media\guy.bmp",72)

While Not KeyHit(1) 'esc

    'rotate object
    TurnObject guy, -5

    'move object
    TranslateObject guy,1,1

    'camera follow
    CloneCameraPosition guy

    DrawGame

    'print some statics
    Text 10,10,"X: "+ObjectX(guy)
    Text 10,30,"Y: "+ObjectY(guy)
    Text 10,50,"angle: "+ObjectAngle(guy)

    DrawScreen

Wend
```

OBJECTFLOAT

KUVAUS

Liittää objektiin desimaaliluvun.

Komentoa voi käyttää myös funktiona, jolloin se palauttaa objektin tiedot. Tällöin palautettu luku on desimaaliluku.

Esimerkki näyttäneen parhaiten miten komentohybridiä voi käyttää.

KÄYTTÖ

```
OBJECTFLOAT objektimuuttuja, arvo  
tai  
OBJECTFLOAT (objektimuuttuja)
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **arvo** = Desimaaliluku.

Katso myös: [OBJECTINTEGER](#), [OBJECTSTRING](#)

ESIMERKKI

```
'assign a custom mouse pointer  
cursor=LoadImage("Media\cursor.bmp")  
ShowMouse cursor  
  
guy1=LoadObject("Media\guy.bmp")  
guy2=LoadObject("Media\guy.bmp")  
guy3=LoadObject("Media\guy.bmp")  
  
PositionObject guy2, -100,0  
PositionObject guy3,100,0  
  
'make them pickable (by ball)  
ObjectPickable guy1,2  
ObjectPickable guy2,2  
ObjectPickable guy3,2  
  
'give guy1 some properties  
ObjectInteger guy1,100  
ObjectFloat guy1,PI  
ObjectString guy1,"friend"  
  
'give guy2 some properties  
ObjectInteger guy2,50  
ObjectFloat guy2,45.6  
ObjectString guy2,"enemy"  
  
'give guy3 some properties  
ObjectInteger guy3,25  
ObjectFloat guy3, -0.5  
ObjectString guy3,"neutral"  
  
ClsColor cbwhite  
Color cbblack  
  
'-----  
'THE GAME ITSELF  
Repeat  
  
    CameraPick MouseX(),MouseY()  
  
    e=PickedObject()  
    If e Then  
        'if picking resulted an object, print its properties  
        CenterText ObjectX(e),ObjectY(e) -70,ObjectInteger(e)  
        CenterText ObjectX(e),ObjectY(e) -60,ObjectFloat(e)  
        CenterText ObjectX(e),ObjectY(e) -50,ObjectString(e)  
    EndIf  
  
    DrawScreen  
  
Until KeyHit(1) 'esc
```

OBJECTFRAME

KUVAUS

Palauttaa animoidun objektin nykyisen framen (0 ->).

Tulos on desimaaliluku.

KÄYTTÖ

OBJECTFRAME (objektimuuttuja)

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [PLAYOBJECT](#), [STOPOBJECT](#), [OBJECTPLAYING](#)

ESIMERKKI

```
'make screen a bit zoomed
SCREEN 250,188,0,cbsizable

FrameLimit 40 'limit game speed

'load animation for both directions, but hide them
zeroright=LoadAnimObject("Media\zerorun1.bmp",47,51,0,12)
MaskObject zeroright,cbmagenta
ShowObject zeroright,OFF
zeroleft=LoadAnimObject("Media\zerorun2.bmp",47,51,0,12)
MaskObject zeroleft,cbmagenta
ShowObject zeroleft,OFF

'default direction
zero=CloneObject(zeroright)

Color cbblack
AddText "Try arrows"
ClsColor cbwhite

'MAIN LOOP
Repeat

    'run right!
    If RightKey() Then
        PaintObject zero,zeroright
        PlayObject zero,2,12,0.25 'animate at speed 0.25
    'run left!
    ElseIf LeftKey() Then
        PaintObject zero,zeroleft
        PlayObject zero,2,12,0.25 'animate at speed 0.25
    Else
        'nothing is pressed so freeze
        PlayObject zero,0
    EndIf

    Text 0,140,"Playing? "+ObjectPlaying(zero)
    Text 0,160,"Frame: "+ObjectFrame(zero)

    DrawScreen

Until EscapeKey()
```

OBJECTINTEGER

KUVAUS

Liittää objektiin kokonaisluvun. Esimerkiksi healtin tai jotain muita tietoja.

Komentoa voi käyttää myös funktiona, jolloin se palauttaa objektin tiedot. Tällöin palautettu luku on kokonaisluku.

Esimerkki näyttäneen parhaiten miten komentohybridiä voi käyttää.

KÄYTTÖ

```
OBJECTINTEGER objektimuuttuja, arvo  
tai  
OBJECTINTEGER (objektimuuttuja)
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **arvo** = Kokonaisluku.

Katso myös: [OBJECTFLOAT](#), [OBJECTSTRING](#)

ESIMERKKI

```
'assign a custom mouse pointer  
cursor=LoadImage ("Media\cursor.bmp")  
ShowMouse cursor  
  
guy1=LoadObject ("Media\guy.bmp")  
guy2=LoadObject ("Media\guy.bmp")  
guy3=LoadObject ("Media\guy.bmp")  
  
PositionObject guy2, -100,0  
PositionObject guy3,100,0  
  
'make them pickable (by ball)  
ObjectPickable guy1,2  
ObjectPickable guy2,2  
ObjectPickable guy3,2  
  
'give guy1 some properties  
ObjectInteger guy1,100  
ObjectFloat guy1,PI  
ObjectString guy1,"friend"  
  
'give guy2 some properties  
ObjectInteger guy2,50  
ObjectFloat guy2,45.6  
ObjectString guy2,"enemy"  
  
'give guy3 some properties  
ObjectInteger guy3,25  
ObjectFloat guy3, -0.5  
ObjectString guy3,"neutral"  
  
ClsColor cbwhite  
Color cbblack  
  
' _ _ _ _ _  
'THE GAME ITSELF  
Repeat  
  
    CameraPick mouseX(),mouseY()  
  
    e=PickedObject()  
    If e Then  
        'if picking resulted an object, print its properties  
        CenterText ObjectX(e),ObjectY(e) -70,ObjectInteger(e)  
        CenterText ObjectX(e),ObjectY(e) -60,ObjectFloat(e)  
        CenterText ObjectX(e),ObjectY(e) -50,ObjectString(e)  
    EndIf  
  
    DrawScreen  
  
Until KeyHit(1) 'esc
```

OBJECTLIFE

KUVAUS

Asettaa objektille kuolemantuomion. Annat sille elinajan, jonka jälkeen se tuhotaan automaattisesti.

Elinaikaa ei ilmoiteta sekunteina, vaan piirrettävissä ruuduissa. Siten jos olet esimerkiksi rajoittanut pelinopeuden 40 FPS (**FrameLimit**-komento), elinaika 80 merkitsee kahta sekuntia.

Tämä toiminto sopii sellaisille objekteille, joilla ei juuri ole väliä, eikä niitä muuteta mitenkään (käännetä jne.) Huomaa, että jos yrität käyttää objektia sen kuoleman jälkeen, se kaataa ohjelman.

Komentoa voi käyttää myös funktiona, jolloin se palauttaa objektin jäljellä olevan elinajan kokonaislukuna.

KÄYTTÖ

```
OBJECTLIFE objektimuuttuja, elinaika  
tai  
OBJECTLIFE (objektimuuttuja)
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **elinaika** = Kokonaisluku. Elinaika piirrettävissä ruuduissa.

ESIMERKKI

```
FrameLimit 40  
  
pacman=LoadObject("Media\pacman.bmp")  
ObjectLife pacman,200 '5*40=200  
  
AddText "This object has 5 seconds to live..."  
  
Repeat  
    DrawScreen  
  
Until EscapeKey()
```

OBJECTORDER

KUVAUS

Yleensä objektit piirretään siinä järjestyksessä kun ne on ladattu. Siksi uudemmat piirretään vanhempien päälle. Voit kuitenkin muuttaa tätä järjestystä siirtämällä objektin sisäistä paikkaa jonossa joko listan alkuun tai loppuun.

Kartat kannattaa yleensä siirtää ensimmäiseksi, jos sitä ei ole ladattu ennen kaikkia muita objekteja.

Lattiaobjektien (MakeObjectFloor) piirtojärjestystä ei voi muuttaa, ja ne piirretään aina kaikkien objektien alle.

Komennolle annetaan järjestysparametriksi joko 1 tai -1. -1 siirtää objektin alimmaiseksi, 1 taas päällimmäiseksi.

KÄYTTÖ

OBJECTORDER objektimuuttuja, järjestysparametri

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.

- **järjestysparametri**

-1 = siirrä alimmaiseksi

1 = siirrä päällimmäiseksi

ESIMERKKI

```
car1=LoadObject("Media\car.bmp",8)
car2=LoadObject("Media\car.bmp",8)

RotateObject car1,90

Color cbblack
AddText "Press 1 to put horinzonal car to bottom"
AddText "Press 2 to put horinzonal car to top"

ClsColor cbwhite

Repeat

    If KeyHit(cbkey1) Then ObjectOrder car2, -1
    If KeyHit(cbkey2) Then ObjectOrder car2,1

    DrawScreen

Until EscapeKey()
```

OBJECTPICK

KUVAUS

Objekti suorittaa poiminnan. Poimintaviiva lähtee suoraan objektin osoittamaan suuntaan, ja CoolBasic tarkastaa mihin se osuu. Voit tarkastaa poimitun objektin funktiolla **PickedObject**. Osumapisteen saat funktioista **PickedX** ja **PickedY**.

Komennon ihanteellinen käyttötarkoitus on ns. *instant hit bullets*, eli luodin osuma. Luoti liikkuu äärettömän nopeasti, ja osuu ensimmäiseen seinään tai objektiin. Kaikki "poimittavat" objektit pitää esitellä komennolla **ObjectPickable**.

Poiminnat suoritetaan täysin matemaattisesti, mikä tekee niistä erittäin nopeita. Itseensä ei voi osua, joten objektin sisältä lähtenyt luoti ei töksähdä omaan ulkokuoreen.

Huomaa, että piilotettuja objekteja ei voi poimia.

KÄYTTÖ

OBJECTPICK objektimuuttuja

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [OBJECTPICKABLE](#), [PIXELPICK](#), [CAMERAPICK](#), [PICKEDOBJECT](#), [PICKEDX](#), [PICKEDY](#)

ESIMERKKI

```
'This is for the gun-splash
Type EFFECTS
  Field x
  Field y
  Field size
End Type

DrawToWorld ON 'draw circles onto game world
FrameLimit 40 'limit speed for fast computers

'Load the map and the hero
map=LoadMap("Media\cdm2.til","Media\tileset.bmp")
soldier=LoadObject("Media\soldier.bmp",360)

RotateObject soldier,90 'make the soldier face upwards
SetupCollision soldier,map,1,4,2 'create sliding collision
ObjectRange soldier,30 'set collision range for radius 15

ObjectPickable map,ON 'make map pickable
PlayObject map,0,0,1 'animate map

'Load and set a custom mouse pointer (the crosshair)
crosshair=LoadImage("Media\crosshair.bmp")
ShowMouse crosshair

AddText "Move: WASD, Aim: Mouse, Shoot: Mouse1"

Repeat
  'Get the angle to the crosshair and turn to it
  aim=GetAngle(200,150,MouseX(),MouseY())
  RotateObject soldier,aim

  'Update Controls
  If KeyDown(cbkeys) Then TranslateObject soldier,0,-3
  If KeyDown(cbkeyw) Then TranslateObject soldier,0,3
  If KeyDown(cbkeyd) Then TranslateObject soldier,3,0
  If KeyDown(cbkeya) Then TranslateObject soldier,-3,0

  'Update shooting
  If MouseDown(1) And reload=0 Then
    ObjectPick soldier
    If PickedObject() Then
      effect.EFFECTS=New(EFFECTS)
      effect\x=PickedX()
      effect\y=PickedY()
      effect\size=2
    EndIf
    reload=6
  EndIf
  If reload Then reload=reload-1

  'draw objects and then EFFECTS
```



```

DrawGame
GoSub updateEFFECTS

'stick the camera to the soldier and draw screen
UpdateGame
CloneCameraPosition soldier
DrawScreen

Until EscapeKey()
End

':::::::::::::::::::::::::::::::::::::::::::

'This sub-program will update all existing EFFECTS
updateEFFECTS:

'Iterate all
For effect.EFFECTS=Each EFFECTS

    'Draw efefct
    Color chyellow
    Circle effect\x-effect\size, effect\y+effect\size, effect\size*2, OFF

    'Update effect size
    effect\size=effect\size+1

    'Remove effect when it gets too large
    If effect\size>10 Then Delete effect

Next effect

Return

```

OBJECTPICKABLE

KUVAUS

Asettaa objektin poimittavaksi. Tällöin **CameraPick** ja **ObjectPick** tunnistavat objektin.

Voit valita erilaisista poiminta-tavoista. Laatikkopoiminta perustuu objektin ympärillä olevaan näkymättömään laatikkoon. Normaalisti se on objektin alkuperäinen koko pikseleinä, mutta sitä voi vaihtaa komennolla **ObjectRange**. Pallo-poiminta käyttää myös objektin halkaisijaa, mutta saattaa olla tarkempi esimerkiksi ylhäältä kuvattujen pelihahmojen kohdalla. Pikselintarkka poiminta on kaikkein tarkin poimintatyyppi. Joskin myös kaikkein raskain.

Kartta-objektille suoritetaan aina tilentarkka poiminta, joten et voi määritellä tyyppiä erikseen. Komenna kartoille vain **ObjectPickable kartta,ON**.

Poiminnat suoritetaan täysin matemaattisesti, mikä tekee niistä erittäin nopeita.

Huomaa, että piilotettuja objekteja ei voi poimia. Poiminnan saa pois päältä esittämällä poimintatyyppiä nollan.

KÄYTTÖ

OBJECTPICKABLE objektimuuttuja, poimintatyyppi

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **poimintatyyppi**

OFF = ei poimintaa

1 = laatikkopoiminta

2 = pallo-poiminta

3 = pikselintarkka poiminta

Katso myös: [CAMERAPICK](#), [OBJECTPICK](#), [PIXELPICK](#), [PICKEDOBJECT](#), [PICKEDX](#), [PICKEDY](#)

ESIMERKKI

```
FrameLimit 40

'Enable graphics commands to cast shapes to
'the game world
DrawToWorld ON

pacman1=LoadObject("Media\pacman.bmp",360)
pacman2=CloneObject(pacman1)
pacman3=CloneObject(pacman1)

ObjectPickable pacman1,1 'box-pick
ObjectPickable pacman2,2 'circle-pick
ObjectPickable pacman3,3 'pixel-perfect

PositionObject pacman1, -100,0
PositionObject pacman3, 100,0

'Load the shooter
guy=LoadObject("Media\guy.bmp",180)
PositionObject guy,0, -100

Color cbblack
AddText "Guide via arrows"
Locate 70,80
AddText "box-pick"
Locate 160,80
AddText "circle-pick"
Locate 260,80
AddText "pixel-perfect"
Color cbblack
ClsColor cbwhite

Repeat

  'Update controls
  If LeftKey() Then TurnObject guy,2
  If RightKey() Then TurnObject guy, -2
  If UpKey() Then MoveObject guy,2
  If DownKey() Then MoveObject guy, -2

  'Make them spin, add some excitement
  TurnObject pacman1,1
  TurnObject pacman2,1
  TurnObject pacman3,1
```

```
'Pick
ObjectPick guy

DrawGame

'Something WAS picked, let's see what it is
e=PickedObject()
Color cbblack
If e Then
    Color cbred
    If e=pacman1 Then
        Text 0,30,"pacman on the left"
    ElseIf e=pacman2 Then
        Text 0,30,"pacman at the center"
    ElseIf e=pacman3 Then
        Text 0,30,"pacman on the right"
    EndIf
    Circle PickedX() -5,PickedY()+5,10,ON
EndIf

'draw the check-zones
Box ObjectX(pacman1) -32,ObjectY(pacman1)+32,64,64,OFF
Circle ObjectX(pacman2) -32,ObjectY(pacman2)+32,64,OFF

DrawScreen

Until EscapeKey()
```

OBJECTPLAYING

KUVAUS

Palauttaa 1, jos objektia animoidaan. Muutoin nolla.

Voit käyttää tätä funktiota määrittelemään esimerkiksi onko animaatio soitettu loppuun.

KÄYTTÖ

OBJECTPLAYING (objektimuuttuja)

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [PLAYOBJECT](#), [STOPOBJECT](#), [OBJECTFRAME](#)

ESIMERKKI

```
'make screen a bit zoomed
SCREEN 250,188,0,cbsizable

FrameLimit 40 'limit game speed

'load animation for both directions, but hide them
zeroright=LoadAnimObject("Media\zerorun1.bmp",47,51,0,12)
MaskObject zeroright,cbmagenta
ShowObject zeroright,OFF
zeroleft=LoadAnimObject("Media\zerorun2.bmp",47,51,0,12)
MaskObject zeroleft,cbmagenta
ShowObject zeroleft,OFF

'default direction
zero=CloneObject(zeroright)

Color cbblack
AddText "Try arrows"
ClsColor cbwhite

'MAIN LOOP
Repeat

    'run right!
    If RightKey() Then
        PaintObject zero,zeroright
        PlayObject zero,2,12,0.25 'animate at speed 0.25
    'run left!
    ElseIf LeftKey() Then
        PaintObject zero,zeroleft
        PlayObject zero,2,12,0.25 'animate at speed 0.25
    Else
        'nothing is pressed so freeze
        PlayObject zero,0
    EndIf

    Text 0,140,"Playing? "+ObjectPlaying(zero)
    Text 0,160,"Frame: "+ObjectFrame(zero)

    DrawScreen

Until EscapeKey()
```

OBJECTRANGE

KUVAUS Asettaa objektin törmäys-ulottuvuuden. Kun objekti ladataan, sille määritellään törmäysulottuvuus automaattisesti sen koon mukaan. Voit kuitenkin muuttaa tätä arvoa niin, että se vastaa toiveitasi. Yleensä sinun ei tarvitse tehdä tätä. Komennotlle välitetään kaksi parametria. Jos toinen jätetään kirjoittamatta, CoolBasic olettaa sen samaksi kuin ensimmäinen. Laatikotörmäyksessä tarvitaan kumpaakin, mutta pallotörmäyksessä pelkkä ympyrän halkaisija (ensimmäinen parametri) riittää. Ellipsin muotoiset törmäyspallot eivät ole tuettuja.
KÄYTTÖ OBJECTRANGE objektimuuttuja, x-halkaisija, [y-halkaisija] • objektimuuttuja = Se muuttuja, mihin objekti ladattiin tai luotiin. • x-halkaisija = Laatikotörmäyksen leveys tai pallotörmäyksen halkaisija. • y-halkaisija (<i>valinnainen</i>) = Laatikotörmäyksen korkeus. Pois jätettynä saman suuruinen x-halkaisijan kanssa.
ESIMERKKI

OBJECTSIGHT

KUVAUS

Palauttaa 1, jos kaksi objektia näkevät toisensa. Muutoin nolla.

Ihanteellinen käyttötarkoitus on havaita milloin vihollinen näkee pelaajan. Kulman ja etäisyyden laskeminen yhdessä tämän kanssa mahdollistaa vihollisen "näkökentän", jota voit itse hallita.

Tarkastus käydään tilekartan kanssa. Muita objekteja ei toistaiseksi huomioida. Lisäksi tarkastus on täysin matemaattinen, ja siksi erittäin nopea.

KÄYTTÖ

OBJECTSIGHT (obj1, obj2)

- obj1 = Ensimmäinen objekti.
- obj2 = Toinen objekti.

ESIMERKKI

```
FrameLimit 40
DrawToWorld ON

'Load the map
map=LoadMap("Media\cdm2.til","Media\tileset.bmp")

'Load player
soldier=LoadObject("Media\soldier.bmp",72)
ObjectRange soldier,32

'Load dumb immobile bot
bot=LoadObject("Media\soldier.bmp")
ObjectRange bot,32
PositionObject bot,50,200

'setup some collisions
SetupCollision soldier,bot,2,2,2
SetupCollision soldier,map,1,4,2

Repeat

    'Update controls
    If LeftKey() Then TurnObject soldier,5
    If RightKey() Then TurnObject soldier, -5
    If UpKey() Then MoveObject soldier,3.5
    If DownKey() Then MoveObject soldier, -2

    'Camera follow
    CloneCameraPosition soldier

    'Enable additional graphics
    DrawGame

    'draw line of sight
    Color cborange
    Line ObjectX(bot),ObjectY(bot),ObjectX(soldier),ObjectY(soldier)

    'determine if players can see each other
    Color cbwhite
    If ObjectSight(soldier,bot) Then
        Text 0,0,"LINE OF SIGHT!"
    EndIf

    'render SCREEN
    DrawScreen

Until EscapeKey()
```

OBJECTSIZEX

KUVAUS

Kertoo objektin leveyden pikseleissä. Huomaa, että luvut eivät vastaa alkuperäisen kuvan kokoa. CoolBasic ottaa huomioon, että objektia on käännetty, ja palauttaa hetkitäsmällisen leveyden.

Kartta-objektin kohdalla palautetaan koko kartan leveys pikseleissä. Jos haluat saada kartan leveyden tileinä, katso **MapWidth** ja **MapHeight**.

Tulos on kokonaisluku.

Objektin korkeuden saat funktiolla **ObjectSizeY**.

KÄYTTÖ

```
OBJECTSIZEX (objektimuuttuja)
```

- **objektimuuttuja** = Se muuttuja, joka sisältää objektin.

Katso myös: [OBJECTSIZEY](#)

ESIMERKKI

```
FrameLimit 40

'load the player
guy=LoadObject("Media\guy.bmp",72)

AddText "Turn object with the arrows."

While Not KeyHit(1) 'esc

    If LeftKey() Then TurnObject guy,5
    If RightKey() Then TurnObject guy, -5

    DrawGame

    'print some statics
    Text 10,20,"Width: "+ObjectSizeX(guy)
    Text 10,40,"Height: "+ObjectSizeY(guy)
    Text 10,60,"angle: "+ObjectAngle(guy)

    DrawScreen

Wend
```

OBJECTSIZEY

KUVAUS

Kertoo objektin korkeuden pikseleissä. Huomaa, että luvut eivät vastaa alkuperäisen kuvan kokoa. CoolBasic ottaa huomioon, että objektia on käännetty, ja palauttaa hetkitäsmällisen korkeuden.

Kartta-objektin kohdalla palautetaan koko kartan korkeus pikseleissä. Jos haluat saada kartan korkeuden tileinä, katso **MapWidth** ja **MapHeight**.

Tulos on kokonaisluku.

Objektin leveyden saat funktiolla **ObjectSizeX**.

KÄYTTÖ

```
OBJECTSIZEY (objektimuuttuja)
```

- **objektimuuttuja** = Se muuttuja, joka sisältää objektin.

Katso myös: [OBJECTSIZEX](#)

ESIMERKKI

```
FrameLimit 40

'load the player
guy=LoadObject("Media\guy.bmp",72)

AddText "Turn object with the arrows."

While Not KeyHit(1) 'esc

    If LeftKey() Then TurnObject guy,5
    If RightKey() Then TurnObject guy, -5

    DrawGame

    'print some statics
    Text 10,20,"Width: "+ObjectSizeX(guy)
    Text 10,40,"Height: "+ObjectSizeY(guy)
    Text 10,60,"angle: "+ObjectAngle(guy)

    DrawScreen

Wend
```


OBJECTSOVERLAP

KUVAUS

Kertoo, ovatko kaksi objektia millään tavoin kosketuksissa toistensa kanssa.

Laatikkokosketus

Objektien päällä on näkymätön laatikko, joka rajaa objektit samaan tapaan kuin niiden kuvarajat. Tämä voi olla epätarkka joidenkin objektien kohdalla (kts. komennon **ImagesCollide** esimerkki).

Pallokosketus

Objektien päällä on näkymätön pallo. Kosketustapa vastaa kahden biljardipallon törmäystä. Tähän vaikuttaa objektien halkaisijat, joita voidaan vaihtaa komennolla **ObjectRange**. Tämä on suositeltava kosketustapa, ja yleensä tarkka.

Pikselikosketus

Kaikkein tarkin kosketustapa. Tarkastus on pikselintarkka, ja se ottaa huomioon objektien kulmat ja jopa animaatio-framen. Tarkastus on myös kaikkein raskain.

Palauttaa 1, jos päällekkäisyyttä esiintyy, muutoin nolla.

KÄYTTÖ

```
OBJECTSOVERLAP (obj1, obj2, [tapa])
```

- **obj1** = Ensimmäinen objekti.
- **obj2** = Toinen objekti.
- **tapa** (valinnainen)
 - 1** = Laatikkokosketus (oletus)
 - 2** = Pallokosketus
 - 3** = Pikselikosketus

ESIMERKKI

```
FrameLimit 40
ShowMouse OFF
DrawToWorld ON

soldier=LoadObject("Media\soldier.bmp",360)
ObjectRange soldier,46

target=LoadObject("Media\crosshair.bmp")

'some text
Color cbblack
AddText "1 = Box-based"
AddText "2 = Circle-based"
AddText "3 = Pixel-based"

'default
method=1

ClsColor cbwhite

Repeat

    'Change mode
    If KeyHit(cbkey1) Then method=1
    If KeyHit(cbkey2) Then method=2
    If KeyHit(cbkey3) Then method=3

    TurnObject soldier,1

    Select method
        Case 1 'box
            Box ObjectX(target) -8,ObjectY(target)+8,16,16,OFF
            Box ObjectX(soldier) -23,ObjectY(soldier)+23,46,46,OFF
            Text 100,250,"Overlap (box): "+ObjectsOverlap(soldier,target,1)
        Case 2 'circle
            Circle ObjectX(target) -8,ObjectY(target)+8,16,OFF
            Circle ObjectX(soldier) -23,ObjectY(soldier)+23,46,OFF
            Text 100,250,"Overlap (circle): "+ObjectsOverlap(soldier,target,2)
        Case 3 'pixel
            Text 100,250,"Overlap (pixel): "+ObjectsOverlap(soldier,target,3)
    End Select

    PositionObject target,MouseX() -200, -MouseY()+150

DrawScreen
```

Until **EscapeKey**()

OBJECTSTRING

KUVAUS

Liittää objektiin merkkijonon. Esimerkiksi nimen tai tiedon siitä, onko se ystävä vai vihollinen.

Komentoa voi käyttää myös funktiona, jolloin se palauttaa objektin tiedot. Tällöin tulos on merkkijono.

Esimerkki näyttäneen parhaiten miten komentohybridä voi käyttää.

KÄYTTÖ

```
OBJECTSTRING objektimuuttuja, arvo  
tai  
OBJECTSTRING (objektimuuttuja)
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **arvo** = Desimaaliluku.

Katso myös: [OBJECTINTEGER](#), [OBJECTFLOAT](#)

ESIMERKKI

```
DrawToWorld ON,ON,ON 'draw text to world coordinates  
'assign a custom mouse pointer  
cursor=LoadImage("Media\cursor.bmp")  
ShowMouse cursor  
  
guy1=LoadObject("Media\guy.bmp")  
guy2=LoadObject("Media\guy.bmp")  
guy3=LoadObject("Media\guy.bmp")  
  
PositionObject guy2, -100,0  
PositionObject guy3,100,0  
  
'make them pickable (by ball)  
ObjectPickable guy1,2  
ObjectPickable guy2,2  
ObjectPickable guy3,2  
  
'give guy1 some properties  
ObjectInteger guy1,100  
ObjectFloat guy1,PI  
ObjectString guy1,"friend"  
  
'give guy2 some properties  
ObjectInteger guy2,50  
ObjectFloat guy2,45.6  
ObjectString guy2,"enemy"  
  
'give guy3 some properties  
ObjectInteger guy3,25  
ObjectFloat guy3, -0.5  
ObjectString guy3,"neutral"  
  
ClsColor cbwhite  
Color cbblack  
  
'- - - - -  
'THE GAME ITSELF  
Repeat  
  
    CameraPick mouseX(),mouseY()  
  
    e=PickedObject()  
    If e Then  
        'if picking resulted an object, print its properties  
        CenterText ObjectX(e),ObjectY(e) -70,ObjectInteger(e)  
        CenterText ObjectX(e),ObjectY(e) -60,ObjectFloat(e)  
        CenterText ObjectX(e),ObjectY(e) -50,ObjectString(e)  
    EndIf  
  
    DrawScreen  
  
Until KeyHit(1) 'esc
```

OBJECTX

KUVAUS

Kertoo objektin X-sijainnin (vaakaan) pelimaailman sisällä. Huomaa, että pelimaailman koordinaatit menevät näin:

□

Tulos on desimaaliluku.

Objektin Y-sijainnin saat funktiolla **ObjectY**.

KÄYTTÖ

OBJECTX (objektimuuttuja)

- **objektimuuttuja** = Se muuttuja, joka sisältää objektin.

Katso myös: [OBJECTY](#), [OBJECTANGLE](#)

ESIMERKKI

```
'ALSO SEE BEGINNER'S LESSON 8

FrameLimit 40

'create an infinite grass scene
lawn=LoadImage("Media\grass.bmp")
grass=MakeObjectFloor()
PaintObject grass,lawn

'load the player
guy=LoadObject("Media\guy.bmp",72)

While Not KeyHit(1) 'esc

    'rotate object
    TurnObject guy, -5

    'move object
    TranslateObject guy,1,1

    'camera follow
    CloneCameraPosition guy

    DrawGame

    'print some statics
    Text 10,10,"X: "+ObjectX(guy)
    Text 10,30,"Y: "+ObjectY(guy)
    Text 10,50,"angle: "+ObjectAngle(guy)

    DrawScreen

Wend
```

OBJECTY

KUVAUS

Kertoo objektin Y-sijainnin (pystyy) pelimaailman sisällä. Huomaa, että pelimaailman koordinaatit menevät näin:

□

Tulos on desimaaliluku.

Objektin X-sijainnin saat funktiolla **ObjectX**.

KÄYTTÖ

OBJECTY (objektimuuttuja)

- **objektimuuttuja** = Se muuttuja, joka sisältää objektin.

Katso myös: [OBJECTX](#), [OBJECTANGLE](#)

ESIMERKKI

```
'ALSO SEE BEGINNER'S LESSON 8

FrameLimit 40

'create an infinite grass scene
lawn=LoadImage("Media\grass.bmp")
grass=MakeObjectFloor()
PaintObject grass,lawn

'load the player
guy=LoadObject("Media\guy.bmp",72)

While Not KeyHit(1) 'esc

    'rotate object
    TurnObject guy, -5

    'move object
    TranslateObject guy,1,1

    'camera follow
    CloneCameraPosition guy

    DrawGame

    'print some statics
    Text 10,10,"X: "+ObjectX(guy)
    Text 10,30,"Y: "+ObjectY(guy)
    Text 10,50,"angle: "+ObjectAngle(guy)

    DrawScreen

Wend
```

OPENTOEDIT

KUVAUS

Avaa tiedoston lukemista JA kirjoittamista varten (tunnetaan myös nimellä *hajasaantikäsittely*). Jos tiedostoa ei ole olemassa, se luodaan. OpenToEdit palauttaa nollan, jos tiedostoa ei voitu kirjoittaa. Mahdollisia syitä ovat levytilan loppuminen, kirjoitussuojaus tai että kyseessä on kansio.

Käytä tätä komentoa tiedoston *päivittämiseen*. Tämä on erityisen hyvä tapa muuttaa tiedoston sisältöä ilman, että pitää kirjoittaa koko tiedosto tai sen loppu uusiksi.

SeekFile ja FileOffset toimivat tässä tilassa.

KÄYTTÖ

OPENTOEDIT (tiedostonimi)

- **tiedostonimi** = Merkkijono, joka määrittää avattavan tiedoston nimen. Isoilla ja pienillä kirjaimilla ei ole eroa.

Katso myös: [OPENTOREAD](#), [OPENTOWRITE](#)

ESIMERKKI

```
'open the file for output and write 10 integers to it
f=OpenToWrite("Media\temp.dat")
  For i=1 To 10
    WriteInt f,i
  Next i
CloseFile f

'- ---

'open the file for editing
f=OpenToEdit("Media\temp.dat")
'replace the 6th Integer with a New value
  SeekFile f, 4*5
  WriteInt f,12345
CloseFile f

'- ---

'open the file for input and print all integers from it
f=OpenToRead("Media\temp.dat")
  For i=1 To 10
    val=ReadInt(f)
    Text 10,i*12, val
  Next i
CloseFile f

DrawScreen

WaitKey
```

OPENTOREAD

KUVAUS

Avaa tiedoston lukemista varten. Kirjoittaminen ei ole sallittua. Jos tiedoston avaaminen epäonnistui, tämä funktio palauttaa nollan. Näin käy esimerkiksi kun tiedostoa ei ole olemassa, tai se onkin kansio.

KÄYTTÖ

```
OPENTOREAD (tiedostonimi)
```

- **tiedostonimi** = Merkkijono, joka määrittää avattavan tiedoston nimen. Isoilla ja pienillä kirjaimilla ei ole eroa.

Katso myös: [READBYTE](#), [READSHORT](#), [READINT](#), [READFLOAT](#), [READSTRING](#), [READLINE](#), [OPENTOWRITE](#), [OPENTOEDIT](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
    lifes=8

    WriteByte f,8
    WriteShort f,lifes*9
    WriteInt f, -8901
    WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

    mybyte=ReadByte(f)
    myshort=ReadShort(f)
    myint=ReadInt(f)
    myfloat=ReadFloat(f)

    Text 10,10,mybyte
    Text 10,30,myshort
    Text 10,50,myint
    Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```

OPENTOWRITE

KUVAUS

Avaa tiedoston kirjoittamista varten. Jos tiedostoa ei ole olemassa, se luodaan. OpenToWrite palauttaa nollan, jos tiedostoa ei voitu kirjoittaa. Mahdollisia syitä ovat levytilan loppuminen, kirjoitussuojaus tai että kyseessä on kansio.

KÄYTTÖ

OPENTOWRITE (tiedostonimi)

- **tiedostonimi** = Merkkijono, joka määrittää avattavan tiedoston nimen. Isoilla ja pienillä kirjaimilla ei ole eroa.

Katso myös: [WRITEBYTE](#), [WRITESHORT](#), [WRITEINT](#), [WRITEFLOAT](#), [WRITESTRING](#), [WRITELINE](#), [OPENTOREAD](#), [OPENTOEDIT](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
    lifes=8

    WriteByte f,8
    WriteShort f,lifes*9
    WriteInt f, -8901
    WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

    mybyte=ReadByte(f)
    myshort=ReadShort(f)
    myint=ReadInt(f)
    myfloat=ReadFloat(f)

    Text 10,10,mybyte
    Text 10,30,myshort
    Text 10,50,myint
    Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```


PAINTOBJECT

KUVAUS

Tämä on monikäyttöinen komento:

Maalaa objekteja toisella objektilla

Voit "uudelleenteksturoida" objektin toisella. Alkuperäinen objekti voi olla piilotettu, ja näin voit muuttaa objektin ulkonäköä kertaheitolla. Täydellinen esimerkiksi skinien vaihtoon lennossa tai vaikkapa pelaajan muodonmuutokseen.

Maalaa objekteja kuvalla

Voit "uudelleenteksturoida" objektin aiemmin ladatulla kuvalla. Näin voit kätevästi luoda objektia lennossa ilman, että niitä ladataan mistään. Luot vain kuvan (MakeImage), piirrät siihen, ja päällystät objektin sillä. Huomaa, että kuvamuuttujaa esitettäessä sen eteen pitää kirjoittaa miinus, esim. PaintObject tyhjä,-imgPäällyste

Lattian päällystäminen

Kun luot lattian funktiolla **MakeObjectFloor**, se on aluksi näkymätön. Se pitää siis päällystää jollakin kuvalla. Tässä tapauksessa käytät kuvamuuttujaa, ja haluttu kuva pitää olla jo valmiiksi ladattuna muistiin funktiolla **LoadImage**.

Partikkelilähteen muuttaminen

Voit muuttaa partikkelilähteen suoltamien partikkelien kuvaa. Näin lumisade muuttuukin tulisateeksi! Käytät olemassaolevaa kuvamuuttujaa, joten kuva pitää olla jo valmiiksi ladattuna muistiin.

Tilesetin muuttaminen

Voit muuttaa tilekartan (MAP-komennot) tilesettiä lennossa ilman, että sitä ladataan hitaasti kovalevyiltä. Käytät jo valmiiksi ladattua kuvamuuttujaa, joka sisältää uuden tilesetin. Tällä tavalla voit käyttää esimerkiksi hienoa negatiivi-välähdystä tai yö-sävytettyä tilekarttaa, joka vaihtuu esimerkiksi kellonajan mukaan.

Huomaa, että master-objektin päällystäminen vaikuttaa kaikkiin siitä tehtyihin klooneihin.

KÄYTTÖ

PAINTOBJECT objektimuuttuja, päällyste

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **päällyste** = Päällystettävän objektin tyylistä riippuen joko toinen objekti tai kuvamuuttuja, joka sisältää päällysteen tai tilesetin.

ESIMERKKI

```
'load a map
map=LoadMap ("Media\cdm2.til", "Media\tileset.bmp")

'load an alternative tileset
newset1=LoadImage ("Media\negative.bmp")
newset2=LoadImage ("Media\tileset.bmp")

'add some info onto screen
Color cborange
AddText "Press SPACE to change tileset"

Repeat

    'change the look?
    If KeyHit(cbkeyspace) Then
        set=Not set
        If set Then
            PaintObject map,newset1
        Else
            PaintObject map,newset2
        EndIf
    EndIf

    'don't forget this!
    DrawScreen

Until EscapeKey()
```

PARTICLEANIMATION

KUVAUS

Tällä komennolla voidaan asettaa partikkelikuvan animoitavuus. Partikkelit animoituvat aina niin, että kuvasarja jakautuu nollasta siihen hetkeen kun partikkelin elämä loppuu. Animaatio soitetaan siis vain kerran, ja sen nopeus riippuu partikkelin eliniästä.

Jotta ominaisuutta voitaisiin käyttää, partikkeleita luodessa (MakeEmitter) on annettava *ANIMOITU* kuva, eli kuva, joka on ladattu LoadAnimImage:lla. ParticleAnimation-komennolle taas kerrotaan kuvasarjan framejen määrä. Sinun täytyy varmistaa, että annat kuvasarjasta täsmälleen oikean koon. Ylimenevät frameet kaatavat ohjelman.

KÄYTTÖ

PARTICLEANIMATION partikkelimuuttuja, framet

- **partikkelimuuttuja** = Se muuttuja, mihin partikkelilähde luotiin.
- **framet** = Kokonaisluku, joka määrittää kuvasarjan pituuden. Ei saa ylittää vastaavaa lukua, kun kuva ladattiin.

Katso myös: [PARTICLEMOVEMENT](#), [PARTICLEEMISSION](#)

ESIMERKKI

```
FrameLimit 40

'8 FRAMES!!!
img=LoadAnimImage ("Media\animparticle.bmp",11,11,0,8)

par=MakeEmitter (img,80)

'speed 5, gravity 0.1
ParticleMovement par,5,0.1

'frequency 5, amount 4, chaos 10 degrees
ParticleEmission par,5,4,10

'animate particles, use 8 frames
ParticleAnimation par,8

RotateObject par,45
PositionObject par, -120, -50

Repeat

    DrawScreen

Until EscapeKey()
```

PARTICLEEMISSION

KUVAUS

Partikkelilähteellä on muutama ominaisuus, joita voi muuttaa vaikka lennossa. Tällä komennolla säädetään emissiotiheyttä, emissiomäärää ja emissiolevitystä.

Emissiotiheys on se aikaväli, jonka välein uusi suoltaminen tapahtuu. Mitä pienempi väli sitä tiheämpään partikkeleita luodaan. Tämä on tärkein arvo.

Emissiomäärä on uusien partikkelien lukumäärä. Aina kun lähde saavuttaa emissiotiheydensä, luodaan tämän verran uusia partikkeleita. Kokeile vaikka säätää emissiotiheys erittäin suureksi ja tätä arvoa suuremmaksi.

Emissiolevitys määrittää asteina sellaisen sektorin, jonka alueelle uudet partikkelit suunnataan. Oletuksena 180, joka sinkoaa partikkeleita kaikkiin suuntiin. Arvo 90 sinkoaa partikkeleita vain puoliympyrän verran, ja nolla sinkoaa partikkeleita tiheänä ketjuna vain määrättyyn suuntaan (siihen suuntaan, mihin lähdeobjekti osoittaa).

KÄYTTÖ

PARTICLEEMISSION partikkelimuuttuja, tiheys, määrä, levitys

- **partikkelimuuttuja** = Se muuttuja, mihin partikkelilähde luotiin.
- **tiheys** = Kokonaisluku. Kuinka tiheällä tahdilla partikkeleita singotaan. Pienempi on enemmän.
- **määrä** = Kokonaisluku. Paljonko uusia partikkeleita generoidaan kerralla.
- **levitys** = Kokonaisluku. Kaaos-arvo, joka määrittää sinkoilevien partikkeleiden sektorin koon. Arvo 0-180.

Katso myös: [PARTICLEANIMATION](#), [PARTICLEMOVEMENT](#)

ESIMERKKI

```
FrameLimit 40

img=LoadImage("Media\particle.bmp")

par=MakeEmitter(img,120) 'lifetime for 3 seconds

'speed 5, gravity 0.1
ParticleMovement par,5,0.1

'frequency 5, amount 4, chaos [-60 -> 60]
ParticleEmission par,5,4,60

'rotate the emitter to face upwards
RotateObject par,90

Repeat

    DrawScreen

Until EscapeKey()
```

PARTICLEMOVEMENT

KUVAUS

Tällä komennolla säädetään partikkelilähteen suoltamien partikkelien liikkuvuuden ominaisuuksia.

Nopeus määrittää kuinka nopeasti partikkelit lähtevät lähteestä. Yksikkö on pikseli. Painovoima taas saa partikkelit tippumaan alas. Painovoimaksi kelpaa lähtönopeudesta riippuen pieni desimaaliarvo väliltä 0.02 -> 0.4. Painovoima ja nopeus voivat olla myös negatiivisia. Negatiivinen nopeus sinkoaa partikkeleita lähteen taakse. Negatiivinen painovoima kiihdyttää partikkeleja kohti taivasta.

Voit halutessasi määrittää valinnaisen kiihtyvyyden. Ykköstä pienemmät arvot hidastavat partikkeleita, kunnes pysähtyvät kokonaan. Ykköstä suuremmat arvot taas kiihdyttävät partikkeleita joka kerta lujempaan vauhtiin. Oletuksena on yksi, mitä ei vaikuta kiihdyttävästi mitenkään. Sopivia arvoja ovat esim. **0.9** tai **0.95** tai **0.97**. Kokeile muuttaa esimerkin rivi tällaiseksi:
[ParticleMovement par,5,0,0.95](#)

KÄYTTÖ

```
PARTICLEMOVEMENT partikkelimuuttuja, nopeus, painovoima, [kiihtyvyys]
```

- **partikkelimuuttuja** = Se muuttuja, mihin partikkelilähde luotiin.
- **nopeus** = Desimaaliluku. Partikkelien lähtönopeus.
- **painovoima** = Desimaaliluku. Partikkeleihin vaikuttava painovoima. Nolla=ei painovoimaa.
- **kiihtyvyys** (*valinnainen*) = Desimaaliluku. Säätelee kiihtyvyyttä/hidastuvuutta. Oletus 1 (ei kiihtyvyyttä).

Katso myös: [PARTICLEANIMATION](#), [PARTICLEEMISSION](#)

ESIMERKKI

```
FrameLimit 40

img=LoadImage("Media\particle.bmp")

par=MakeEmitter(img,120) 'lifetime for 3 seconds

'speed 5, gravity 0.1
ParticleMovement par,5,0.1

'frequency 5, amount 4, chaos [-60 -> 60]
ParticleEmission par,5,4,60

'rotate the emitter to face upwards
RotateObject par,90

Repeat

    DrawScreen

Until EscapeKey()
```

PEEKBYTE

KUVAUS

Lukee muistipalasta yhden tavun, eli kokonaisluvun väliltä 0-255.

KÄYTTÖ

PEEKBYTE (muistipala, paikka)

- **muistipala** = Se muuttuja, mihin muistipala luotiin.
- **paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.

Katso myös: [PEEKSHORT](#), [PEEKINT](#), [PEEKFLOAT](#), [POKEBYTE](#), [POKESHORT](#), [POKEINT](#), [POKEFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMblock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```

PEEKFLOAT

KUVAUS

Lukee muistipalasta neljätavuisen (32-bittisen) desimaaliluvun.

KÄYTTÖ

PEEKFLOAT (muistipala, paikka)

- muistipala** = Se muuttuja, mihin muistipala luotiin.
- paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.

Katso myös: [PEEKBYTE](#), [PEEKSHORT](#), [PEEKINT](#), [POKEBYTE](#), [POKESHORT](#), [POKEINT](#), [POKEFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMblock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```

PEEKINT

KUVAUS

Lukee muistipalasta neljätavuisen (32-bittisen) kokonaisluvun.

KÄYTTÖ

PEEKINT (muistipala, paikka)

- **muistipala** = Se muuttuja, mihin muistipala luotiin.
- **paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.

Katso myös: [PEEKBYTE](#), [PEEKSHORT](#), [PEEKFLOAT](#), [POKEBYTE](#), [POKESHORT](#), [POKEINT](#), [POKEFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMblock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```

PEEKSHORT

KUVAUS

Lukee muistipalasta kaksitavuisen (16-bittisen) kokonaisluvun. Palautettu luku voi olla väliltä 0-65536.

KÄYTTÖ

```
PEEKSHORT (muistipala, paikka)
```

- **muistipala** = Se muuttuja, mihin muistipala luotiin.
- **paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.

Katso myös: [PEEKBYTE](#), [PEEKINT](#), [PEEKFLOAT](#), [POKEBYTE](#), [POKESHORT](#), [POKEINT](#), [POKEFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMblock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```


PICKCOLOR

KUVAUS

Poimii ruudulta pisteen ja asettaa nykyisen piirtovärin vastaamaan poimittua väriä. Voit esimerkiksi poimia väripaletista piirtovärin. Jos haluat poimia värin kuvasta ruudun sijaan, käytä mieluummin PickImageColor-komentoa.

KÄYTTÖ

PICKCOLOR x, y

- **x** = Kokonaisluku tai desimaaliluku. Poimintapisteen sijainti vaakasuunnassa.
- **y** = Kokonaisluku tai desimaaliluku. Poimintapisteen sijainti pystysuunnassa.

Katso myös: [PICKIMAGECOLOR](#), [PICKIMAGECOLOR2](#), [COLOR](#), [GETRGB](#)

ESIMERKKI

```
'THIS PROGRAM WILL CHALLENGE MS PAINT

FrameLimit 40

'load palette
palette= LoadImage("Media\palette.bmp")

'Make canvas
canvas= MakeImage(300,300)
ClsColor cbwhite
DrawToImage canvas
    Cls
DrawToScreen

ClsColor cbwhite
Color cbblack

'add info
Locate 110,10
AddText "Pick a color via palette right click"
AddText "Draw using the mouse"

ClsColor cbwhite

Repeat

    'draw palette
    DrawImage palette,0,0,0,OFF

    'draw preview
    Box 20,220,20,20

    'pick colour
    If MouseDown(2)
        If MouseX()<100 And MouseY()<200 Then
            PickColor MouseX(),MouseY()
        EndIf
    EndIf

    'draw
    If MouseDown(1)
        If MouseX()>100 Then
            DrawToImage canvas
                Line oldx-100,oldy,MouseX() -100,MouseY()
                oldx=MouseX()
                oldy=MouseY()
            DrawToScreen
        EndIf
    EndIf

    'draw canvas
    DrawImage canvas,100,0,0,OFF

    DrawScreen

Until EscapeKey()
```

PICKEDANGLE

KUVAUS

Poiminnasta nauhoitetaan aina sen koordinaattien lisäksi myös sen pinnan kulma, johon törmäys tapahtui. Kulma esitetään aina niin, että se on törmäyspinnasta kohtisuoraan pois päin.

Funktio palauttaa poimitun seinän kohtisuoran kulman kokonaislukuna.

KÄYTTÖ

PICKEDANGLE ()

Katso myös: [PICKEDX](#), [PICKEDY](#), [PICKEDOBJECT](#)

ESIMERKKI

```
FrameLimit 40

'Enable graphics commands to cast shapes to
'the game world
DrawToWorld ON

pacman=LoadObject("Media\pacman.bmp")

ObjectPickable pacman,2 'circle-pick

'Load the shooter
guy=LoadObject("Media\guy.bmp",180)
PositionObject guy,0, -100

Color cbblack
AddText "Guide via arrows"
AddText "SPACE to shoot"

ClsColor cbwhite

Repeat

    'Update controls
    If LeftKey() Then TurnObject guy,2
    If RightKey() Then TurnObject guy, -2
    If UpKey() Then MoveObject guy,2
    If DownKey() Then MoveObject guy, -2

    'Pick
    ObjectPick guy

    Color cbblack
    DrawGame

    'Something WAS picked, let's see what it is
    e=PickedObject()
    If e=pacman
        Color cbred
        Text 0,40,"PICK"
        Text 0,50,PickedAngle()
        Circle PickedX() -4,PickedY()+4,8,OFF

        If KeyDown(cbkeyspace) Then
            Line PickedX(),PickedY(),PickedX()+Cos(PickedAngle()+10)*20,PickedY()+Sin(PickedAngle()+10)*20
            Line PickedX(),PickedY(),PickedX()+Cos(PickedAngle()-10)*20,PickedY()+Sin(PickedAngle()-
10)*20
        EndIf
    EndIf

    'draw the check-zones
    Circle ObjectX(pacman) -32,ObjectY(pacman)+32,64,OFF

    DrawScreen

Until EscapeKey()
```

PICKEDOBJECT

KUVAUS

Tämä funktio kertoo mikä objekti poimittiin viimeeksi suoritettulla **CameraPick**- tai **ObjectPick**-komennolla. Jos poiminta ei osunut yhteenkään objektiin, palautetaan nolla.

KÄYTTÖ

```
PICKEDOBJECT ()
```

Katso myös: [PICKEDX](#), [PICKEDY](#), [OBJECTPICK](#), [PIXELPICK](#), [CAMERAPICK](#)

ESIMERKKI

```
FrameLimit 40

'Enable graphics commands to cast shapes to
'the game world
DrawToWorld ON

pacman1=LoadObject("Media\pacman.bmp",360)
pacman2=CloneObject(pacman1)
pacman3=CloneObject(pacman1)

ObjectPickable pacman1,1 'box-pick
ObjectPickable pacman2,2 'circle-pick
ObjectPickable pacman3,3 'pixel-perfect

PositionObject pacman1, -100,0
PositionObject pacman3, 100,0

'Load the shooter
guy=LoadObject("Media\guy.bmp",180)
PositionObject guy,0, -100

Color cbblack
AddText "Guide via arrows"
Locate 70,80
AddText "box-pick"
Locate 160,80
AddText "circle-pick"
Locate 260,80
AddText "pixel-perfect"
Color cbblack
ClsColor cbwhite

Repeat

  'Update controls
  If LeftKey() Then TurnObject guy,2
  If RightKey() Then TurnObject guy, -2
  If UpKey() Then MoveObject guy,2
  If DownKey() Then MoveObject guy, -2

  'Make them spin, add some excitement
  TurnObject pacman1,1
  TurnObject pacman2,1
  TurnObject pacman3,1

  'Pick
  ObjectPick guy

  DrawGame

  'Something WAS picked, let's see what it is
  e=PickedObject()
  Color cbblack
  If e Then
    Color cbred
    If e=pacman1 Then
      Text 0,30,"pacman on the left"
    ElseIf e=pacman2 Then
      Text 0,30,"pacman at the center"
    ElseIf e=pacman3 Then
```

```
        Text 0,30,"pacman on the right"  
    EndIf  
    Circle PickedX() -5,PickedY()+5,10,ON  
EndIf  
  
'draw the check-zones  
Box ObjectX(pacman1) -32,ObjectY(pacman1)+32,64,64,OFF  
Circle ObjectX(pacman2) -32,ObjectY(pacman2)+32,64,OFF  
  
DrawScreen  
  
Until EscapeKey()
```

PICKEDX

KUVAUS

Tämä funktio kertoo missä kohtaa tuorein poiminta törmäsi objektiin.
Funktio palauttaa poimitun X-koordinaatin kokonaislukuna.

KÄYTTÖ

PICKEDX ()

Katso myös: [PICKEDY](#), [PICKEDANGLE](#), [PICKEDOBJECT](#)

ESIMERKKI

```
FrameLimit 40

'Enable graphics commands to cast shapes to
'the game world
DrawToWorld ON

pacman1=LoadObject("Media\pacman.bmp",360)
pacman2=CloneObject(pacman1)
pacman3=CloneObject(pacman1)

ObjectPickable pacman1,1 'box-pick
ObjectPickable pacman2,2 'circle-pick
ObjectPickable pacman3,3 'pixel-perfect

PositionObject pacman1, -100,0
PositionObject pacman3, 100,0

'Load the shooter
guy=LoadObject("Media\guy.bmp",180)
PositionObject guy,0, -100

Color cbblack
AddText "Guide via arrows"
Locate 70,80
AddText "box-pick"
Locate 160,80
AddText "circle-pick"
Locate 260,80
AddText "pixel-perfect"
Color cbblack
ClsColor cbwhite

Repeat

  'Update controls
  If LeftKey() Then TurnObject guy,2
  If RightKey() Then TurnObject guy, -2
  If UpKey() Then MoveObject guy,2
  If DownKey() Then MoveObject guy, -2

  'Make them spin, add some excitement
  TurnObject pacman1,1
  TurnObject pacman2,1
  TurnObject pacman3,1

  'Pick
  ObjectPick guy

  DrawGame

  'Something WAS picked, let's see what it is
  e=PickedObject()
  Color cbblack
  If e Then
    Color cbred
    If e=pacman1 Then
      Text 0,30,"pacman on the left"
    ElseIf e=pacman2 Then
      Text 0,30,"pacman at the center"
    ElseIf e=pacman3 Then
```

```
        Text 0,30,"pacman on the right"  
    EndIf  
    Circle PickedX() -5,PickedY()+5,10,ON  
EndIf  
  
'draw the check-zones  
Box ObjectX(pacman1) -32,ObjectY(pacman1)+32,64,64,OFF  
Circle ObjectX(pacman2) -32,ObjectY(pacman2)+32,64,OFF  
  
DrawScreen  
  
Until EscapeKey()
```

PICKEDY

KUVAUS

Tämä funktio kertoo missä kohtaa tuorein poiminta törmäsi objektiin.
Funktio palauttaa poimitun Y-koordinaatin kokonaislukuna.

KÄYTTÖ

PICKEDY ()

Katso myös: [PICKEDX](#), [PICKEDANGLE](#), [PICKEDOBJECT](#)

ESIMERKKI

```
FrameLimit 40

'Enable graphics commands to cast shapes to
'the game world
DrawToWorld ON

pacman1=LoadObject("Media\pacman.bmp",360)
pacman2=CloneObject(pacman1)
pacman3=CloneObject(pacman1)

ObjectPickable pacman1,1 'box-pick
ObjectPickable pacman2,2 'circle-pick
ObjectPickable pacman3,3 'pixel-perfect

PositionObject pacman1, -100,0
PositionObject pacman3, 100,0

'Load the shooter
guy=LoadObject("Media\guy.bmp",180)
PositionObject guy,0, -100

Color cbblack
AddText "Guide via arrows"
Locate 70,80
AddText "box-pick"
Locate 160,80
AddText "circle-pick"
Locate 260,80
AddText "pixel-perfect"
Color cbblack
ClsColor cbwhite

Repeat

    'Update controls
    If LeftKey() Then TurnObject guy,2
    If RightKey() Then TurnObject guy, -2
    If UpKey() Then MoveObject guy,2
    If DownKey() Then MoveObject guy, -2

    'Make them spin, add some excitement
    TurnObject pacman1,1
    TurnObject pacman2,1
    TurnObject pacman3,1

    'Pick
    ObjectPick guy

    DrawGame

    'Something WAS picked, let's see what it is
    e=PickedObject()
    Color cbblack
    If e Then
        Color cbred
        If e=pacman1 Then
            Text 0,30,"pacman on the left"
        ElseIf e=pacman2 Then
            Text 0,30,"pacman at the center"
        ElseIf e=pacman3 Then
```

```
        Text 0,30,"pacman on the right"  
    EndIf  
    Circle PickedX() -5,PickedY()+5,10,ON  
EndIf  
  
'draw the check-zones  
Box ObjectX(pacman1) -32,ObjectY(pacman1)+32,64,64,OFF  
Circle ObjectX(pacman2) -32,ObjectY(pacman2)+32,64,OFF  
  
DrawScreen  
  
Until EscapeKey()
```


PICKIMAGECOLOR

KUVAUS

Poimii kuvasta pisteen ja asettaa nykyisen piirtovärin vastaamaan poimittua väriä. Voit esimerkiksi poimia väripaletista piirtovärin.

Jos haluat poimia värin suoraan ruudulta, käytä mieluummin PickColor-komentoa.

KÄYTTÖ

PICKIMAGECOLOR kuvamuuttuja, x, y

- **kuvamuuttuja** = Muuttuja, johon kuva ladattiin tai luotiin.
- **x** = Kokonaisluku tai desimaaliluku. Poimintapisteen sijainti vaakasuunnassa.
- **y** = Kokonaisluku tai desimaaliluku. Poimintapisteen sijainti pystysuunnassa.

Katso myös: [PICKIMAGECOLOR2](#), [PICKCOLOR](#)

ESIMERKKI

```
img=LoadImage("Media\palette.bmp")

ClsColor 255,255,255
Repeat

    DrawImage img,0,0,0,OFF

    If mouseX()<ImageWidth(img) And mouseY()<ImageHeight(img)
        PickImageColor img,mouseX(),mouseY()

        Box 100,100,50,30
    EndIf

    DrawScreen

Until KeyHit(1)
```

PICKIMAGECOLOR2

KUVAUS

Poimii kuvasta pisteen ja asettaa nykyisen piirtovärin vastaamaan poimittua väriä. Voit esimerkiksi poimia väripaletista piirtovärin.

Puskuri pitää lukita ennen komennon käyttöä. Tämä on hieman nopeampi kuin *PickImageColor*.

Jos haluat poimia värin suoraan ruudulta, käytä mieluummin PickColor-komentoa.

KÄYTTÖ

PICKIMAGECOLOR2 kuvamuuttuja, x, y

- **kuvamuuttuja** = Muuttuja, johon kuva ladattiin tai luotiin.
- **x** = Kokonaisluku tai desimaaliluku. Poimintapisteen sijainti vaakasuunnassa.
- **y** = Kokonaisluku tai desimaaliluku. Poimintapisteen sijainti pystysuunnassa.

Katso myös: [PICKIMAGECOLOR](#), [PICKCOLOR](#)

ESIMERKKI

```
img=LoadImage("Media\palette.bmp")

ClsColor 255,255,255
Repeat

    DrawImage img,0,0

    If MouseX()<ImageWidth(img) And MouseY()<ImageHeight(img)

        Lock Image(img)
        PickImageColor2 img,MouseX(),MouseY()
        Unlock Image(img)

        Box 100,100,50,30
    EndIf

    DrawScreen

Until KeyHit(1)
```

PIXELPICK

KUVAUS

Objekti suorittaa poiminnan. Poimintaviiva lähtee suoraan objektin osoittamaan suuntaan, ja CoolBasic tarkastaa mihin se osuu. Ero **ObjectPick**iin on se, että poiminta suoritetaan objektin sisältä. Komento on suunniteltu kuvaan perustuvaan kenttään. Ainahan ei ole pakko käyttää tilekarttaa :)

Komennon ihanteellinen käyttötarkoitus on ns. *instant hit bullets*, eli luodin osuma. Luoti liikkuu äärettömän nopeasti, ja osuu ensimmäiseen seinään. Kaikki "poimittavat" objektit pitää esitellä komennolla **ObjectPickable** arvolle **3**.

Huomaa, että piilotettuja objekteja ei voi poimia. Voit käyttää komentoa vaikkapa luolalentelyihin, jossa kenttä ladataan objektina (ilman pyöritysarvoa), ja jonka sisällä alukset lentelevät.

Valinnaisella täydellisyysarvolla määritetään poiminnan tarkkuus. Suuremmat arvot toimivat nopeammin suurilla etäisyyksillä. Oletuksena 1, joka on tarkin mahdollinen poiminta.

KÄYTTÖ

PIXELPICK objektimuuttuja, [täydellisyys]

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin
- **täydellisyys** (**valinnainen**) = Kokonaisluku 1 ->. Poiminnan tarkkuus/nopeus. Oletuksena 1 (tarkin)

Katso myös: [OBJECTPICKABLE](#), [OBJECTPICK](#), [CAMERAPICK](#), [PICKEDOBJECT](#), [PICKEDX](#), [PICKEDY](#)

ESIMERKKI

```
Const gravity= -0.02

'declare variables as decimal
Dim target_movex As Float, target_movey As Float
Dim movex As Float, movey As Float

'change the display mode (low resolution, zoomed display)
SCREEN 320,240,16,cbsizable
SetWindow "",3
DrawToWorld ON

'load the image-map (NO rotation detail)
map=LoadObject("Media\level.bmp")
ObjectPickable map,3

'load the ship
ship=LoadObject("Media\ship.bmp",72)

Repeat

    oldx#=ObjectX(ship) 'memorize the old coordinates
    oldy#=ObjectY(ship)

    'turn the ship
    If LeftKey() Then TurnObject ship,5
    If RightKey() Then TurnObject ship, -5

    'thrust/update physics
    If UpKey() Then
        target_movex=Cos(ObjectAngle(ship))*3
        target_movey=Sin(ObjectAngle(ship))*3

        If target_movex< movex Then movex=movex-0.05
        If target_movex> movex Then movex=movex+0.05
        If target_movey< movey Then movey=movey-0.05
        If target_movey> movey Then movey=movey+0.05
    EndIf

    'move the ship according to physics
    movey=movey+gravity
    If movey< -4 Then movey= -4
    TranslateObject ship,movex,movey

    'make the camera follow the ship
    CloneCameraPosition ship

    'COLLISION FOR THE LEVEL
    CameraPick 160,120
    If PickedObject()=map Then
```

```
        PositionObject ship,oldx,oldy
        movey=0
    EndIf

    DrawGame 'for drawing

    'figure out where the pick intersects with the level
    PixelPick ship
    If PickedObject() Then
        Circle PickedX() -5,PickedY()+5,10
    EndIf

    DrawScreen

Forever
```

PLAYANIMATION

KUVAUS

Tämä komento alkaa soittamaan pyydettyä videotiedostoa. Tuettuna ovat kaikki DirectShown tukemat tiedostomuodot, koodekit ja filtit. Tästä syystä oikeat ohjelmistot tulee olla asennettuna.

Ruudun täytyy olla tarpeeksi iso, tai videota ei voi näyttää. Tarkasta täsmällinen koko funktioilla AnimationWidth ja AnimationHeight. Muuten videota voi venyttää ruudussa, joskin se toimii nopeimmin alkuperäisessä koossaan.

Huomaa, että kun video avataan, se lähtee välittömästi pyörimään. Video soi taustalla ilman keskeytyksiä alkuperäisen aikajanan mukaisesti, joten videota ei voi keskeyttää jättämällä kutsumatta DrawAnimation-komentoa.

Jos haluat poistua videotiostosta ennen kuin se loppuu, sinun täytyy kutsua **StopAnimation**-komentoa.

Funktio palauttaa nollan, jos videotiedostoa ei voitu avata. Muutoin palautetaan ykkönen.

KÄYTTÖ

```
PLAYANIMATION (tiedostonimi)
```

- **tiedostonimi** = Merkkijono, polku videotiedostoon.

Katso myös: [DRAWANIMATION](#), [STOPANIMATION](#)

ESIMERKKI

```
SCREEN 640,480

'load the video
video= PlayAnimation("myMovie.avi")

'draw video
Repeat
    DrawAnimation video
    DrawScreen
Until EscapeKey() Or KeyHit(28)

'close the video input
StopAnimation video
```

PLAYOBJECT

KUVAUS

Tämä komento animoi objektin. Objekti pitää olla ladattuna funktiolla **LoadAnimObject**. Aloitus- ja lopetuskohdat määrittävät, mikä osa animoidusta kuvasarjasta soitetaan. Kaikki loput parametrit ovat valinnaisia. Huomaa, että ensimmäinen animaatioframe on nolla, joten jos haluat soittaa animaation alusta alkaen, ensimmäiseksi parametriksi on annettava nolla.

Voit käyttää miinusmerkkistä nopeutta, jos haluat toistaa animaation takaperin. Silloin alkuarvon pitää olla suurempi kuin lopetusarvon. Esim. **10->0**, nopeus **-5**. Kaikkia frameja käsitellään desimaalilukuina, joten saat halutessasi melko tarkan animaatiokeston. Nykyisen framen saat funktiolla **ObjectFrame**.

Valinnainen jatkumoparametri määrittää, jatketaanko animaation soittamista samasta kohdasta, mihin se pysäytettiin komennolla **StopObject**. Tällä tavalla voit *tauottaa* animaatioita. Yleensä aloittelijat eivät tarvitse jatkumoa, joten sen voi jättää kirjoittamatta.

Animaatio soitetaan kerran, jonka jälkeen se pysähtyy ja palaa samaan frameen, joka esiteltiin animaation alkupaikaksi.

Tilekarttoja animoidaan eri tavalla. Alku- ja loppuarvoilla ei silloin ole merkistystä, pistä vaikka nollat kumpaankin. Sen sijaan animaationopeus on erilainen kuin normaaleissa objekteissa. Siinä se ilmoitetaan monenko sekunnin välein animaatio päivitetään. Esimerkiksi arvo 0.5 on päivittää kartan animaation 2 kertaa sekuntissa, ja arvo 4 joka 4 sekunnin välein.

KÄYTTÖ

```
PLAYOBJECT objektimuuttuja, [alku], [loppu], [nopeus], [jatkumo]
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
 - **alku** (**valinnainen**) = Mistä kohtaa animaatiota aletaan soittamaan. Alku on nolla.
 - **loppu** (**valinnainen**) = Animaation loppukohta.
 - **nopeus** (**valinnainen**) = Animaation nopeus (suurempi on nopeampi). Oletus 0.1.
 - **jatkumo** (**valinnainen**)
- OFF** = aloita aina alusta (oletus)
ON = jatka siitä mihin animaatio on pysäytetty

Katso myös: [LOOPOBJECT](#), [STOPOBJECT](#), [LOADANIMOBJECT](#)

ESIMERKKI

```
'make screen a bit zoomed
SCREEN 250,188,0,cbsizable

FrameLimit 40 'limit game speed

'load animation for both directions, but hide them
zeroright=LoadAnimObject("Media\zerorun1.bmp",47,51,0,12)
MaskObject zeroright,cbmagenta
ShowObject zeroright,OFF
zeroleft=LoadAnimObject("Media\zerorun2.bmp",47,51,0,12)
MaskObject zeroleft,cbmagenta
ShowObject zeroleft,OFF

'default direction
zero=CloneObject(zeroright)

Color cbblack
AddText "Try arrows"
ClsColor cbwhite

'MAIN LOOP
Repeat

    'run right!
    If RightKey() Then
        PaintObject zero,zeroright
        PlayObject zero,2,11,0.25 'animate at speed 0.25
    'run left!
    ElseIf LeftKey() Then
        PaintObject zero,zeroleft
        PlayObject zero,2,11,0.25 'animate at speed 0.25
    Else
        'nothing is pressed so freeze
        PlayObject zero,0
    EndIf

    Text 0,140,"Playing? "+ObjectPlaying(zero)
    Text 0,160,"Frame: "+ObjectFrame(zero)

DrawScreen
```

Until **EscapeKey**()

PLAYSOUND

KUVAUS

Soittaa kaikenlaisia ääniä. Tämän komennon erikoisuus on siinä, että sitä voidaan käyttää moneen tarkoitukseen. Sillä voi:

- Soittaa aiemmin ladatun äänen
- Soittaa musiikkia suoraan kovalevyltä
- Soittaa CD-raitoja
- Asettaa äänen voimakkuuden, sijainnin, taajuuden ja toiston

Ladatun äänen soittaminen

Ensimmäiseksi parametriksi kirjoitat äänimuuttujan nimen, eli sen muuttujan, mihin ääni ladattiin funktiolla *LoadSound*.

Musiikin soittaminen

Ensimmäiseksi parametriksi kirjoitat merkkijonon, joka on soitettavan musiikkitiedoston tiedostonimi. Musiikki soitetaan suoraan kovalevyltä, eikä sitä tarvitse erikseen ladata muistiin. Tunnetut tiedostoformaatit ovat: **raw, mod, s3m, xm, it, mid, rmi, wav, mp2, mp3, ogg, wma, asf ja mo3**.

CD-raidan soittaminen

Ensimmäiseksi parametriksi kirjoitat kokonaisluvun väliltä 0-99. Luku määrittää soitettavan raidan järjestysnumeron.

Voit lisäksi valinnaisilla parametreilla muuttaa äänen voimakkuutta, sijaintia ja taajuutta. Sijainti muuttaa äänen paneerausta vasemman ja oikean kaiuttimen välillä. Jos haluat muuttaa äänen jatkuvalla soitolle, käytä siihen **SetSound**-komentoa.

Tätä komentoa voi käyttää myös funktiona (ilmoitetaan sulkeiden kanssa), jolloin se palauttaa äänikanavan, jonne ääni ohjattiin. Äänikanavan avulla voit myöhemmin säätää äänen voimakkuutta, toistoa, sijaintia ja taajuutta lennossa. Lisätietoja **SetSound**-komennon ohjeesta.

KÄYTTÖ

```
PLAYSOUND soitto, [voimakkuus], [sijainti], [taajuus]
tai
PLAYSOUND (soitto, [voimakkuus], [sijainti], [taajuus])
```

- **soitto** = Äänimuuttuja, merkkijono tai kokonaisluku väliltä 0-99. Katso lisätietoja ylhäältä.
- **voimakkuus** (**valinnainen**) = Luku väliltä 0-100. Nolla on ei ääntä, 100 on täysillä. Oletuksena 100. Tätä parametria ei ole pakko kirjoittaa.
- **sijainti** (**valinnainen**) = Luku väliltä -100 ja 100. Nolla on keskitetty ääni, -100 on vasen kaiutin ja 100 oikea kaiutin. Oletuksena 0. Tätä parametria ei ole pakko kirjoittaa.
- **taajuus** (**valinnainen**) = Kokeile lukuja 8000-44000. Ilmoitetaan hertseinä. Jätä pois, jos et halua muuttaa taajuutta.

Katso myös: [SETSOUND](#), [LOADSOUND](#)

ESIMERKKI

```
PlaySound "Media\SK_Battle2.mp3" 'start playing background music

sfx=LoadSound("Media\Blaster.wav")

AddText "Hit SPACE to play a laser sound"
AddText "Try arrows"

Repeat

    If KeyHit(cbkeyspace) Then
        laserchannel=PlaySound(sfx,50) 'play sound at 50% volume
    EndIf

    'play in left speaker
    If KeyHit(cbkeyleft) Then PlaySound sfx, 100, -100

    'play in right speaker
    If KeyHit(cbkeyright) Then PlaySound sfx,100, 100

    'play high pitch
    If KeyHit(cbkeyup) Then PlaySound sfx,100,0,60000

    'play low pitch
    If KeyHit(cbkeydown) Then PlaySound sfx,100,0,8000

    'print some info
    Text 60,200,"Laser sound playing: "+SoundPlaying(laserchannel)

    DrawScreen

Until EscapeKey()
```


POINTCAMERA

KUVAUS

Tämä komento pistää kameran osoittamaan objektin suuntaan. Myöhemmin kameraa voi esimerkiksi liikuttaa eteenpäin - kohti objektia.

KÄYTTÖ

POINTCAMERA objekti

- **objekti** = Objektimuuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [POINTOBJECT](#)

ESIMERKKI

```
FrameLimit 40

'Make infinite grass
floor=MakeObjectFloor()
lawn=LoadImage("Media\grass.bmp")
PaintObject floor,lawn

target=LoadObject("Media\crosshair.bmp")

AddText "Move the crosshair with arrows..."

Repeat

    'move the crosshair at speed 2
    If LeftKey() Then TranslateObject target, -3,0
    If RightKey() Then TranslateObject target,3,0
    If UpKey() Then TranslateObject target,0,3
    If DownKey() Then TranslateObject target,0, -3

    'Point camera to the crosshair and move it
    'towards
    PointCamera target
    MoveCamera 1

    'Visualize the camera angle
    DrawGame
    Circle 190,140,20,OFF
    Line 200,150,200+Cos(CameraAngle())*10,150-Sin(CameraAngle())*10

    DrawScreen

Until EscapeKey()
```

Objektit : **komennot**

POINTOBJECT

KUVAUS

Kääntää objektin osoittamaan toista objektia. Esimerkiksi vihollinen tähtää pelaajaa.

KÄYTTÖ

```
POINTOBJECT objektimuuttuja, kohde_objekti
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **kohde_objekti** = Se objekti, jota kohti käännetään.

Katso myös: [GETANGLE2](#)

ESIMERKKI

```
FrameLimit 40 'limit game speed

aim=LoadObject("Media\crosshair.bmp")
soldier=LoadObject("Media\soldier.bmp",360)

Color cborange
AddText "Guide the soldier with the arrows..."

ClsColor cbdark

Repeat

    'Move object
    TranslateObject soldier,RightKey()*2,0
    TranslateObject soldier, -LeftKey()*2,0
    TranslateObject soldier,0,UpKey()*2
    TranslateObject soldier,0, -DownKey()*2

    'Keep the aim at the cross
    PointObject soldier,aim

    DrawScreen

Until EscapeKey()
```

POKEBYTE

KUVAUS

Kirjoittaa muistipalaan yhden tavun. Tavu on positiivinen kokonaisluku väliltä 0-255.

Pidä huoli, että et kirjoita muistialueen ulkopuolelle, koska se saattaa kaataa ohjelman tai jopa koko järjestelmän. Ensimmäinen kirjoituspaikka on nolla ja viimeinen KOKO-1.

KÄYTTÖ

POKEBYTE muistipala, paikka, arvo

- **muistipala** = Se muuttuja, mihin muistipala luotiin.
- **paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.
- **arvo** = Mielellään kokonaisluku.

Katso myös: [POKESHORT](#), [POKEINT](#), [POKEFLOAT](#), [PEEKBYTE](#), [PEEKSHORT](#), [PEEKINT](#), [PEEKFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMblock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```

POKEFLOAT

KUVAUS

Kirjoittaa muistipalaan neljän tavun (32-bittisen) desimaaliluvun. Arvon tulisi mahtua joukkoon 3.4E +/- 38 (7 numeroa).

Pidä huoli, että et kirjoita muistialueen ulkopuolelle, koska se saattaa kaataa ohjelman tai jopa koko järjestelmän. Ensimmäinen kirjoituspaikka on nolla ja viimeinen KOKO-1.

KÄYTTÖ

POKEFLOAT muistipala, paikka, arvo

- **muistipala** = Se muuttuja, mihin muistipala luotiin.
- **paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.
- **arvo** = Mielellään desimaaliluku.

Katso myös: [POKEBYTE](#), [POKESHORT](#), [POKEINT](#), [PEEKBYTE](#), [PEEKSHORT](#), [PEEKINT](#), [PEEKFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMblock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```

POKEINT

KUVAUS

Kirjoittaa muistipalaan neljän tavun (32-bittisen) kokonaisluvun. Arvon tulisi olla väliltä -2 147 483 647 ja 2 147 483 647.

Pidä huoli, että et kirjoita muistialueen ulkopuolelle, koska se saattaa kaataa ohjelman tai jopa koko järjestelmän. Ensimmäinen kirjoituspaikka on nolla ja viimeinen KOKO-1.

KÄYTTÖ

```
POKEINT muistipala, paikka, arvo
```

- **muistipala** = Se muuttuja, mihin muistipala luotiin.
- **paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.
- **arvo** = Mielellään kokonaisluku.

Katso myös: [POKEBYTE](#), [POKESHORT](#), [POKEFLOAT](#), [PEEKBYTE](#), [PEEKSHORT](#), [PEEKINT](#), [PEEKFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMBlock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```

POKESHORT

KUVAUS

Kirjoittaa muistipalaan kahden tavun (16-bittisen) kokonaisluvun. Arvon tulisi olla väliltä 0-65536.

Pidä huoli, että et kirjoita muistialueen ulkopuolelle, koska se saattaa kaataa ohjelman tai jopa koko järjestelmän. Ensimmäinen kirjoituspaikka on nolla ja viimeinen KOKO-1.

KÄYTTÖ

POKESHORT muistipala, paikka, arvo

- **muistipala** = Se muuttuja, mihin muistipala luotiin.
- **paikka** = Sijainti mistä luetaan. Nolla on ensimmäinen.
- **arvo** = Mielellään kokonaisluku.

Katso myös: [POKEBYTE](#), [POKEINT](#), [POKEFLOAT](#), [PEEKBYTE](#), [PEEKSHORT](#), [PEEKINT](#), [PEEKFLOAT](#)

ESIMERKKI

```
mymem=MakeMEMblock(11)

'Write some information to the memblock
PokeByte mymem, 0, 7
PokeShort mymem, 1, -3400
PokeInt mymem, 3, 28561200
PokeFloat mymem, 7, 3.141592

'Read the information and print it
ourbyte=PeekByte(mymem,0)
ourshort=PeekShort(mymem,1)
ourint=PeekInt(mymem,3)
ourfloat#=PeekFloat(mymem,7)

Text 10,10,ourbyte
Text 10,30,ourshort
Text 10,50,ourint
Text 10,70,ourfloat

DrawScreen

WaitKey
```

POSITIONCAMERA

KUVAUS

Asettaa kameran tiettyihin pelimaailman koordinaatteihin. (0,0) on pelimaailman keskipiste.

KÄYTTÖ

POSITIONCAMERA x, y

- **x** = Kokonaisluku tai desimaaliluku. Kameran sijainti vaakasuunnassa. Pelimaailman koordinaatti.
- **y** = Kokonaisluku tai desimaaliluku. Kameran sijainti pystysuunnassa. Pelimaailman koordinaatti.

ESIMERKKI

```
FrameLimit 40

'Make scene
floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

guy=LoadObject("Media\guy.bmp")

AddText "Press SPACE for default camera position"

Repeat

    'move camera as absolute
    TranslateCamera 1,1

    'return to origin?
    If KeyDown(cbkeyspace) Then PositionCamera 0,0

    'Print the camera coordinates
    DrawGame
    Text 20,20,CameraX()
    Text 20,40,CameraY()

    DrawScreen

Until EscapeKey()
```

POSITIONMOUSE

KUVAUS

Tämä komento sijoittaa hiirikursorin haluttuun paikkaan ruudulla.

Komennon paras käyttötarkoitus ilmenee esimerkistä.

KÄYTTÖ

```
POSITIONMOUSE x, y
```

- **x** = Vaakasuora sijainti pikseleinä.
- **y** = Pystysuora sijainti pikseleinä.

Katso myös: [MOUSEX](#), [MOUSEY](#), [MOUSEMOVEX](#), [MOUSEMOVEY](#)

ESIMERKKI

```
'NOTE! You can hide the mouse pointer with  
'ShowMouse OFF
```

```
cursor= LoadImage ("Media\cow.bmp")
```

```
x=200
```

```
y=150
```

```
Repeat
```

```
    x=x+MouseMoveX()
```

```
    y=y+MouseMoveY()
```

```
    PositionMouse 200,150
```

```
    DrawImage cursor,x,y
```

```
    DrawScreen
```

```
Until EscapeKey()
```


POSITIONOBJECT

KUVAUS

Asettaa objektin haluttuun sijaintiin. Voit esimerkiksi siirtää pelihahmon sen alkupaikkaan.

Huomaa, että koordinaatisto pelimaailmassa on erilainen kuin ruudulla:

□ □

Keskus on nolla, ja koordinaatit voivat olla myös negatiivisia. Viime kädessä kamera kuitenkin päättää mihin kohtaan objektit piirretään.

Z-arvon muuttaminen on edistyneemmille käyttäjille. Sen avulla voidaan teeskennellä syvyysvaikutelmaa, mikä saa objektin piirtymään sijaintinsa puolesta perspektiivissä taustaa vasten. Positiiviset Z-arvot tuovat objektia "lähemmäs kameraa", ja negatiiviset loitontavat sitä.

KÄYTTÖ

```
POSITIONOBJECT objektimuuttuja, x, y, [Z]
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **x** = Lukuarvo, objektin sijainti X-suunnassa.
- **y** = Lukuarvo, objektin sijainti Y-suunnassa.
- **Z** (valinnainen, **edistyneemmille**) = Lukuarvo, sijainti syvyys suunnassa. Oletuksena nolla.

Katso myös: [SCREENPOSITIONOBJECT](#), [CLONEOBJECTPOSITION](#), [CLONECAMERAPOSITION](#)

ESIMERKKI

```
FrameLimit 40

'Load three cows
cow1=LoadObject("Media\cow.bmp")
cow2=LoadObject("Media\cow.bmp")
cow3=LoadObject("Media\cow.bmp")

'Position two of them to different place
PositionObject cow2, -100,0
PositionObject cow3,0,100

Color cbblack
AddText "Arrows move camera"

ClsColor cbwhite

Repeat

    'Control camera in one line!
    TranslateCamera RightKey() -LeftKey(),UpKey() -DownKey()

    DrawScreen

Until EscapeKey()
```

PRINT

KUVAUS

Kirjoittaa tekstiä ruudulle. Fontti pysyy aina samana. Komento kirjoittaa tekstin suoraan näytön etupuskuriin, joten se ilmestyy ruudulle välittömästi ilman DrawScreeniä. Tulostuksen jälkeen "rivi vaihtuu", joten seuraava Print tulostaa rivin verran alaspäin.

Voit asettaa tekstin sijainnin käyttämällä **Locate**-komentoa.

Emme suosittele tämän komennon käyttämistä, koska se saattaa aiheuttaa ongelmia muun grafiikan kanssa. Käytä paljon mieluummin Textiä tai AddTextiä. Print-komento on jätetty CoolBasiciin totuttelusyistä.

Valitamme, että Print-komento siitä huolimatta esiintyy joissakin esimerkeissämme.

KÄYTTÖ

```
PRINT [teksti]
```

- **teksti** (*valinnainen*) = Tulostettava kokonaisluku, desimaaliluku tai merkkijono. Jätä pois, jos haluat tulostaa tyhjän rivin.

Katso myös: [LOCATE](#), [WRITE](#)

ESIMERKKI

```
For i=1 To 5

    Print i
    Print 'an empty line

Next i

WaitKey
```

PUTPIXEL

KUVAUS

Kirjoittaa (32-bittisen) pikselin nykyiseen ei-lukittuun piirtopuskuriin.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

PUTPIXEL *x*, *y*, pikseli, [*puskuri*]

- **x** = Kokonaisluku, pikselin X-sijainti.
- **y** = Kokonaisluku, pikselin Y-sijainti.
- **pikseli** = Kokonaisluku. Pikselin lukuarvo. Pikseli, joka on luettu funktiolla GetPixel
- **puskuri** (*valinnainen*) = Puskuriosoitin. Kts. funktiot SCREEN() ja IMAGE(). Oletuksena nykyinen puskuuri

Katso myös: [GETPIXEL](#), [SCREEN](#), [IMAGE](#)

ESIMERKKI

```
SCREEN 640,480
```

```
AddText "Press any key to copy..."
```

```
'First make a gradient
```

```
For i=0 To 240
```

```
    Color 0,i+15,0
```

```
    Line 0,i,640,i
```

```
Next i
```

```
DrawScreen OFF
```

```
WaitKey
```

```
'make 154 481 iterations, copy each pixel
```

```
For y=0 To 240
```

```
For x=0 To 640
```

```
    PutPixel x,480-y,GetPixel(x, y)
```

```
Next x
```

```
Next y
```

```
DrawScreen
```

```
WaitKey
```

PUTPIXEL2

KUVAUS

Kirjoittaa (32-bittisen) pikselin nykyiseen piirtopuskuriin. Komento on nopeampi kuin **PutPixel**.

Puskurin täytyy olla lukittu. Tämä tapahtuu komennolla **Lock**. Pikselikomennot lukittuun puskuriin ovat erittäin nopeita.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

PUTPIXEL2 x, y, pikseli, [puskuri]

- **x** = Kokonaisluku, pikselin X-sijainti.
- **y** = Kokonaisluku, pikselin Y-sijainti.
- **pikseli** = Kokonaisluku. Pikselin lukuarvo. Pikseli, joka on luettu funktiolla GetPixel2.
- **puskuri** (**valinnainen**) = Puskuriosoitin. Kts. funktiot SCREEN() ja IMAGE(). Oletuksena nykyinen puskuri.

Katso myös: [LOCK](#), [GETPIXEL2](#), [SCREEN](#), [IMAGE](#)

ESIMERKKI

```
SCREEN 640,480

AddText "Press any key to copy..."

'First make a gradient
For i=0 To 240
    Color 0,i+15,0
    Line 0,i,640,i
Next i

DrawScreen OFF

WaitKey

'Lock screen
Lock

'make 154 481 iterations, copy each pixel
For y=0 To 240
For x=0 To 640
    PutPixel2 x,480-y,GetPixel2(x, y)
Next x
Next y

'Unlock screen
Unlock

DrawScreen

WaitKey
```

RAND

KUVAUS

Generoi satunnaisluvun annetulta väliltä. Luku on kokonaisluku, toisin kuin Rnd.

Käytetään lisäämään satunnaisuutta ja vivahteikuutta.

KÄYTTÖ

RAND (mistä, mihin)

- **mistä** (valinnainen) = Desimaaliluku. Pienin alaraja. Kokonaisluku.
- **mihin** = Kokonaisluku. Suurin yläraja. Kokonaisluku.

Katso myös: [RND](#), [RANDOMIZE](#)

ESIMERKKI

```
Text 10,10,Rand(-10,10)
Text 10,20,Rand(6) '0-6
Text 10,30,Rand(44,55)
```

```
DrawScreen
WaitKey
```

RANDOMIZE

KUVAUS

Alustaa satunnaislukugeneraattorin. Ohjelmointikielissä luvut eivät oikeasti ole satunnaisia. Todellisuudessa ne poimitaan eräänlaisesta listasta. Generaattorin alustamisella tarkoitetaan listaosoittimen siirtämistä eri paikkaan. Niin, että seuraavalla kerralla generoidaan erilaisia lukuja.

Jos haluat luoda esim. satunnaista metsää, voit alustaa satunnaisluvut tietylle arvolle. Näin pelikenttä pysyy muuttumattomana joka kerta kun peli käynnistetään, vaikka "metsä" näyttääkin satunnaiselta.

Tätä komentoa ei ole pakko käyttää ohjelman alussa (toisin kuin muissa ohjelmointikielissä), koska CoolBasic on jo alustanut satunnaislukugeneraattorin täysin satunnaiseksi.

Aja tämän komennon esimerkkikoodi muutaman kerran läpi niin huomaat, että generoidut luvut ovat samoja joka kerralla. Muuta siemenlukua, niin saat erilaisia sarjoja.

VINKKI: Komentamalla `Randomize Timer()` saat luvuista "oikeasti" satunnaisia, eli sarja muuttuu joka kerta kun ohjelma käynnistetään.

KÄYTTÖ

```
RANDOMIZE siemenluku
```

- **siemenluku** = Jokin kokonaisluku.

Katso myös: [RND](#), [RAND](#)

ESIMERKKI

```
'Try to change this number
Randomize 5

Text 0,0,"This week's lottery numbers are:"

    Text 10,20,Rand(1,49)
    Text 10,30,Rand(1,49)
    Text 10,40,Rand(1,49)
    Text 10,50,Rand(1,49)
    Text 10,60,Rand(1,49)
    Text 10,70,Rand(1,49)
    Text 10,80,Rand(1,49)

Text 0,100,"Press any key to quit..."

DrawScreen
WaitKey
```

READ

KUVAUS

Lukee tietoa datalistasta.

Ei ole väliä, onko sinulla yksi iso datalista vai monta pienempää listaa. Kaikki yhteenkuuluvat listat pitää kuitenkin kirjoittaa peräkkäin ilman, että välissä on muuta koodia. Jos luet listan yli, saat siitä virheilmoituksen.

Voit lukea erilaisia tietotyyppejä. Palautetun arvon tyyppi riippuu puhtaasti datalistasta.

KÄYTTÖ

READ ()

Katso myös: [DATA](#), [RESTORE](#)

ESIMERKKI

```
Restore datalist

'Read all 3 values.
'Btw, try to loop 4
For i=1 To 3
    Print READ()
Next i

WaitKey
End

datalist:
DATA 56,78.9,"Hello"
```

READBYTE

KUVAUS

Lukee yhden tavun tiedostosta. Samalla tiedosto-osoitin siirtyy yhden askeleen eteenpäin.

Tavun arvo on kokonaisluku väliltä 0-255.

KÄYTTÖ

READBYTE (tiedostomuuttuja)

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.

Katso myös: [WRITEBYTE](#), [READSHORT](#), [READINT](#), [READFLOAT](#), [READSTRING](#), [READLINE](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
  lifes=8

  WriteByte f,8
  WriteShort f,lifes*9
  WriteInt f, -8901
  WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

  mybyte=ReadByte(f)
  myshort=ReadShort(f)
  myint=ReadInt(f)
  myfloat=ReadFloat(f)

  Text 10,10,mybyte
  Text 10,30,myshort
  Text 10,50,myint
  Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```


READFLOAT

KUVAUS

Lukee tiedostosta 32-bittisen desimaaliluvun. Samalla tiedosto-osoitin siirtyy neljä askelta eteenpäin.

Palautettu arvo on desimaaliluku.

KÄYTTÖ

READFLOAT (tiedostomuuttuja)

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.

Katso myös: [WRITEFLOAT](#), [READBYTE](#), [READSHORT](#), [READINT](#), [READSTRING](#), [READLINE](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
  lifes=8

  WriteByte f,8
  WriteShort f,lifes*9
  WriteInt f, -8901
  WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

  mybyte=ReadByte(f)
  myshort=ReadShort(f)
  myint=ReadInt(f)
  myfloat=ReadFloat(f)

  Text 10,10,mybyte
  Text 10,30,myshort
  Text 10,50,myint
  Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```

READINT

KUVAUS

Lukee tiedostosta 32-bittisen kokonaisluvun. Samalla tiedosto-osoitin siirtyy neljä askelta eteenpäin.

Tulos on kokonaisluku.

KÄYTTÖ

READINT tiedostomuuttuja

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.

Katso myös: [WRITEINT](#), [READBYTE](#), [READSHORT](#), [READFLOAT](#), [READSTRING](#), [READLINE](#)

ESIMERKKI

```
x=200
y=150

AddText "Use arrows to move"
AddText "F1=save game"
AddText "F2=load game"

Repeat

    If KeyHit(cbkeyf1) Then GoSub savegame
    If KeyHit(cbkeyf2) Then GoSub loadgame

    If LeftKey() Then x-1
    If RightKey() Then x+1
    If UpKey() Then y-1
    If DownKey() Then y+1

    Box x,y,10,10

    DrawScreen

Until EscapeKey()
End

'This sub-program will save game
savegame:

    f=OpenToWrite("Media\simpleSave.dat")

    WriteInt f,x
    WriteInt f,y

    CloseFile f

Return

'This sub-program will load game
loadgame:

    f=OpenToRead("Media\simpleSave.dat")

    x=ReadInt(f)
    y=ReadInt(f)

    CloseFile f

Return
```

READLINE

KUVAUS

Lukee tekstitiedostosta rivin. Palauttaa siis merkkijonon.

KÄYTTÖ

READLINE (tiedostomuuttuja)

- **tiedostomuuttuja** = Se muuttuja, mihin tiedosto avattiin.

ESIMERKKI

```
'make window bigger as otherwise the text won't fit
SCREEN 580,420,16,cwindowed

'open the file
f=OpenToRead("Media\example.txt")

textline=0

'print the whole file to the screen
While Not EOF(f)

    Text 0,textline,ReadLine(f)

    textline=textline+12

Wend

'show the screen
DrawScreen

'don't forget this!
CloseFile f

WaitKey
```

READSHORT

KUVAUS

Lukee WORD-arvon (2 tavua/16-bittiä, kokonaisluku) tiedostosta. Samalla tiedosto-osoitin siirtyy kaksi askelta eteenpäin.

Palautettu arvo on kokonaisluku väliltä 0-65536.

KÄYTTÖ

READSHORT (tiedostomuuttuja)

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.

Katso myös: [WRITESHORT](#), [READBYTE](#), [READINT](#), [READFLOAT](#), [READSTRING](#), [READLINE](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
  lifes=8

  WriteByte f,8
  WriteShort f,lifes*9
  WriteInt f, -8901
  WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

  mybyte=ReadByte(f)
  myshort=ReadShort(f)
  myint=ReadInt(f)
  myfloat=ReadFloat(f)

  Text 10,10,mybyte
  Text 10,30,myshort
  Text 10,50,myint
  Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```

READSTRING

KUVAUS

Lukee tiedostosta merkkijonon. Palauttaa siis merkkijonon.

KÄYTTÖ

```
READSTRING (tiedostomuuttuja)
```

- **tiedostomuuttuja** = Se muuttuja, mihin tiedosto avattiin.

ESIMERKKI

```
'Please see other file reading functions'
```

REDIM

KUVAUS

Uudelleenvenyttää taulukkoa. Ulottuvuuksien määrää ei voi muuttaa, mutta joskus koon muuttaminen on esim. muistin kannalta hyödyllistä. Tai kun pitää varata taulukko uudelleen tietylle määrälle vihollisia (joiden määrä vaihtelee kentittäin).

Venytysoperaatio tuhoaa taulukon sisällön. Tämän voi estää komentamalla **ClearArray OFF**. Tämä ei kuitenkaan toistaiseksi toimi merkijonotaulukoille.

KÄYTTÖ

```
DIM muuttuja (ulottuvuus1_elementit [,ulottuvuus2_elementit] [,ulottuvuus3_elementit]
[,ulottuvuus4_elementit] [,ulottuvuus5_elementit])
```

- **ulottuvuus1_elementit** = Kokonaisluku, joka määrittää ensimmäisen ulottuvuuden solujen määrän.
- **ulottuvuus2_elementit** (valinnainen) = Kokonaisluku, joka määrittää toisen ulottuvuuden solujen määrän.
- **ulottuvuus3_elementit** (valinnainen) = Kokonaisluku, joka määrittää kolmannen ulottuvuuden solujen määrän.
- **ulottuvuus4_elementit** (valinnainen) = Kokonaisluku, joka määrittää neljännen ulottuvuuden solujen määrän.
- **ulottuvuus5_elementit** (valinnainen) = Kokonaisluku, joka määrittää viidennen ulottuvuuden solujen määrän.

Katso myös: [CLEARARRAY](#), [DIM](#)

ESIMERKKI

```
Dim ARRAY(8)

Redim ARRAY(12)
```

REPEAT

KUVAUS

Muodostaa silmukan, jota toistetaan niin kauan, kunnes loppuehto on TOSI. Loppuehto voi olla joko **Forever** (silmukka pyörii loputtomiin) tai **Until** (silmukka pyörii, kunnes ehto on TOSI).

Normaalisti kaikissa ohjelmissa on jonkin sortin pääsilmukka, jonka sisällä kaikki pelin toiminnot päivitetään.

Silmukasta voidaan rikkoutua kesken kaiken pois komennolla **Exit**.

KÄYTTÖ

```
REPEAT  
    'koodia  
UNTIL ehto
```

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
i=1  
  
Repeat  
  
    Print "Press RETURN to QUIT."  
  
    'If 10 000 iterations have passed break free  
    i=i+1  
    If i>10000 Then Exit  
  
Until KeyHit(cbkeyreturn)
```

REPLACE

KUVAUS

Korvaa merkkijonosta esiintymiä toisella.

Isoilla ja pienillä kirjaimilla ON väliä.

Korvaaminen suoritetaan koko merkkijonolle.

Tämä toiminto on tuttu erilaisista tekstinkäsittelyohjelmista.

Komento on ihanteellinen, kun haluat esim. poistaa turhia merkkejä, kuten välilyöntejä.

KÄYTTÖ

```
REPLACE (jono, etsittävä, korvaava)
```

- **jono** = Merkkijono.
- **etsittävä** = Merkkijonoesiintymä jonka haluat korvattavan.
- **korvaava** = Tällä korvataan. Jos haluat poistaa esiintymän, pistä tähän tyhjä merkkijono.

ESIMERKKI

```
theline$="Don't you worry, babe. I'm an expert."
```

```
'Print the original
```

```
Print theline$
```

```
'Print replace "babe" with "John"
```

```
Print Replace(theline$,"babe", "John")
```

```
'Print remove the "you" word
```

```
Print Replace(theline$,"you ","")
```

```
WaitKey
```


RESETOBJECTCOLLISION

KUVAUS

Jättää väliaikaisesti huomioimatta kaikki objektille asetetut törmäykset.

KÄYTTÖ

RESETOBJECTCOLLISION objektimuuttuja

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [SETUPCOLLISION](#)

ESIMERKKI

```
FrameLimit 40

'load map
map=LoadMap("Media\testmap.til","Media\negative.bmp")

'load player
mrpac=LoadObject("Media\pacman.bmp",36)

'setup a collision between the player and map
SetupCollision mrpac,map,1,4,2

AddText "Arrows to move..."
AddText "HOLD SPACE to ignore collision"

Repeat

    'Controls
    If LeftKey() Then TurnObject mrpac,10
    If RightKey() Then TurnObject mrpac, -10
    If UpKey() Then MoveObject mrpac,5
    If DownKey() Then MoveObject mrpac, -5

    If KeyDown(57) Then 'space
        ResetObjectCollision mrpac
    EndIf

    'camera follow
    CloneCameraPosition mrpac

    DrawScreen

Until EscapeKey()
```

RESIZEIMAGE

KUVAUS

Tämä komento suurentaa tai pienentää kuvaa. Komennolle välitetään uusi koko pikseleinä.

Voit skaalata kuvaa myös negatiivisilla arvoilla. Näin voit **peilata** kuvan vaaka- tai pystysuunnassa.

Nopeuden säästämiseksi CoolBasic ei filtteriä kuvaa millään tavalla. Siksi venytetyn kuvan pikselit näkyvät. Blineaarisen filtteröinnin saa päälle komennolla **Smooth2D**, jolloin skaalatut kuvat pehmenevät.

HUOM! Tämä operaatio on suhteellisen raskas, eikä sitä voi käyttää reaaliajassa. Hyvä tapa on venyttää kuvat valmiiksi ohjelman alussa. Muuten pelisi hidastuu huomattavasti.

KÄYTTÖ

RESIZEIMAGE kuvamuuttuja, leveys, korkeus

- **kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.
- **leveys** = Kokonaisluku. Uusi leveys pikseleinä.
- **korkeus** = Kokonaisluku. Uusi korkeus pikseleinä.

Katso myös: [SMOOTH2D](#)

ESIMERKKI

```
Dim images(80)

FrameLimit 40

Text 0,0,"Pre-calculating.. Please wait"
DrawScreen

'First Pre-calculate all images
masterimage= LoadImage("Media\cow.bmp")

'Generate 80 resized versions from the master Image
For i=1 To 80
    images(i)=CloneImage(masterimage)

    newwidth#=ImageWidth(masterimage)*(i/10.0)
    newheight#=ImageHeight(masterimage)*(i/10.0)

    ResizeImage images(i),newwidth,newheight
Next i

'The actual game-loop
plus=1
i=1
Repeat

    DrawImage images(i),0,0

    'animate
    i=i+plus
    If i<1 Then i=1: plus= -plus
    If i>80 Then i=80: plus= -plus

    DrawScreen

Until EscapeKey()
```

RESIZEMEMBLOCK

KUVAUS

Muuttaa muistipalan kokoa. Edellinen sisältö säilytetään, joskin muistipalan paikka tietokoneen muistissa saattaa vaihtua.

KÄYTTÖ

RESIZEMEMBLOCK muistipala, koko

- muistipala** = Se muuttuja, mihin muistipala luotiin.
- koko** = Uusi koko tavuina. Kokonaisluku.

ESIMERKKI

```
'create a memory block with 64 bytes
mem=MakeMEMblock(64)

'print the old size
memsize=MEMblockSize(mem)
Text 0,0,memsize

'increase its size to 128 bytes
ResizeMEMblock mem,128

'print the new size
memsize=MEMblockSize(mem)
Text 0,20,memsize

DrawScreen

WaitKey

DeleteMEMblock mem 'free the memory we used
```

RESTORE

KUVAUS

Asettaa datalistan lukukohdan tiettyyn kohtaan ohjelmassa.

KÄYTTÖ

RESTORE kirjanmerkki

Kirjanmerkkiä pitää seurata datalista.

Katso myös: [DATA](#), [READ](#)

ESIMERKKI

```
Dim map(6,5)

'-----

'read level 1
Restore level1

For y=1 To 5
For x=1 To 6
    map(x,y)=READ()
Next x
Next y

'-----

'main loop
Repeat

    For y=1 To 5
    For x=1 To 6
        If map(x,y) Then Box x*40,y*40,40,40
    Next x
    Next y

    Box 40,40,240,200,OFF

    DrawScreen

Until EscapeKey()
End

'-----
'you can store more levels, too.

level1:

DATA 0,0,0,0,0,0
DATA 0,0,0,1,0,0
DATA 0,0,1,1,1,0
DATA 0,0,0,1,0,0
DATA 1,0,0,0,0,1
```

RETURN

KUVAUS

Tämä komento palauttaa funktiosta arvon tai palauttaa aliohjelman sen kutsupaikkaan.

Jos palautat funktiosta arvoa, sinun täytyy kirjoittaa komennon perään palautettava muuttuja tai lauseke. Pääohjelmassa komennon perään ei kirjoiteta mitään.

Huomaa, että pystyt palauttamaan erilaisia arvoja, kuin mitä funktion tyyppiä alun perin määriteltiin. On suositeltavaa siten varmistaa oikean tyyppinen palautusarvo **Int**-, **Float**- ja **Str**-funktioilla tai oikean tyyppisellä muuttujalla.

KÄYTTÖ

```
RETURN [palautusarvo]
```

Palautusarvo kirjoitetaan vain silloin, kun palaat funktiosta. Se voi olla mikä tahansa lauseke.

Katso myös: [FUNCTION](#), [GOSUB](#)

ESIMERKKI

Repeat

```
c=Input("Give tempereture in Celcius degrees? ")
```

```
DrawScreen
```

```
Until KeyHit(28) 'loop until RETURN-key
```

```
CloseInput
```

```
'-----
```

Repeat

```
Text 0,0,c+" Celcius equals to "+Fahrenheit(c)+" Fahrenheit"
```

```
DrawScreen
```

```
Until EscapeKey()
```

```
'-----
```

```
'FUNCTIONS
```

```
'-----
```

```
'This function converts Celcius tempereature into
```

```
'Fahrenheit temperature
```

```
Function Fahrenheit(degrees)
```

```
fah=(degrees-32)/1.8
```

```
Return fah
```

```
End Function
```

RIGHT

KUVAUS

Lukee merkkijonon lopusta tietyn määrän merkkejä.

Soveltuu esimerkiksi merkkijonojen pituuden rajoittamiseen.

Katso esimerkki.

KÄYTTÖ

```
RIGHT (jono, montako)
```

- **jono** = Merkkijono, josta luetaan.
- **montako** = Montako merkkiä oikealta luetaan. Positiivinen kokonaisluku.

Katso myös: [LEFT](#), [MID](#)

ESIMERKKI

```
mystring$="Hello World"

For i=1 To Len(mystring$)
    Text 10, i*10, Right(mystring$,i)
Next i

DrawScreen

WaitKey
```

RIGHTKEY

KUVAUS

Palauttaa 1, jos oikeaa nuolinäppäintä painetaan. Muutoin palautetaan nolla.

KÄYTTÖ

RIGHTKEY ()

Katso myös: [LEFTKEY](#), [UPKEY](#), [DOWNKEY](#)

ESIMERKKI

AddText "Use arrows to move..."

x=200:y=150

Repeat

'update controls
If LeftKey() Then x=x-1
If RightKey() Then x=x+1
If UpKey() Then y=y-1
If DownKey() Then y=y+1

'draw "player"
Box x,y,10,10

DrawScreen

Until EscapeKey()

RND

KUVAUS

Generoi satunnaisluvun annetulta väliltä. Luku on desimaaliluku.

Käytetään lisäämään satunnaisuutta ja vivahteikuutta.

KÄYTTÖ

```
RND ([mistä], mihin)
```

- **mistä** (*valinnainen*) = Pienin alaraja. Desimaaliluku.
- **mihin** = Suurin yläraja. Desimaaliluku.

Katso myös: [RAND](#), [RANDOMIZE](#)

ESIMERKKI

```
Text 10,10,Rnd(-10,10)
Text 10,20,Rnd(6) '0-6
Text 10,30,Rnd(55.4,66.7)
```

```
DrawScreen
WaitKey
```


ROTATECAMERA

KUVAUS

Kääntää kameran määrättyyn kulmaan. Kulma annetaan asteina.

KÄYTTÖ

ROTATECAMERA kulma

- **kulma** = Kokonaisluku tai desimaaliluku. Mihin kulmaan kamera käännetään.

Katso myös: [TURNCAMERA](#)

ESIMERKKI

```
FrameLimit 40

floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

guy=LoadObject("Media\guy.bmp")

Repeat

    'Perform this code every 40. times
    '(once a second since FrameLimit is 40)
    If t=0 Then
        RotateCamera Rand(360)
        t=40
    Else
        t=t-1
    EndIf

    MoveCamera 1

    'draw some 2d-graphics, too
    DrawGame
    Line 200,150,200+Cos(CameraAngle())*100,150-Sin(CameraAngle())*100

    DrawScreen

Until EscapeKey()
```

ROTATEIMAGE

KUVAUS

Tämä komento kierittää kuvaa halutun verran. Kulma annetaan asteina.

HUOM! Kun kuvaa kierretään, sen hallintapiste siirtyy vasemmasta yläkulmasta kuvan keskelle piirrettyjen kuvien helpottamiseksi ohjelmoijaa varten. Kuva piirretään keskipisteestä lähtien.

Nopeuden säästämiseksi CoolBasic ei filtteröi kuvaa millään tavalla. Siksi kierretyn kuvan pikselit näkyvät. Lineaarisen filteröinnin saa päälle komennolla **Smooth2D**, jolloin kierretyt kuvat pehmenevät. Tasakulmat (90,180,270) kannattaa kääntää esim. 90.1, jotta illuusio ääriviivoista pysyisi.

HUOM! Tämä operaatio on suhteellisen raskas, eikä sitä voi käyttää reaaliajassa. Hyvä tapa on kiertää kuvat valmiiksi ohjelman alussa. Muuten pelisi hidastuu huomattavasti.

KÄYTTÖ

ROTATEIMAGE kuvamuuttuja, kulma

- **kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.
- **kulma** = Kokonaisluku tai desimaaliluku asteina.

Katso myös: [SMOOTH2D](#)

ESIMERKKI

```
Dim images(360)

FrameLimit 40

Text 0,0,"Pre-calculating.. Please wait"
DrawScreen

Smooth2D ON

'First Pre-calculate all images
masterimage= LoadImage("Media\cow.bmp")

'Generate 360 rotated versions from the master Image
For i=0 To 360
    images(i)=CloneImage(masterimage)

    RotateImage images(i), -i
Next i

Color cbblack
ClsColor cbwhite

'The actual game-loop
Repeat

    'Draw at the center
    DrawImage images(angle),200,150

    'change the angle
    angle=WrapAngle(angle+1)

    Text 0,0,"Angle: "+angle

    DrawScreen

Until EscapeKey()
```

ROTATEOBJECT

KUVAUS

Kääntää objektin täsmälleen haluttuun kulmaan. Välittömästi.

Kulma tulisi antaa väliltä 0-360.

KÄYTTÖ

```
ROTATEOBJECT objektimuuttuja, kulma
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **kulma** = Kokonaisluku tai desimaaliluku. Mihin kulmaan objekti käännetään.

Katso myös: [TURNOBJECT](#), [POINTOBJECT](#), [CLONEOBJECTORIENTATION](#)

ESIMERKKI

```
FrameLimit 40 'limit game speed

soldier=LoadObject("Media\soldier.bmp",36)

'make another object
soldier2=CloneObject(soldier)
PositionObject soldier2,0, -100

ClsColor cbwhite

Repeat

    'update this code every 1.5 seconds
    If TIMER()>moment+1500 Then
        'increase soldier angle
        RotateObject soldier,ObjectAngle(soldier)+10

        'rotate to random angle
        RotateObject soldier2,Rand(360)

        moment=TIMER()
    EndIf

    DrawScreen

Until EscapeKey()
```

ROUNDDOWN

KUVAUS

Pyöristää desimaaliluvun alaspäin. Tuloksena on kokonaisluku.

Huomaa, että miinusmerkkisillä luvuilla pyöristetään tavallaan "väärään suuntaan", kohti pienempää lukua. Siten esim. **-2.1 -> -3**.

Katso myös Int ja RoundUp erityyppisiä pyöristyksiä varten.

KÄYTTÖ

ROUNDDOWN (luku)

- **luku** = Jokin kokonais- tai desimaaliluku.

Katso myös: [ROUNDUP](#), [INT](#)

ESIMERKKI

```
Repeat
    value#=Input("Input a decimal number? ")

    DrawScreen
Until KeyHit (cbKeyReturn)

CloseInput

Repeat
    Text 20,20,"RoundDown: "+RoundDown(value#)

    Text 20,50,"Hit Escape to quit..."

    DrawScreen
Until EscapeKey()
```

ROUNDUP

KUVAUS

Pyöristää desimaaliluvun ylöspäin. Tuloksena on kokonaisluku.

Huomaa, että miinusmerkkisillä luvuilla pyöristetään tavallaan "väärään suuntaan", kohti suurempaa lukua. Siten esim. **-2.7 -> -2**.

Katso myös Int ja RoundDown erityyppisiä pyöristyksiä varten.

KÄYTTÖ

ROUNDUP (luku)

- **luku** = Jokin kokonais- tai desimaaliluku.

Katso myös: [ROUNDDOWN](#), [INT](#)

ESIMERKKI

```
Repeat
    value#=Input("Input a decimal number? ")

    DrawScreen
Until KeyHit (cbKeyReturn)

CloseInput

Repeat
    Text 20,20,"RoundUp: "+RoundUp(value#)

    Text 20,50,"Hit Escape to quit..."

    DrawScreen
Until EscapeKey()
```

RSET

KUVAUS

Asettaa merkkijonon absoluuttisen pituuden. Lisää ylimääräisiä välilyöntejä merkkijonon alkuun niin, että merkkijonosta tulee halutun pituinen. Teksti pysyy oikealle tasattuna.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

```
RSET (jono, pituus)
```

- **jono** = Tasattava merkkijono.
- **pituus** = Uusi pituus välilyöntien lisäyksen jälkeen.

Katso myös: [LSET](#)

ESIMERKKI

```
st$=Rset("Hastalavista, baby!",28)
```

```
Text 10,10,"#" + st$ + "#"
```

```
DrawScreen
```

```
WaitKey
```

SAFEEXIT

KUVAUS

Safe Exit on ominaisuus, jonka avulla käyttäjä voi milloin tahansa lopettaa pelin painamalla ESC. Tämä toiminto on oletuksena päällä.

Toiminnon avulla ohjelman voi lopettaa, vaikka se päätyisi loppumattomaan silmukkaan tai käyttäjä unohtaisi kirjoittaa tarvittavat koodit ohjelman lopettamiseksi.

Toisaalta tämä ominaisuus ei aina ole tervetullut, ja sen saa toki pois päältä. Ohjelmoija voi esimerkiksi haluta, että ESC-näppäin toisikin ruudulle valikon, josta pelaaja voi vielä valita KYLLÄ tai EI.

Safe Exit ei toimi silloin, kun **Wait**-komento on vielä kesken.

Toisenlaisen lopetuksen eston (estä X-nappulan klikkaaminen) saat komennolla SetWindow.

KÄYTTÖ

SAFEEXIT päällä

- **päällä**

OFF = toiminto pois päältä

ON = toiminto päälle

ESIMERKKI

```
SAFEEXIT OFF 'disable escape auto-exit
```

```
AddText "Press escape..."
```

```
Repeat
```

```
    DrawScreen
```

```
Until EscapeKey()
```

```
Text 20,20,"You pressed escape!"
```

```
DrawScreen
```

```
ClearKeys
```

```
WaitKey
```

SAR

KUVAUS

Suorittaa luvulle binäärisiirron oikealle. Toinen luku määrittää siirron suuruuden.

Käytännössä tämä on nopea jakolasku (nopeampi kuin /). Sitä käytetään jakamaan lukua eksponentiaalisesti, joten aivan kaikkiin lukuihin ei päästä.

Siirtämällä oikealle kerran, jaat luvun kahdella. Siirtämällä kaksi, jaat neljällä. Siirtämällä kolme, jaat kahdeksalla jne...

Sar eroaa Sar:sta täyttämällä tyhjät bitit etumerkillä: 0 positiivisille luvuille ja 1 negatiivisille.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

a SAR b

- **a** = luku
- **b** = siirtojen määrä

Katso myös: [SHL](#), [SHR](#)

ESIMERKKI

```
Text 0,0,(16 SHR 2)
DrawScreen
WaitKey
'Prints "4" because 16/4=4
```


SAVEIMAGE

KUVAUS

Tallentaa kuvan levyille BMP-muotoon.

KÄYTTÖ

SAVEIMAGE kuvamuuttuja, tiedostonimi, [frame]

- kuvamuuttuja** = Se muuttuja, joka sisältää kuvan.
- tiedostonimi** = Merkkijono, .BMP-pääte.
- frame** (*valinnainen*) = Kokonaisluku. Kuvan animaatio-frame. Oletuksena nolla.

Katso myös: [SCREENSHOT](#)

ESIMERKKI

```
'Create an image
img= MakeImage(256, 256)

'Draw to it
DrawToImage img

    'Draw 100 random colour circles
    For i=1 To 100
        Color Rand(255), Rand(255), Rand(255)
        Box Rand(255),Rand(255),Rand(20,100),Rand(20,100)
    Next i

DrawToScreen

'Put our master-piece onto screen
DrawImage img, 0, 0

Color cbwhite
Text 10,270,"Press any key to save this to 'C:\CBart.bmp'"

'Refresh screen
DrawScreen

WaitKey

'Save the image
SaveImage img, "C:\CBart.bmp"
```

SAVEPROGRAM

KUVAUS

Tämä komento tallentaa pelin kaikki muuttujat ja taulukot tallennustiedostoon, joka voidaan myöhemmin ladata, ja pelitilanne palauttaa.

Huomaa, että komento ei tallenna tietoa kuvista, äänistä eikä mistään muustakaan ladatusta mediasta. Latauksen jälkeen kaikki kuvat ja muu media on ladattava takaisin ennen kuin peli voi jatkaa. Komento ei siis sellaisenaan sovi suuren pelin pikatallennukseen/lataukseen.

VAROITUS! Pelin tallennus on erittäin herkkä virhetilanteille. Tätä ominaisuutta tulisi käyttää vasta kun peli on valmis. Yhdenkin muuttujan lisääminen ohjelmaan tallennustiedoston tekemisen jälkeen voi sotkea niiden arvot latausvaiheessa. Siksi on hyvä mainita, että tallennustiedostot eivät ole yhteensopivia päivitettyjen peliversioiden kanssa.

Suosittelemme edelleen, että kirjoitat oman tallennus- ja latauskoodin peliäsi varten.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

SAVEPROGRAM tiedostonimi

- **tiedostonimi** = Merkkijono, tallennustiedoston nimi tai polku.

Katso myös: [LOADPROGRAM](#)

ESIMERKKI

```
'Prevent the game from running too fast
FrameLimit 40

AddText "Move around using the arrows"
AddText "F5 = Save game"
AddText "F7 = Load game"

'Default Position
x=200
y=150

Repeat

    'Upadte controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1

    'draw the ball
    Color cbred
    Circle x-16,y-16,32,32

    'Save/load
    If KeyHit(cbkeyf5) Then SaveProgram "savegame.sav"
    If KeyHit(cbkeyf7) Then LoadProgram "savegame.sav"

    s$=Input("?")

    Text 300,0,x
    Text 300,20,y

    DrawScreen

Until EscapeKey()
```

SCREEN

KUVAUS

Tämä komento toimii sekä käskynä että funktiona.

Jos käytät tätä **komentona**, se vaihtaa ikkunan kokoa tai näyttötilaa. Voit halutessasi asettaa ikkunan venytettäväksi. Tällöin käytetään vanhaa ikkunakokoa, mutta uutta resoluutiota. Yleensä kannattaakin asettaa ensin normaali ikkunatila, ja heti perään venytetty ikkunatila. Oletusruudun koko on 400, 300.

Valinnaisen TILA-parametrin kohdalle voi kirjoittaa jonkin seuraavista sisäänrakennetuista vakioista: **cbFullScreen**, **cbWindowed**, **cbSizable**.

HUOM! Komento tuhoaa muistista kaikki kuvat ja objektit. Ohjelma kaatuu, jos yrität käyttää vanhoja kuvia ja objekteja. Komennon jälkeen on hyvä ladata tarvittavat kuvat ja objektit takaisin.

Jos käytät tätä **funktiona**, se osoittaa näyttöpuskuriin mm. CopyBoxin ja pikselikomentojen kanssa. Lisätietoa noiden kommentojen oppaista.

KÄYTTÖ

```
SCREEN leveys, korkeus, [värisyvyys], [tila]
tai
SCREEN ()
```

- **leveys** = Kokonaisluku. Ruudun/ikkunan leveys pikseleinä.
- **korkeus** = Kokonaisluku. Ruudun/ikkunan korkeus pikseleinä.
- **wärisyvyys** (**valinnainen**) = Kokonaisluku. 0 (automaattinen), 16, 24 tai 32. Riippuu mitä tiloja näytönohjain tukee.
- **tila** (**valinnainen**)
 - 0** = koko ruutu
 - 1** = ikkuna (oletus)
 - 2** = venytetty ikkuna

ESIMERKKI

```
Print "Press any key to change screen..."
WaitKey

SCREEN 500,350
Print "width 500 \ height 350"
Print
Print "Press any key to change screen..."
WaitKey

SCREEN 250,175,0,cbsizable
Print "width 500 \ height 350 \ sizable"
Print
Print "Press any key to change screen..."
WaitKey

SCREEN 640,480,16,cbfullscreen
Print "FullScreen 640, 480, in 16-bit mode"
Print
Print "Press any key to quit..."
WaitKey
```

SCREENDEPTH

KUVAUS

Kertoo näyttötilan värisyvyyden, joka on yleensä 16 tai 32.

Palautettu luku on kokonaisluku.

KÄYTTÖ

```
SCREENDEPTH ()
```

ESIMERKKI

Repeat

```
Text 10,10,"Your desktop colour depth is "+ScreenDepth()+"-bit."
```

```
DrawScreen
```

```
Until EscapeKey ()
```

SCREENGAMMA

KUVAUS

Säätää näytön värisävyä ajon aikana. Voit esimerkiksi häivyttää koko näytön mustaksi tai valkoiseksi. Näytön on oltava kokoruudun tilassa tai komento ei toimi. Katso esimerkki.

Käytä komentoa vain kerran loopin aikana. Sitä ei myöskään kannata käyttää jokaisella kerralla loopissa, vaan päivittää vain tarvittaessa.

Voit käyttää väriarvojen tilalla CoolBasicin sisäänrakennettuja [värivakioita](#).

KÄYTTÖ

SCREENGAMMA punainen, vihreä, sininen

- **punainen** = Kokonaisluku. Punainen komponentti.
- **vihreä** = Kokonaisluku. Vihreä komponentti.
- **sininen** = Kokonaisluku. Sininen komponentti.

ESIMERKKI

```
SCREEN 640,480,16,cbfullscreen

themap=LoadImage("Media/map.bmp")

While Not EscapeKey()

    Cls

    'First draw the map
    DrawImage themap,100,50

    oldgamma=gamma

    If LeftKey() Then gamma=gamma-1
    If RightKey() Then gamma=gamma+1

    If gamma< -255 Then gamma= -255
    If gamma>255 Then gamma=255

    If KeyDown(cbkeyreturn) Then
        'hue red
        ScreenGamma 255,0,0
    Else
        'if changed, update gamma
        If oldgamma<>gamma Then ScreenGamma gamma,gamma,gamma
    EndIf

    Text 0,0,"Use arrows to adjust screen gamma."
    Text 0,14,"Press Return for Blood Fill"

    DrawScreen

Wend

End
```

SCREENHEIGHT

KUVAUS

Kertoo ruudun tai ikkunan korkeuden pikseleinä.

Palautettu luku on kokonaisluku.

KÄYTTÖ

```
SCREENHEIGHT ()
```

Katso myös: [SCREENWIDTH](#)

ESIMERKKI

```
word$="This text is aligned to right bottom corner"
```

Repeat

```
Text ScreenWidth() -TextWidth(word),ScreenHeight() -TextHeight(word),word
```

DrawScreen

```
Until EscapeKey()
```

SCREENPOSITIONOBJECT

KUVAUS

Asettaa objektin haluttuun sijaintiin ikkunan/ruudun mukaan. Voit esimerkiksi sijoittaa objektin hiiren koordinaatteihin (MouseX, ja MouseY) niin saat vaikka animoidun hiirisoittimen!

Koordinaatisto ruudun mukaan:



KÄYTTÖ

SCREENPOSITIONOBJECT objektimuuttuja, x, y

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **x** = Ruudun X-koordinaatti.
- **y** = Ruudun Y-koordinaatti.

Katso myös: [POSITIONOBJECT](#), [CLONEOBJECTPOSITION](#), [CLONECAMERAPOSITION](#)

ESIMERKKI

```
'first make the background
lawn=MakeObjectFloor()
PaintObject lawn,LoadImage("Media\grass.bmp")

obj=LoadObject("Media\cow.bmp",72) 'load the 'cursor'

PositionCamera Rand( -500,500),Rand( -500,500) 'throw us somewhere random

AddText "Move around with the arrows"
AddText "Positioning an object as screen coordinates..."

Repeat

    'Move the camera via arrows
    If LeftKey() Then TranslateCamera -1,0
    If RightKey() Then TranslateCamera 1,0
    If UpKey() Then TranslateCamera 0,1
    If DownKey() Then TranslateCamera 0, -1

    TurnObject obj,5

    'Position the object at mouse
    ScreenPositionObject obj,MouseX(),MouseY()

    DrawScreen

Until EscapeKey()
```

SCREENSHOT

KUVAUS

Ottaa pelistä kuvankaappauksen ja tallentaa sen levyllä BMP-tiedostoon.

Kuva otetaan mielellään **juuri ennen** DrawScreen-komentoa.

Tämä on suositeltavampi tapa kuin PrintScreenin käyttö.

KÄYTTÖ

SCREENSHOT tiedostonimi

- **tiedostonimi** = Merkkijono. Tallennettavan kuvakaappauksen nimi tai tiedostopolku. Tiedostopäätteen pitää olla .BMP.

ESIMERKKI

```
Text 0,0,"Hit F12 when you want to take a screenshot"
Text 0,20,"The screenshot will be saved to C:\capture.bmp"
Text 0,40,"Press any key to begin..."
DrawScreen
WaitKey

Repeat

    'draw 100 random boxes
    For i=1 To 100
        Color Rand(255),Rand(255),Rand(255)
        Box Rand(400),Rand(300),Rand(100),Rand(100)
    Next i

    'take snapshot
    If KeyHit(cbkeyf12) Then SCREENSHOT "C:\capture.bmp"

    DrawScreen

Until EscapeKey()
```


SCREENWIDTH

KUVAUS

Kertoo ruudun tai ikkunan leveuden pikseleinä.

Palautettu luku on kokonaisluku.

KÄYTTÖ

```
SCREENWIDTH ()
```

Katso myös: [SCREENHEIGHT](#)

ESIMERKKI

```
word$="This text is aligned to right bottom corner"
```

Repeat

```
    Text ScreenWidth() -TextWidth(word),ScreenHeight() -TextHeight(word),word
```

```
    DrawScreen
```

```
Until EscapeKey()
```

SEEKFILE

KUVAUS

Joka kerta kun tiedostosta luetaan tai tiedostoon kirjoitetaan, sen sisäinen tiedosto-osoitin muuttaa paikkaa (riippuen montako tavua tiedostoon kirjoitetaan). Kun tiedosto avataan, sen tiedosto-osoitin on aina alussa, nollassa (huom, ei yksi).

Tällä komennolla voit siirtää tiedosto-osoittimen paikkaa tiedostossa, ja asettaa seuraavan kohdan, johon tietoa kirjoitetaan tai tietoa luetaan. Huomaa, että laskuri lähtee nollasta, mikä on tiedoston ensimmäinen paikka.

Käytä tätä komentoa tiedoston *päivittämiseen*. Tämä on erityisen hyvä tapa muuttaa tiedoston sisältöä ilman, että pitää kirjoittaa koko tiedosto tai sen loppu uusiksi. Huomaa, että kokonaisluvut (int) vievät 4 tavua tai 2 tavua (short). Desimaaliluvut puolestaan vievät myös 4 tavua. Merkkijonojen kanssa pitää olla tarkkana, koska niillä ei ole vakiopituutta.

Jos tiedosto sisältää vain 4 tavun kokonaislukuja, pääset käsiksi esim. 7. lukuun komentamalla **SeekFile tiedosto, 24** (koska 4*7-4=24).

Käytä **OpenToEdit**-funktia tiedoston avaamiseen, jos käytät SeekFileä tai FileOffsetia.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

SEEKFILE tiedosto, paikka

- **tiedosto** = Tiedostomuuttuja.
- **paikka** = Uusi paikka tiedostossa. Lähtee nollasta. Kokonaisluku.

Katso myös: [OPENTOREAD](#), [OPENTOWRITE](#), [OPENTOEDIT](#)

ESIMERKKI

```
'open the file for output and write 10 integers to it
f=OpenToWrite("Media\temp.dat")
  For i=1 To 10
    WriteInt f,i
  Next i
CloseFile f

'-----

'open the file for editing
f=OpenToEdit("Media\temp.dat")
'replace the 6th Integer with a New value
  SeekFile f, 4*5
  WriteInt f,12345
CloseFile f

'-----

'open the file for input and print all integers from it
f=OpenToRead("Media\temp.dat")
  For i=1 To 10
    val=ReadInt(f)
    Text 10,i*12, val
  Next i
CloseFile f

DrawScreen

WaitKey
```

SELECT

KUVAUS

Toivottavasti tunnet jo IF-lauseen, koska tämä on vähän sama. Select-rakenne vastaa sarjaa ElseIf-lausekkeita. Esitystapa on vain hieman erilainen, ehkäpä mukavampi.

Select-rakenteella voidaan tarkastaa useita ehtoja. Se vertaa sille annettuun arvoon muita esiteltyjä arvoja. Komento on ihanteellinen, kun katsotaan minkälaisia arvoja muuttujassa on. Rakenne menee näin:

- 1) SELECT muuttuja
- 2) CASE arvo
- 3) komento
- 4) [CASE arvo]
- 5) [komento]
- 6) [DEFAULT]
- 7) [komento]
- 8) END SELECT

Hakasulkeella olevat kohdat ovat valinnaisia. Kuulostaako hämmäntävältä? Otetaan esimerkki:

```
TARKASTELE muuttujaa
    ONKO 1
        TEE JOTAIN...
    ONKO 2
        TEE JOTAIN
LOPETA TARKASTELU
```

Oikeastaan se tekee siis saman, kuin sarja IF- tai ELSEIF-käskyjä. CoolBasicin kielellä tämä olisi:

```
SELECT muuttuja
    CASE 1
        PRINT "yksi"
    CASE 2
        PRINT "kaksi"
END SELECT
```

Jos mikään ehdoista ei täsmää, voit haluta ottaa senkin huomioon. DEFAULT-käsky vastaa IF-rakenteen ELSE-käskyä.

```
SELECT muuttuja
    CASE 1
        PRINT "yksi"
    CASE 2
        PRINT "kaksi"
    DEFAULT
        PRINT "joku muu"
END SELECT
```

Voit myös yhdistellä ehtoja samoin kuin IF-rakenteessa OR-avainsanan avulla:

```
SELECT muuttuja
    CASE 1
        PRINT "yksi"
    CASE 2, 3
        PRINT "kaksi tai kolme"
    DEFAULT
        PRINT "joku muu"
END SELECT
```

KÄYTTÖ

```
SELECT vertailuarvo
    CASE lista
        'komentoja...
END SELECT
```

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
'A TEXT ADVENTURE
'=====

beginning:
ClearText
ClearKeys
CloseInput
```

```

AddText "What would you like to do?"
AddText ""
AddText "1. Go out"
AddText "2. Go to fishing"
AddText "3. Eat an apple"
AddText ""

Repeat
    c=Input("?")

    DrawScreen

Until KeyHit(cbkeyreturn)

Select c
    Case 1
        GoTo outside
    Case 2
        GoTo fishing
    Case 3
        GoTo eat
    Default
        AddText "Invalid choise"
        DrawScreen
        Wait 2000
        GoTo beginning
End Select

'-----

outside:
ClearText
ClearKeys
CloseInput

AddText "OK, you're on your yard. What to do now?"
AddText ""
AddText "1. Go back inside"
AddText "2. Go to fishing"
AddText ""

Repeat
    c=Input("?")

    DrawScreen

Until KeyHit(cbkeyreturn)

Select c
    Case 1
        GoTo beginning
    Case 2
        GoTo fishing
    Default
        AddText "Invalid choice"
        DrawScreen
        Wait 2000
        GoTo outside
End Select

'-----

fishing:
ClearText
ClearKeys
CloseInput

AddText "You noticed you are quite a good angler."
AddText "You finished this game, congrats!"

DrawScreen
Wait 10000
End

'-----

eat:
ClearText
ClearKeys

```

CloseInput

AddText "Yamm.. no need of Doctor for a while."

DrawScreen

Wait 3000

GoTo beginning

SETFONT

KUVAUS

Tämä komento aktivoi aikaisemmin ladatun fontin, ja kaikki tekstikomennot *Printiä* ja *Writea* lukuunottamatta siirtyvät käyttämään uutta fonttia.

KÄYTTÖ

SETFONT fonttimuuttuja

- **fonttimuuttuja** = Se muuttuja, johon fontti ladattiin *LoadFont*-komennolla.

Katso myös: [LOADFONT](#), [DELETEFONT](#)

ESIMERKKI

```
arial=LoadFont("arial",20)
impact=LoadFont("impact",28)
system=LoadFont("fixedsys")
coolb=LoadFont("coolb")
verdana=LoadFont("verdana",18,ON,OFF,ON) 'bold and underlined
times=LoadFont("times new roman",20,OFF,ON) 'italic
courier=LoadFont("courier new",40)
```

Repeat

```
Color cborange
SetFont arial
Text 40,20,"I'll look into it"

Color cblightgreen
SetFont impact
Text 100,100,"A sound plan!"

Color cbyellow
SetFont system
Text 260,270,"Dead man walking"

Color cbcyan
SetFont system
Text 20,200,"This..is..my BOOMSTICK!"

Color cbsilver
SetFont verdana
Text 70,220,"What do you want?!"

Color cblightred
SetFont times
Text 150,180,"Night beckons"

Color cbgreen
SetFont courier
CenterText 200,130,"Don't worry, be happy!"
```

DrawScreen

Until EscapeKey()

SETMAP

KUVAUS

Tällä komennolla voit piilottaa ja näyttää tilekartan kerroksia. Tilekartoissa on vain kaksi piirrettävää kerrosta: tausta ja päällys (numerointi 0 ja 1). Objektit piirretään näiden kahden väliin. Jos pelisi käyttää vain taustakerrosta, voi päällisen kytkeä pois näkyvistä, mikä nopeuttaa hieman suorituskykyä. Jos taas haluat taustan läpinäkyväksi, ja karttasi käyttää vain päällyskerrosta, voit kytkeä taustan pois päältä. Taustalla voi olla Floor-objekti, joka liikkuu hieman eri tasossa karttaan nähden (eri Z-syvyys). Tällä tavalla saat hienon syvyysvaikutelman sivulta kuvattuihin tasoloikkiin. Esimerkki valaisee asiaa.

KÄYTTÖ

SETMAP karttamuuttuja, taustakerros, päällikerros

- **karttamuuttuja** = Se objektimuuttuja, mihin kartta ladattiin.
- **taustakerros**

ON = piirrä

OFF = älä piirrä

- **päällikerros**

ON = piirrä

OFF = älä piirrä

ESIMERKKI

```
'Make an infinite background
grass=MakeObjectFloor()
PaintObject grass,LoadImage("Media\grass.bmp")
'Make it run as it were far away
PositionObject grass,0,0, -2

'Load the map
map=LoadMap("Media\testmap.til","Media\tileset.bmp")

AddText "Use Arrows to move..."
AddText "F1 = toggle back layer on/off"
AddText "F2 = toggle fore layer on/off"

'Defaults
back=1
fore=1

PositionCamera 0, -90
RotateCamera 0,90 'make the camera face upwards

PlayObject map,0,0,1

'The game-loop itself
Repeat

    If KeyHit(cbkeyf1) Then back=Not back
    If KeyHit(cbkeyf2) Then fore=Not fore

'Controls
If LeftKey() Then MoveCamera -1,0
If RightKey() Then MoveCamera 1,0
If UpKey() Then MoveCamera 0,1
If DownKey() Then MoveCamera 0, -1

SetMap map,back,fore 'update map properties

'Let's draw the game world before the text
'Otherwise the text would be covered by the map
DrawGame

Text 20,70,"BACK: "+back
Text 20,82,"FORE: "+fore

DrawScreen

Until EscapeKey()
```

SETSOUND

KUVAUS

Muuttaa äänen olemusta. Äänen pitää olla jo soimassa, jotta tätä komentoa voisi käyttää. Tätä varten ääni piti soittaa funktiona, jotta saat kanavamuuttujan. Katso esimerkki :)

Voit muuttaa äänen voimakkuutta, sijaintia ja taajuutta. Sijainti muuttaa äänen paneerausta vasemman ja oikean kaiuttimen välillä. Saat näin vaikka hienon vaikutelman helikopterin ohilentämisestä.

Voit asettaa *ladatun* äänen jatkuvaksi soitoksi komentamalla: **SetSound kanavamuuttuja,ON**

KÄYTTÖ

SETSOUND kanavamuuttuja, efekti, [voimakkuus], [sijainti], [taajuus]

- **kanavamuuttuja** = Kanavamuuttuja, jonka sait PlaySound-funktiolta.
- **efekti** = Äänen erikoissoittotapa: **OFF** ei toistoa, **ON** toistuva ääni.
- **voimakkuus** (*valinnainen*) = Luku väliltä 0-100. Nolla on ei ääntä, 100 on täysillä. Oletuksena 100. Tätä parametria ei ole pakko kirjoittaa.
- **sijainti** (*valinnainen*) = Luku väliltä -100 ja 100. Nolla on keskitetty ääni, -100 on vasen kaiutin ja 100 oikea kaiutin. Oletuksena 0. Tätä parametria ei ole pakko kirjoittaa.
- **taajuus** (*valinnainen*) = Kokeile lukuja 8000-44000. Ilmoitetaan hertseinä. Jätä pois, jos et halua muuttaa taajuutta.

Katso myös: [LOADSOUND](#)

ESIMERKKI

```
AddText "Hit SPACE to play a laser sound"
AddText "Try arrows"

vol=100 'default full volume

Repeat

    'endless loop of music
    If SoundPlaying(music)=OFF Then
        music=PlaySound("Media\SK_Battle2.mp3")
    EndIf

    'change pan
    If LeftKey() Then pan=pan-1
    If RightKey() Then pan=pan+1
    If pan< -100 Then pan= -100
    If pan>100 Then pan=100

    'change volume
    If UpKey() Then vol=vol+1
    If DownKey() Then vol=vol-1
    If vol<0 Then vol=0
    If vol>100 Then vol=100

    'apply changes
    SetSound music,0,vol,pan

    Text 40,40,"Volume: "+vol
    Text 40,55,"Pan: "+pan

    DrawScreen

Until EscapeKey()
```


SETTILE

KUVAUS

Käytä tätä komentoa kartan rakentamisessa. Tämä komento animoi tilesetin, eli sen kuvan, jota käytetään tilekartan piirtämiseen. Jokaisella tilellä on oma järjestysnumerosa, jotka alkavat kuvan vasemmasta ylälaidasta ja juoksevat lukusuuntaan - riviltä riville. Esimerkiksi:

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

Voit animoida jokaisen tilen erikseen. Se tapahtuu antamalla tilen jälkeen animaatioon osallistuvien tilejen määrän. Esim. arvo 1 animoi tilen ja yhden sen jälkeen. Sekvenssin pituus on siis kaksi. Aseta tilen animointi pois numerolla 0 (tai OFF). Viimeistä tilesetin tileä ei voi animoida; sekvenssi ei kierrä tilesetin alkuun.

Voit vaihtoehtoisesti asettaa sekvenssin hitauskertoimen, eli sen kuinka monta kertaa normaalia hitaammin animaatio kiertää. Numero 1 on normaali, ja 2 kaksinkertainen hitaus jne. Tämä mahdollistaa vaihtelevannopeuksiset sekvenssit. Oletuksena on 1 (normaali).

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

```
SETTILE tile, pituus, hitaus
```

- **tile** = Kokonaisluku. Tilen järjestysnumero (1, 2, 3, ...).
- **pituus** = Kokonaisluku. Montako tileä animoidun tilen jälkeen osallistuu sekvenssiin (0=ei animaatiota).
- **hitaus** (*valinnainen*) = Kokonaisluku. Hitauskerroin. Oletuksena 1.

Katso myös: [MAKEMAP](#), [EDITMAP](#)

ESIMERKKI

```
'Create a tilemap
map=MakeMap(9,9,32,32)

'fill it randomly
For y=1 To 9
For x=1 To 9
  c=Rand(1,3)
  If c=1 Then
    EditMap map,0,x,y,2
  ElseIf c=2
    EditMap map,0,x,y,4
  Else
    EditMap map,0,x,y,6
  EndIf
Next x
Next y

'Load a tileset and apply it to the map
tileset=LoadImage("Media\tileset.bmp")
PaintObject map,tileset

'Set some animation sequences
SetTitle 2,1
SetTitle 4,1,3 'slowness factor 3

'-----

'animate map
PlayObject map,0,0,0.5

Repeat
  DrawScreen
Until EscapeKey()
```

SETUPCOLLISION

KUVAUS

Tässä se on. Kaikkien rakastama automaattinen törmäyksen tunnistus. Etkä tarvitse juuri muuta kuin tämän komennon! Tällä komennolla esittelet CoolBasicille mitkä objektit voivat törmätä keskenään ja siten eivät voi mennä päällekkäin tai kulkea lävitse. Tämä tapahtuu esittelemällä CoolBasicille kaksi objektia, törmäyksen tyytit sekä menettelytapa törmäyksen sattuessa. Aivan kaikkia mahdollisia törmäysyhdistelmiä ei ole vielä saatavilla, mutta niitä lisätään tulevilla ohjelmapäivityksissä. Voit esitellä seuraavat törmäykset:

Törmäävä	->	Törmättävä	Menettelytavat	Esitys
 (laatikko) 1	->	 (laatikko) 1	liukuva 2	1,1,2
 (laatikko) 1	->	 (kartta) 4	liukuva 2	1,4,2
 (pallo) 2	->	 (pallo) 2	pysähdys (1) liukuva (2)	2,2,1 2,2,2
 (pallo) 2	->	 (kartta) 4	pysähdys (1) liukuva (2)	2,4,1 2,4,2

Jos et tajunnut taulukosta mitään, otetaan esimerkki. Haluat pallo-pallo -törmäyksen. Haluat sen liukuvana. Joten katsot taulukosta... se on siis 2,2,2.

Karttaa tai liikkumatonta pelimaailmaa koskevat törmäykset kannattaa esitellä viimeiseksi.

KÄYTTÖ

SETUPCOLLISION objektimuuttuja, kohde_objekti, lähdetyyppi, kohdetyyppi, menettelytapa

- objektimuuttuja** = Se objekti, jonka pitäisi törmätä johonkin.
- kohde_objekti** = Se objekti, **johon** voidaan törmätä.
- lähdetyyppi** = Minkätyyppinen törmäys lähdeobjektilla on? 1=laatikko, 2=pallo.
- kohdetyyppi** = Minkätyyppinen törmäys kohdeobjektilla on? 1=laatikko, 2=pallo.
- menettelytapa** = Miten menetellään törmäyksen sattuessa? 1=pysähdy, 2=liukuva.

Katso myös: [COUNTCOLLISIONS](#) [CLEARCOLLISIONS](#)

ESIMERKKI

```

FrameLimit 40

'load map
map=LoadMap("Media\testmap.til","Media\negative.bmp")

'load player
guy=LoadObject("Media\guy.bmp",72)
PositionObject guy,0, -40

'load a neutral unit
fellow=LoadObject("Media\guy.bmp",72)
PositionObject fellow, -100, -40

'setup a circle-circle sliding collision to the fellow
SetupCollision guy,fellow,2,2,2

'setup a collision between the player and map
SetupCollision guy,map,1,4,2

Repeat
    'Controls
    If LeftKey() Then TurnObject guy,5
    If RightKey() Then TurnObject guy, -5
    If UpKey() Then MoveObject guy,3
    If DownKey() Then MoveObject guy, -3

    'camera follow
    UpdateGame
    CloneCameraPosition guy
    
```

DrawScreen

Until EscapeKey ()

SETWINDOW

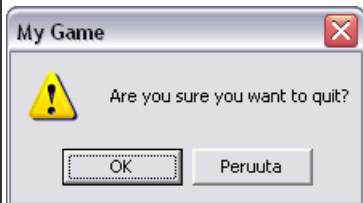
KUVAUS

Tätä komentoa voidaan käyttää kolmeen tarkoitukseen:

- Asettaa ikkunan otsikko
- Suurentaa/pienentää ikkuna
- Asettaa ikkunan lopetuskysymys, kun käyttäjä painaa X-painiketta ikkunan yläkulmasta

Ikkunan otsikon asettaminen on komennon yleisin käyttötarkoitus.

Ikkunoiduissa peleissä käyttäjä voi lopettaa pelin ikkunan X-painikkeesta. Estääksesi tämän (peli on lopetettava *sinun* tavallasi), voit asettaa erityisen kysymyksen, joka antaa käyttäjän vielä peruuttaa valinta. Kysymyslaatikossa on OK-, ja Peruuta-painikkeet.



Tämän lisäksi ohjelmoija voi suurentaa ikkunan koko ruutuun, palauttaa sen normaalitilaan tai pienentää sen tehtäväpalkkiin. Nämä toimivat ainoastaan ikkunoidussa tilassa. Lisäksi ikkunan ruutuun suurentaminen toimii ainoastaan grafiikkatilassa, joka on esitetty parametrilla *cbSizable*. Lisätietoja **SCREEN**-komennon ohjeesta. **HUOM!** Tämän ominaisuuden väärinkäyttö voi aiheuttaa outoja lopputuloksia. Suosittelemme, että kirjoitat tämän tilalle *nollan*, jos et tiedä mitä teet.

KÄYTTÖ

SETWINDOW otsikko, [erityistila], [lopetuskysymys]

- **otsikko** = Merkkijono, ikkunan otsikkopalkissa näkyvä teksti. Oletuksena tyhjä.
- **erityistila** (valinnainen)
0 = ei toimenpiteitä (oletus)
1 = normaali ikkunakoko
2 = pienennä tehtäväpalkkiin
3 = suurennä koko ruudulle
- **lopetuskysymys** (valinnainen) = Merkkijono. Viesti, joka näytetään jos käyttäjä yrittää lopettaa ohjelmaa ikkunan X-painikkeesta. Oletuksena ei mitään.

ESIMERKKI

```
SCREEN 400,300,16,cbsizable

SetWindow "Hello World!",0,"Are you sure?"

AddText "Try to close the window by clicking the X"
AddText ""
AddText "F1 = minimize"
AddText "F2 = maximize"
AddText "F3 = normalize"

Repeat

    If KeyHit(cbkeyf1) Then SetWindow "Hello World!",2,"Are you sure?"
    If KeyHit(cbkeyf2) Then SetWindow "Hello World!",3,"Are you sure?"
    If KeyHit(cbkeyf3) Then SetWindow "Hello World!",1,"Are you sure?"

DrawScreen

Until EscapeKey()
```

SHL

KUVAUS

Suorittaa luvulle binäärsiirron vasemmalle. Toinen luku määrittää siirron suuruuden.

Käytännössä tämä on nopea kertolasku (nopeampi kuin *). Sitä käytetään kertomaan lukua eksponentiaalisesti, joten aivan kaikkiin lukuihin ei päästä.

Siirtämällä vasemmalle kerran, kerrot luvun kahdella. Siirtämällä kaksi, kerrot neljällä. Siirtämällä kolme, kerrot kahdeksalla jne...

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

a SHL b

- a = luku
- b = siirtojen määrä

Katso myös: [SHR](#), [SAR](#)

ESIMERKKI

Text 0,0,(3 SHL 4)

DrawScreen

WaitKey

'Prints "48" because 3*16=48

SHOWMOUSE

KUVAUS

Tämä komento näyttää tai piilottaa hiirikursorin ikkunoidussa tilassa.

Komento ei toimi koko ruudun tilassa.

Voit myös asettaa oman hiirikursorin kirjoittamalla komennon perään sen kuvamuuttujan nimen, mihin kursori on ladattu. Tämä toimii myös koko ruudun tilassa.

Huomioi, että jos käytät omaa hiirikursoria, kuvan hallintapiste (normaalisti kuvan vasemmassa yläreunassa) siirtyy pysyvästi kuvan keskelle. Tämä tarkoittaa sitä, että kuvan keskipiste on hiirikursorin kohdalla. Tämä on erityisen kätevää, kun halutaan tehdä hiirikursorista tähtäin. Siksi "perinteisen" kursorin kuvan pitäisi näyttää jokseenkin tältä:



Kun taas tähtäinkuva voisi näyttää esim. tältä:



KÄYTTÖ

SHOWMOUSE tyyli

- tyyli

OFF = piilota kursori

ON = näytä normaali kursori

kuvamuuttuja = näytä oma kursori

ESIMERKKI

```
AddText "F1 = hide cursor"
```

```
AddText "F2 = show cursor"
```

```
AddText "F3 = custom pointer"
```

```
AddText "F4 = custom target"
```

```
pointer=LoadImage("d:\cursor.bmp")
```

```
crosshair=LoadImage("d:\crosshair.bmp")
```

```
Repeat
```

```
  If KeyHit(cbkeyf1) Then ShowMouse OFF
```

```
  If KeyHit(cbkeyf2) Then ShowMouse ON
```

```
  If KeyHit(cbkeyf3) Then ShowMouse pointer
```

```
  If KeyHit(cbkeyf4) Then ShowMouse crosshair
```

```
  DrawScreen
```

```
Until EscapeKey()
```

SHOWOBJECT

KUVAUS

Tämä komento piilottaa tai näyttää objektin. Piilotettuja objekteja ei voi poimia, eikä niille tarkasteta törmäystä. Piilotettuja objekteja ei piirretä mitenkään, joten ne eivät juuri vie suorituskykyä. Jos haluat tehdä objektin näkymättömäksi, mutta haluat sen osallistuvan törmäykseen, komenna **GhostObject objekti,0**.

Yleensä objekti piilotetaan, kun siitä halutaan tehdä ns. Master-objekti. Siitä luodaan klooneja, mutta Master-objektiin ei kosketa. Lisätietoa **CloneObject**in oppaasta.

KÄYTTÖ

SHOWOBJECT objektimuuttuja, näkyvä

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **näkyvä**
ON = näkyvä
OFF = piilotettu

ESIMERKKI

```
map=LoadImage("Media\map.bmp")

guy=LoadObject("Media\guy.bmp")

RotateObject guy,90

AddText "Press SPACE to show/hide object"

'defaults to true (visible)
vis=1

Repeat

    DrawImage map,0,0

    If KeyHit(cbkeyspace) Then vis= Not vis
    ShowObject guy,vis

    DrawScreen

Until EscapeKey()
```

SHR

KUVAUS

Suorittaa luvulle binäärisiirron oikealle. Toinen luku määrittää siirron suuruuden.

Käytännössä tämä on nopea jakolasku (nopeampi kuin /). Sitä käytetään jakamaan lukua eksponentiaalisesti, joten aivan kaikkiin lukuihin ei päästä.

Siirtämällä oikealle kerran, jaat luvun kahdella. Siirtämällä kaksi, jaat neljällä. Siirtämällä kolme, jaat kahdeksalla jne...

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

a SHR b

- **a** = luku
- **b** = siirtojen määrä

Katso myös: [SHL](#), [SAR](#)

ESIMERKKI

Text 0,0,(16 SHR 2)

DrawScreen

WaitKey

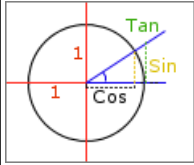
'Prints "4" because 16/4=4

SIN

KUVAUS

Laskee kulman sinin. Kulma annetaan asteina.

Tätä funktiota käytetään lähinnä apukeinona laskemaan paljonko pisteen pitää liikkua pystysuunnassa, kun tiedetään liikkeen suunta (kulma) ja nopeus.



Oletetaan, että haluat laskea seuraavat ammuksen koordinaatit. Ammus on laukaistu kulmassa 60 astetta, nopeudella 5. Tästä voidaan muodostaa suorakulmainen kolmio. Ammus kulkee vinoa sivua pitkin. Sin laskee tämän kolmion pystysivun pituuden. Siten **y# = y# - Sin (60) * 5**.

Tulos on aina desimaaliluku väliltä -1 ja 1.

Sinin pariin tarvitaan kosinia (kts. Cos).

KÄYTTÖ

SIN (luku)

- **luku** = Kokonaisluku tai desimaaliluku.

Katso myös: [COS](#), [ASIN](#)

ESIMERKKI

```
FrameLimit 40 'prevent the game from running too fast

AddText "Arrows to aim"
AddText "Space to shoot"
AddText "Esc to exit"

angle=90

Repeat
    'draw the ground
    Line 0,150,400,150

    'draw the turret
    Line 200,150,200+Cos(angle)*20,150-Sin(angle)*20

    If LeftKey() Then angle=angle+1
    If RightKey() Then angle=angle-1

    If angle<10 Then angle=10
    If angle>170 Then angle=170

    'shoot?
    If KeyHit(cbkeyspace) And bulletflying=0 Then
        bulletx#=200+Cos(angle)*20
        bullety#=150-Sin(angle)*20

        bulletmovex#=Cos(angle)*2 '2=speed
        bulletmovey#=-Sin(angle)*2 '2=speed

        bulletflying=1
    EndIf

    'update and draw bullet
    If bulletflying Then

        bulletx#=bulletx#+bulletmovex#
        bullety#=bulley#+bulletmovey#

        Dot bulletx#,bulley#

        'if the bullet goes outside screen, remove it
        If bulletx#<0 Or bulletx#>400 Or bulley#<0 Then
```

```
bulletflying=0
EndIf
```

```
EndIf
```

```
'draw the screen
DrawScreen
```

```
Until EscapeKey()
```

SMOOTH2D

KUVAUS

Asettaa pehmeät kuvamanipulaatiot käyttöön tai pois käytöstä.

Kun kuvaa käsitellään komennoilla **RotateImage** ja **ResizeImage**, CoolBasic suorittaa ne mahdollisimman nopeasti. Tämä voi jättää kuvaan karkeutta, ja esimerkiksi venytetyn kuvan pikselit suurenevat ja jäävät näkyviin.

Jos tämä toiminto kytketään päälle, CoolBasic silottaa kuvaa, ja antaa pehmennetyn vaikutelman. Toiminnot käyttävät silloin lineaarista suodatusta, joka on kuitenkin hieman raskaampi toteuttaa.

Komento vaikuttaa myös objekteihin.

Toiminto on oletuksellisesti pois päältä.

KÄYTTÖ

SMOOTH2D päällä

- **päällä**
ON = piirrä pehmeästi
OFF = piirrä karkeasti

ESIMERKKI

```
FrameLimit 40

Text 0,0,"Loading..."
DrawScreen

Dim images(2,360)

obj1=LoadImage("Media\cow.bmp"): ResizeImage obj1,60,50
obj2=LoadImage("Media\cow.bmp"): ResizeImage obj2,60,50

'operate obj1
For i=0 To 360
    img=CloneImage(obj1)
    RotateImage img, -i
    images(1,i)=img
Next i

Smooth2D ON

'operate obj2
For i=0 To 360
    img=CloneImage(obj2)
    RotateImage img, -i
    images(2,i)=img
Next i

'THE MAIN LOOP
'-----

ClsColor cbwhite
Color cbblack

Locate 20,200

AddText "The first one isn't so smooth"
AddText "The second one is smoother"

ClsColor cbwhite

Repeat

    angle=WrapAngle(angle+1)

    DrawImage images(1,angle),140,40

    DrawImage images(2,angle),240,40

    DrawScreen

Until EscapeKey()
```

SOUNDPLAYING

KUVAUS

Palauttaa 1, jos ääni on soimassa. Nolla, jos ei. Käytä tätä funktiota kertomaan onko ääni jo loppunut. Tämä on kätevää, jos haluat esim. toistaa musiikkia loputtomiin.

KÄYTTÖ

SOUNDPLAYING (kanavamuuttuja)

- **kanavamuuttuja** = Kanavamuuttuja, jonka sait PlaySound-funktiolta.

ESIMERKKI

```
AddText "Hit SPACE to play/stop a music"
```

```
playing=OFF
```

```
Repeat
```

```
    'user controls
```

```
    If KeyHit(cbkeyspace) Then
```

```
        playing=Not playing
```

```
        If playing Then
```

```
            music=PlaySound("Media\SK_Battle2.mp3")
```

```
        Else
```

```
            StopSound music
```

```
        EndIf
```

```
    EndIf
```

```
    'print a funny text
```

```
    If SoundPlaying(music) Then
```

```
        Text 40,40,"Music is playing :)"
```

```
    Else
```

```
        Text 40,40,"Music is not playing :("
```

```
        playing=OFF
```

```
    EndIf
```

```
    DrawScreen
```

```
Until EscapeKey()
```

SQRT

KUVAUS

Palauttaa annetun luvun neliöjuuren.

Neliöjuuri on se luku, joka kertomalla itsellään saadaan alkuperäinen luku. Esim. luvun 9 neliöjuuri on 3, koska $3*3=9$.

Neliöjuurta ei voi ottaa negatiivisista luvuista.

Neliöjuuren tulos on aina desimaaliluku.

KÄYTTÖ

SQRT (luku)

- **luku** = Kokonaisluku tai desimaaliluku.

ESIMERKKI

```
For i=2 To 5
    number=i*i
    Text 20, i*12, "The square root of "+number+" is "+Sqrt(number)
Next i

DrawScreen
WaitKey
```

STARTSEARCH

KUVAUS

Käytä tätä komentoa ennen **FindFileä**. Se valmistelee nykyisen hakemiston tiedostonetsintään.

Etsintä täytyy muistaa lopettaa komennolla **EndSearch**.

KÄYTTÖ

```
STARTSEARCH
```

Katso myös: [FINDFILE](#), [ENDSEARCH](#)

ESIMERKKI

```
'Change directory to Media
ChDir "Media"

'Prepare search
StartSearch

    'Scan all files and print all .BMP images
    count=0
    Repeat

        file$=FindFile()

        If file$="" Then Exit 'no more files

        If Lower(Right(file$,4))=".bmp" Then
            Print file$

            count=count+1
        EndIf

    Forever

'End search
EndSearch

Print "Found "+count+" files"

WaitKey
```

STOPANIMATION

KUVAUS

Komento sulkee käynnissä olevan animaation. Komennon ihanteellinen käyttötapa on pysäyttää pelin alkuvideo kun jotain näppäintä painetaan. Keskeneräinen video tulee aina lopettaa, koska videot pyörivät taustalla, vaikka niitä ei piirrettäisikään DrawAnimation-komennolla.

KÄYTTÖ

```
STOPANIMATION videomuuttuja
```

- **videomuuttuja** = Se muuttuja, mihin video käynnistettiin PlayAnimation-funktiolla.

Katso myös: [PLAYANIMATION](#), [ANIMATIONPLAYING](#)

ESIMERKKI

```
SCREEN 640,480

'load the video
video= PlayAnimation("myMovie.avi")

'draw video
Repeat

    DrawAnimation video

    DrawScreen

Until EscapeKey() Or KeyHit(28)

'close the video input
StopAnimation video
```

STOPOBJECT

KUVAUS

Pysäyttää objektin animaation, ja jättää sen nykyiseen frameen.

Jos haluat pysäyttää animaation niin, että se palaa alkuasentoon, komenna mieluummin **PlayObject objekti,OFF**

KÄYTTÖ

STOPOBJECT objektimuuttuja

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.

Katso myös: [PLAYOBJECT](#), [LOOPOBJECT](#)

ESIMERKKI

```
'make screen a bit zoomed
SCREEN 250,188,0,cbsizable

FrameLimit 40 'limit game speed

'load animation for both directions
zeroright=LoadAnimObject("Media\zerorun1.bmp",47,51,0,12)
MaskObject zeroright,cbmagenta

zeroleft=LoadAnimObject("Media\zerorun2.bmp",47,51,0,12)
MaskObject zeroleft,cbmagenta
ShowObject zeroleft,OFF

'default direction
zero=zeroright

Color cbblack
AddText "Try arrows"
ClsColor cbwhite

'MAIN LOOP
Repeat

    'run right!
    If RightKey() Then
        ShowObject zeroright,ON
        ShowObject zeroleft,OFF
        zero=zeroright
        LoopObject zero,2,11,0.25,ON 'animate at speed 0.25
    'run left!
    ElseIf LeftKey() Then
        ShowObject zeroleft,ON
        ShowObject zeroright,OFF
        zero=zeroleft
        LoopObject zero,2,11,0.25,ON 'animate at speed 0.25
    Else
        'nothing is pressed so freeze
        StopObject zero
    EndIf

    DrawScreen

Until EscapeKey()
```


STOPSOUND

KUVAUS

Lopettaa äänen toistamisen.

KÄYTTÖ

STOPSOUND kanavamuuttuja

- **kanavamuuttuja** = Kanavamuuttuja, jonka sait PlaySound-funktiolta.

ESIMERKKI

```
AddText "Hit SPACE to play/stop a music"
```

```
playing=OFF
```

```
Repeat
```

```
    'user controls
```

```
    If KeyHit(cbkeyspace) Then
```

```
        playing=Not playing
```

```
        If playing Then
```

```
            music=PlaySound("Media\SK_Battle2.mp3")
```

```
        Else
```

```
            StopSound music
```

```
        EndIf
```

```
    EndIf
```

```
    'print a funny text
```

```
    If SoundPlaying(music) Then
```

```
        Text 40,40,"Music is playing :)"
```

```
    Else
```

```
        Text 40,40,"Music is not playing :("
```

```
        playing=OFF
```

```
    EndIf
```

```
    DrawScreen
```

```
Until EscapeKey()
```

STR

KUVAUS

Muuntaa minkä tahansa arvon merkkijonoksi.

Jotkin komennot ja funktiot vaativat merkkijonon. Käytä tätä funktiota muuntamaan esim. kokonaislukuja merkkijonoiksi, jonka jälkeen se kelpaa parametriksi.

Muunnettaessa desimaalilukua desimaalipiste, miinusmerkki ja eksponentti säilytetään.

KÄYTTÖ

STR (arvo)

- **arvo** = Kokonaisluku, desimaaliluku tai merkkijono. Mielellään jokin lukuarvo.

Katso myös: [INT](#), [FLOAT](#)

ESIMERKKI

```
'initialize a number
number = 50012

'convert it to string and store to variable "num"
num$ = Str(number)

'extract last two characters from it
lasttwo$ = Right(num$,2)

'print the result
Text 10,10,"as string: " + lasttwo$
Text 10,30,"as integer: " + Int(lasttwo$)

DrawScreen

WaitKey
```

STRING

KUVAUS

Rakentaa uuden merkkijonon toistamalla annettua merkkijonoa annettun määrä kertoja.

KÄYTTÖ

```
STRING (jono, määrä)
```

- **jono** = Toistettava merkki tai merkkijono.
- **määrä** = Montako kertaa merkkijonoa toistetaan. Kokonaisluku.

ESIMERKKI

```
AddText "Use LEFT and RIGHT arrow to change length"
```

```
length=1
```

```
Repeat
```

```
    If KeyHit(cbkeyleft) Then length-1
```

```
    If KeyHit(cbkeyright) Then length+1
```

```
    If length<0 Then length=0
```

```
    build$=String("Hello",length)
```

```
    Text 0,50,build$
```

```
    DrawScreen
```

```
Until KeyHit(cbkeyreturn)
```

STRINSERT

KUVAUS

Tämän funktion avulla käyttäjä voi lisätä merkkijonon keskelle pätkän tekstiä.

KÄYTTÖ

```
STRINSERT (jono, paikka, teksti)
```

- **jono** = Aluperäinen merkkijono.
- **paikka** = Mihin kohtaan tekstiä lisätään. Kokonaisluku.
- **teksti** = Lisättävä merkkijono.

ESIMERKKI

```
'original string
stri$="Don't worry babe, I'm an expert."
Text 10,10,stri$

'insert a new word at position 6
stri$=StrInsert(stri$,6,"you ")
Text 10,30,stri$

DrawScreen

WaitKey
```

STRMOVE

KUVAUS

Tämän funktion avulla käyttäjä voi siirtää pätkän tekstiä merkkijonon sisällä toiseen paikkaan.

KÄYTTÖ

STRMOVE (jono, paikka, pituus, siirto)

- **jono** = Aluperäinen merkkijono.
- **paikka** = Mistä kohtaa siirtäminen alkaa. Kokonaisluku.
- **pituus** = Siirrettävän pätkän pituus. Kokonaisluku.
- **siirto** = Montako merkkiä eteenpäin tekstiä siirretään. Kokonaisluku.

ESIMERKKI

```
'original string
stri$="Don't you worry, babe. I'm an expert."
Text 10,10,stri$

'move word "you" after "worry"
stri$=StrMove(stri$,6,4,6)
Text 10,30,stri$

DrawScreen

WaitKey
```

STRREMOVE

KUVAUS

Tämän funktion avulla käyttäjä voi poistaa merkkijonon keskeltä pätkän tekstiä.

KÄYTTÖ

STRREMOVE (jono, paikka, montako)

- **jono** = Alkuperäinen merkkijono.
- **paikka** = Mistä kohtaa tekstiä poistetaan. Kokonaisluku.
- **montako** = Poistettavien merkkien lukumäärä. Kokonaisluku.

ESIMERKKI

```
'original string
stri$="Don't worry babe, I'm an expert."
Text 10,10,stri$

'delete word "babe"
stri$=StrRemove(stri$,12,5)
Text 10,30,stri$

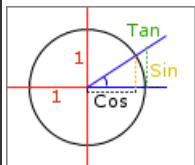
DrawScreen

WaitKey
```

TAN

KUVAUS

Laskee kulman tangentin. Kulma annetaan asteina.



Tangenttia ei voi ottaa pystysuorista kulmista (esim. 90,270,-90,-270).

Käytetään laskemaan paljonko on suorakulmaisen kolmion Y:n pituus, kun X on 1. Tangentti 45 asteen kulmasta on 1.

Tulos on desimaaliluku.

Komento on tarkoitettu edistyneemmille käyttäjille. Sen hyöty tavalliselle peliohjelmoijalle jäänee vähäiseksi.

KÄYTTÖ

TAN (luku)

- **luku** = Kokonaisluku tai desimaaliluku.

Katso myös: [ATAN](#)

ESIMERKKI

```
Text 0,0,TAN(30)
DrawScreen
WaitKey
```

TEXT

KUVAUS

Kirjoittaa tekstiä ruudulle nykyisellä fontilla. Teksti tasataan siten, että sen vasen yläkulma on annetuissa koordinaateissa.

Tulostettu teksti on taustaltaan läpinäkyvää.

Joillakin laitteistoilla on havaittu hidastumista, kun Text-komentoa käytetään. Jos haluat tulostaa sellaista tekstiä, joka ei muutu usein, käytä mieluummin AddTextiä. Ongelman on todettu häviävän, kun hakee näytönohjaimeensa uudet ajurit.

KÄYTTÖ

TEXT x, y, teksti

- **x** = Tekstin sijainti vaakasuunnassa. Kokonaisluku.
- **y** = Tekstin sijainti pystysuunnassa. Kokonaisluku.
- **teksti** = Tulostettava merkkijono.

Katso myös: [CENTERTEXT](#), [VERTICALTEXT](#), [ADDTXT](#)

ESIMERKKI

```
arial=LoadFont("arial",20)
impact=LoadFont("impact",28)
system=LoadFont("fixedsys")
coolb=LoadFont("coolb")
verdana=LoadFont("verdana",18,ON,OFF,ON) 'bold and underlined
times=LoadFont("times new roman",20,OFF,ON) 'italic
courier=LoadFont("courier new",40)
```

Repeat

```
Color cborange
SetFont arial
Text 40,20,"I'll look into it"
```

```
Color cblightgreen
SetFont impact
Text 100,100,"A sound plan!"
```

```
Color cbyellow
SetFont system
Text 260,270,"Dead man walking"
```

```
Color cbcyan
SetFont system
Text 20,200,"This..is..my BOOMSTICK!"
```

```
Color cbsilver
SetFont verdana
Text 70,220,"What do you want?!"
```

```
Color cblightred
SetFont times
Text 150,180,"Night beckons"
```

```
Color cbgreen
SetFont courier
CenterText 200,130,"Don't worry, be happy!"
```

```
DrawScreen
```

```
Until EscapeKey()
```


TEXTHEIGHT

KUVAUS

Kertoo merkkijonon korkeuden pikseleinä. Nykyinen fontti otetaan huomioon. Käytä tätä funktiota laskemaan tekstin koko *ruudulla*.
Tulos on kokonaisluku.

KÄYTTÖ

TEXTHEIGHT (jono)

- **jono** = Merkkijono, jonka korkeus määritetään.

Katso myös: [TEXTWIDTH](#), [LEN](#)

ESIMERKKI

```
AddText "Move mouse..."
```

```
Repeat
```

```
    xpos=MouseX()-TextWidth("Hasta la vista, baby!")
    ypos=MouseY()-TextHeight("Hasta la vista, baby!")
```

```
    'Print the text so its right bottom corner is at the mouse
    Text xpos,ypos,"Hasta la vista, baby!"
```

```
    DrawScreen
```

```
Until EscapeKey()
```

TEXTWIDTH

KUVAUS

Kertoo merkkijonon pituuden pikseleinä. Nykyinen fontti otetaan huomioon. Käytä tätä funktiota laskemaan tekstin koko *ruudulla*.

Tulos on kokonaisluku.

KÄYTTÖ

```
TEXTWIDTH (jono)
```

- **jono** = Merkkijono, jonka pituus määritetään.

Katso myös: [TEXTHEIGHT](#), [LEN](#)

ESIMERKKI

```
AddText "Move mouse..."
```

```
Repeat
```

```
    xpos=MouseX()-TextWidth("Hasta la vista, baby!")
    ypos=MouseY()-TextHeight("Hasta la vista, baby!")
```

```
    'Print the text so its right bottom corner is at the mouse
```

```
    Text xpos,ypos,"Hasta la vista, baby!"
```

```
    DrawScreen
```

```
Until EscapeKey()
```

TIME

KUVAUS

Kertoo tietokoneen kellonajan.

Tulos on merkkijono.

KÄYTTÖ

TIME ()

Katso myös: [DATE](#)

ESIMERKKI

Repeat

```
Text 40,40,Date() 'print date
Text 40,60,Time() 'print time
Text 40,80,TIMER() 'print system timer
```

DrawScreen

Until EscapeKey()

TIMER

KUVAUS

Kertoo järjestelmän ajastimen erittäin tarkasti. Järjestelmän ajastin lähtee nolasta kun tietokone käynnistetään. Se kasvaa 1000 kertaa sekunnissa, joten ajastimen avulla voidaan suorittaa millisekunnin tarkkoja ajanottoja.

Ajastinta ei heti luulisi tarvitsevan, mutta se osoittautunee kaikille ohjelmoijille erittäin hyödylliseksi - ainakin jossain vaiheessa. Ajastinta käytetään esimerkiksi suorittamaan tietyin väliajoin jokin tehtävä tai mittaamaan aikaa kahden tehtävän väliltä. Ajastinta voi käyttää esim. nopeusmittauksiin tai tietyn ajan odottamiseen.

Palautettu luku on kokonaisluku.

KÄYTTÖ

TIMER ()

ESIMERKKI

```
AddText "Press Return to halt program for 3 seconds..."

Repeat

    'update blink state every 100th millisecond
    If TIMER()>record+100 Then
        blink=Not blink
        record=TIMER()
    EndIf

    If blink Then Box 40,40,8,10

    'halt?
    If KeyHit(cbkeyreturn)

        halt=TIMER()
        Repeat
            Until TIMER()>halt+3000

        ClearKeys
    EndIf

    DrawScreen

Until EscapeKey()
```

TRANSLATECAMERA

KUVAUS

Liikuttaa kameraa pysty- ja vaakasuunnassa **piittaamatta** sen kulmasta. Tämä tarkoittaa sitä, että kamera voi pyöriä matkalla mielin määrin (esim. osoittaa kohti jotain objektia), mutta se pakotetaan liikkumaan tiettyä (suoraa) rataa pitkin.

KÄYTTÖ

TRANSLATECAMERA ylös, sivulle

- **ylös** = Kokonaisluku tai desimaaliluku. Paljonko kameraa liikutetaan pystysuunnassa.
- **sivulle** = Kokonaisluku tai desimaaliluku. Paljonko kameraa liikutetaan vaakasuunnassa.

Katso myös: [MOVECAMERA](#)

ESIMERKKI

```
FrameLimit 40

'Make scene
floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

guy=LoadObject("Media\guy.bmp")

AddText "Press SPACE for default camera position"

Repeat

    'move camera as absolute
    TranslateCamera 1,1

    'return to origin?
    If KeyDown(cbkeyspace) Then PositionCamera 0,0

    'Print the camera coordinates
    DrawGame
    Text 20,20,CameraX()
    Text 20,40,CameraY()

    DrawScreen

Until EscapeKey()
```

TRANSLATEOBJECT

KUVAUS

Liikuttaa objektia vaaka- ja pystysuunnassa piittaamatta sen kulmasta. Tämä on ihanteellinen tapa liikuttaa objektia määrätysä kulmassa silloin kun se esimerkiksi pyörii matkalla.

Parametrit kuvaavat nopeutta vaaka- ja pystykomponenteissa, joten käännettyssä kulmassa joudut soveltamaan **Sin**- ja **Cos**-funktioita.

Z-arvon muuttaminen on edistyneemmille käyttäjille. Sen avulla voidaan teeskennellä syvyysvaikutelmaa, mikä saa objektin piirtymään sijaintinsa puolesta perspektiivissä taustaa vasten.

KÄYTTÖ

TRANSLATEOBJECT objektimuuttuja, vaaka, pysty, [Z]

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **vaaka** = Lukuarvo, vaakanopeus.
- **pysty** = Lukuarvo, pystynopeus.
- **Z** (valinnainen, **edistyneemmille**) = Lukuarvo, paljonko objekti liikkuu syvyys suunnassa. Oletuksena nolla.

Katso myös: [MOVEOBJECT](#), [POSITIONOBJECT](#)

ESIMERKKI

```
Const gravity=0.05

'Prevent the game from running too fast
FrameLimit 40

'Load the ammunition, but hide it
fireball=LoadObject("Media\fireball.bmp",18)
ShowObject fireball,OFF

'load the cannon
cannon=LoadObject("Media\cannon.bmp",180)

'load the turret
turret=LoadObject("Media\turret.bmp")

PositionObject turret, -160, -35
PositionObject cannon, -160, -35

AddText "Use arrows to aim"
AddText "SPACE to shoot"

'Make "sky"
ClsColor cbdarkblue

'Initialize the game engine
permission=True

Repeat

    'update controls
    If LeftKey() Then angle=angle+2
    If RightKey() Then angle=angle-2
    If angle<10 Then angle=10
    If angle>170 Then angle=170

    'update SHOOTING
    If KeyHit(cbkeyspace) And permission=True
        ShowObject fireball,ON 'show the fireball

        'position and tilt it according to the cannon
        CloneObjectPosition fireball,cannon
        CloneObjectOrientation fireball,cannon

        'move a little bit away from the center
        MoveObject fireball,16

    permission=False

'Calculate the movement (speed 2)
```

```

        plusx#=Cos(angle)*3
        plusy#=Sin(angle)*3

EndIf

'Update moving fireball
If permission=False Then
    TranslateObject fireball,plusx#,plusy#
    plusy#=plusy-gravity

    'rotate the fireball to left
    TurnObject fireball,20

    'if the fireball goes outside screen, remove it
    If ObjectX(fireball)< -200 Or ObjectX(fireball)>200 Or ObjectY(fireball)>150 Or ObjectY(fireball)<
-50 Then
        ShowObject fireball,OFF
        permission=True
    EndIf
EndIf

'Turn the cannon
RotateObject cannon,angle

'draw ground
Color cbgreen
Box 0,200,400,100
Color cbwhite
Line 0,200,400,200

DrawScreen

Until EscapeKey()

```

TRIM

KUVAUS

Poistaa merkkijonon alusta ja lopusta ylimääräiset välilyönnit ja sarkaimet. Käytä tätä komentoa siistimään käyttäjän syötteitä tai tiedostosta luettuja rivejä.

Kun käsittelet merkkijonoja ja luet niistä tietoa, niiden on hyvä olla valmiiksi siistitty niin, että virheille jää vähemmän tilaa.

KÄYTTÖ

TRIM (jono)

- **jono** = Merkkijono, joka riisutaan...

ESIMERKKI

```
nastystring$="    I have nasty leading and tailing spaces  "

Text 0,0,"Before:"
Text 0,20,"#" +nastystring$+"#"

'remove leading and tailing spaces
nastystring$=Trim(nastystring$)

Text 0,60,"After:"
Text 0,80,"#" +nastystring$+"#"

DrawScreen
WaitKey
```


TURNCAMERA

KUVAUS

Kääntää kameran kulmaa vasemmalle tai oikealle: Positiiviset luvut vasemmalle, negatiiviset oikealle.

Huomaa, että tämä komento EI käännä kameraa määrättyyn kulmaan. Käytä tätä varten mieluummin **RotateCamera**:aa.

KÄYTTÖ

TURNCAMERA kulma

- **kulma** = Kokonaisluku tai desimaaliluku. Paljonko kameraa käännetään.

Katso myös: [ROTATECAMERA](#)

ESIMERKKI

```
FrameLimit 40

floor=MakeObjectFloor()
texture=LoadImage("Media\grass.bmp")
PaintObject floor,texture

guy=LoadObject("Media\guy.bmp")

Repeat

    'Make the camera run circle
    TurnCamera 2
    MoveCamera 1

    DrawScreen

Until EscapeKey()
```

TURNOBJECT

KUVAUS

Kääntää objektia halutun verran oikealle tai vasemmalle. **Eli menosuunta muttuu.** Täydellinen komento ylhäältä kuvattuihin peleihin, jossa kääntymistä ohjataan nuolinäppäimillä.

Mitä sulavampi kääntyvyysarvo objektille on annettu latausvaiheessa, sitä pehmeämmin objekti *näyttää* kääntyvän. Objektin menosuunta ei välttämättä ole sama kuin objektin käännetty kuva. Lisätietoa funktiosta **LoadObject**.

Positiiviset arvot kääntävät objektia vasemmalle, negatiiviset oikealle.

KÄYTTÖ

```
TURNOBJECT objektimuuttuja, kulma
```

- **objektimuuttuja** = Se muuttuja, mihin objekti ladattiin tai luotiin.
- **kulma** = Kokonaisluku tai desimaaliluku. Paljonko objektia käännetään.

Katso myös: [ROTATEOBJECT](#), [POINTOBJECT](#), [CLONEOBJECTORIENTATION](#)

ESIMERKKI

```
FrameLimit 40

soldier=LoadObject("Media\soldier.bmp",72)

AddText "Arrows to move"

Repeat
    'Update controls
    If LeftKey() Then TurnObject soldier,5
    If RightKey() Then TurnObject soldier, -5
    If UpKey() Then MoveObject soldier, 2
    If DownKey() Then MoveObject soldier, -2

    DrawScreen

Until EscapeKey()
```

TYPE ja FIELD

KUVAUS

Tämä koodilohko esittelee uuden TYYPIN ohjelmaasi. Siksi se kannattaa kirjoittaa ohjelman alkuun. Tyypit ovat kokoelmia, joihin voidaan luoda ja joista voidaan poistaa jäseniä. Kullakin jäsenellä on omat muuttujansa. Ne ovat ikään kuin itsenäisiä objekteja, jotka muodostavat listan.

Tyyppien avulla voit hallita suuria määriä vihollisia, ammuksia, efektejä ja objekteja helposti ja vaivattomasti.

TYYPPI-lohko lopetetaan komenttoon **END TYPE**. Älä unohda sitä! Tämän lohkon sisään taas kirjoitetaan **FIELD**-sanaa käyttäen kaikki muuttujat, jotka siis monistetaan automaattisesti jokaiselle jäsenelle.

Esimerkki kertoo kaiken tarpeellisen, mutta suosittelemme ehdottomasti, että luet myös [tämän](#).

KÄYTTÖ

```
TYPE nimi
  FIELD muuttuja1
  FIELD muuttuja2
  ...
END TYPE
```

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
Type BULLETS
  Field obj
End Type

' _ _ _ _ _
'MAIN PROGRAM
' _ _ _ _ _

FrameLimit 40

'load the soldier, turning detail 72
guy=LoadObject("Media\soldier.bmp",72)

'load the bullet (master-object) and hide it
bullet=LoadObject("Media\bullet.bmp")
ShowObject bullet,OFF

ClsColor cbdark

AddText "Arrows to turn, SPACE to shoot"

'MAIN LOOP
Repeat
  'Update turning
  If LeftKey() Then TurnObject guy,5
  If RightKey() Then TurnObject guy, -5

  'Shoot?
  If KeyDown(cbkeyspace) And reload=0

    newBullet.BULLETS= New(BULLETS)
    newBullet\obj=CloneObject(bullet)
    CloneObjectPosition newBullet\obj,guy
    CloneObjectOrientation newBullet\obj,guy
    'put the bullet at the end of the barrel
    MoveObject newBullet\obj,24

    reload=4
  EndIf

  'Update all BULLETS (move them forward)
  i=0
  For iBullet.BULLETS= Each BULLETS

    MoveObject iBullet\obj,6 'speed=6

    'delete bullet from collection if it
    'gets too far away
    If Distance2(iBullet\obj,guy)>200 Then
```

```
        DeleteObject iBullet\obj
        Delete iBullet
    EndIf
```

```
    i=i+1
```

```
Next iBullet
```

```
'Update reloading
```

```
If reload>0 Then reload=reload-1
```

```
Text 30,30,"Bullets to update: "+i
```

```
'Don't forget this :)
```

```
DrawScreen
```

```
Until EscapeKey()
```

UNLOCK

KUVAUS

Poistaa puskurin lukituksen. Puskuri pitää aina muistaa vapauttaa, kun lukitusta ei enää tarvita.

Komento on tarkoitettu edistyneemmille käyttäjille.

KÄYTTÖ

UNLOCK [puskuri]

- **puskuri** (valinnainen) = Puskuriosoitin. Kts. funktiot SCREEN() ja IMAGE(). Oletuksena nykyinen puskuuri.

Katso myös: [LOCK](#), [GETPIXEL2](#), [PUTPIXEL2](#), [SCREEN](#), [IMAGE](#)

ESIMERKKI

```
SCREEN 640,480

AddText "Press any key to copy..."

'First make a gradient
For i=0 To 240
    Color 0,i+15,0
    Line 0,i,640,i
Next i

DrawScreen OFF

WaitKey

'Lock screen
Lock

'make 154 481 iterations, copy each pixel
For y=0 To 240
For x=0 To 640
    PutPixel2 x,480-y,GetPixel2(x, y)
Next x
Next y

'Unlock screen
Unlock

DrawScreen

WaitKey
```

UPDATEGAME

KUVAUS

Päivittää pelimaailman animaatiot ja törmäyssysteemin.

Tätä komentoa ei ole pakko kutsua, vaan CoolBasic suorittaa sen automaattisesti ennen kuin ruutu piirretään.

Joskus UpdateGameea tarvitaan ennen komentoa **CloneCameraPosition**, jotta pieni kosmetiikkavirhe saadaan korjatuksi. Tämä koskee vain tilanteita, joissa seurattavaan hahmoon on asetettu törmäys jonkin muun pelimaailman objektin tai kartan kanssa. Kokeile esimerkkiä ilman UpdateGameea ja sen kanssa.

KÄYTTÖ

UPDATEGAME

ESIMERKKI

```
FrameLimit 30 'limit performance for the fast computers

map=LoadMap("Media\testmap.til","Media\tileset.bmp")
guy=LoadObject("Media\guy.bmp",36)
PositionObject guy,0,50

'make a collision between the guy and the map
SetupCollision guy,map,1,4,2

Repeat

    If LeftKey() Then TurnObject guy,10
    If RightKey() Then TurnObject guy, -10
    If UpKey() Then MoveObject guy,3
    If DownKey() Then MoveObject guy, -3

    'try to comment this and walk against the walls
    UpdateGame

    CloneCameraPosition guy

    DrawScreen

Until EscapeKey()
```

UPKEY

KUVAUS
Palauttaa 1, jos ylös-nuolinäppäintä painetaan. Muutoin palautetaan nolla.

KÄYTTÖ
UPKEY ()

Katso myös: [LEFTKEY](#), [RIGHTKEY](#), [DOWNKEY](#)

ESIMERKKI

```
AddText "Use arrows to move..."

x=200:y=150

Repeat
    'update controls
    If LeftKey() Then x=x-1
    If RightKey() Then x=x+1
    If UpKey() Then y=y-1
    If DownKey() Then y=y+1

    'draw "player"
    Box x,y,10,10

    DrawScreen

Until EscapeKey()
```

UPPER

KUVAUS

Muuttaa tekstin isolla kirjoitetuksi.

Toistaiseksi kaikkien maiden merkistöt eivät ole täysin tuettuja.

Komennon hyöty lienee muotoilun lisäksi kahden sanan tai lauseen yhtäläisyyden tarkastamisessa. Esimerkki valaisee asiaa.

KÄYTTÖ

```
UPPER (jono)
```

- **jono** = Merkkijono

Katso myös: [LOWER](#)

ESIMERKKI

```
'compare two differently written slogans

If Upper("I'll be Back")=Upper("i'LL Be bACk") Then
    Text 0,0,"Match"
Else
    Text 0,0,"Not a match"
EndIf

DrawScreen

WaitKey
```


VERTICALTEXT

KUVAUS

Kirjoittaa tekstiä ruudulle nykyisellä fontilla. Teksti tasataan siten, että sen vasen yläkulma on annetuissa koordinaateissa. oisin kuin **Text**, tämä kirjoittaa tekstin *pystysuoraan*.

Tulostettu teksti on taustaltaan läpinäkyvää.

Joillakin laitteistoilla on havaittu hidastumista, kun tätä komentoa käytetään. Ongelman on todettu häviävän, kun hakee näytönohjaimeensa uudet ajurit.

KÄYTTÖ

VERTICALTEXT x, y, teksti

- **x** = Tekstin sijainti vaakasuunnassa. Kokonaisluku.
- **y** = Tekstin sijainti pystysuunnassa. Kokonaisluku.
- **teksti** = Tulostettava merkkijono.

Katso myös: [TEXT](#), [CENTERTEXT](#)

ESIMERKKI

Repeat

```
'print the partial text vertically onto screen
VerticalText 30,30,Left("VERT!CAL",i)
```

```
'animate 10 times in second
```

```
If TIMER()>atime+100 Then
```

```
    i+1
```

```
    If i>8 Then i=1
```

```
    atime=TIMER()
```

```
EndIf
```

```
DrawScreen
```

```
Until EscapeKey()
```

WAIT

KUVAUS

Pysäyttää ohjelman suorituksen tietyksi ajaksi. Tänä aikana grafiikkaa ei piirretä, joten muut ikkunat voivat "pyyhkiä" ruudun tyhjäksi. Ohjelma ei myöskään reagoi syötteisiin hiireltä tai näppäimistöltä. Kaikki syötteet menevät niille varattuihin puskureihin, ja ne käsitellään, kun tauko päättyy. Tästä syystä joskus on hyvä käyttää **ClearKeys**- ja **ClearMouse**-komentoja heti Wait:n jälkeen.

Komentoa voi käyttää pelin hidastamiseen, mutta emme suosittele Wait:n käyttämistä. Suositeltavampia vaihtoehtoja ovat FrameLimit ja oma odotus-rutiini:

delay=TIMER():Repeat:Until TIMER(>delay+2000 'odottaa 2 sekuntia

Aika ilmoitetaan millisekunteina. 1000 millisekuntia vastaa yhtä sekuntia.

KÄYTTÖ

WAIT aika

- **aika** = Odotettava aika millisekunteina. Kokonaisluku.

Katso myös: [TIMER](#)

ESIMERKKI

```
CenterText 200,150,"The test is about to begin in 5 seconds..."
DrawScreen

Wait 5000 'wait 5 seconds

Repeat

    Text 10,10,counter
    counter=counter+1

    DrawScreen

    Wait 1000 'wait 1 second

Until EscapeKey()
```

WAITKEY

KUVAUS

Tämä komento toimii sekä käskynä että funktiona.

Komento pysäyttää ohjelman ja odottaa, kunnes käyttäjä painaa jotain näppäintä. Jos käytät tätä komentoa funktiona, palautetaan painetun näppäimen [SCAN-koodi](#).

KÄYTTÖ

```
WAITKEY  
tai  
WAITKEY ()
```

Katso myös: [WAITMOUSE](#)

ESIMERKKI

```
Print "Press any key..."  
  
WaitKey  
  
Print "OK, press anything again..."  
  
key=WaitKey() 'wait and read  
  
Print "You pressed: "+key  
  
'wait a key and close the program  
WaitKey
```

WAITKEY

KUVAUS

Tämä komento toimii sekä käskynä että funktiona.

Komento pysäyttää ohjelman ja odottaa, kunnes käyttäjä painaa jotain hiiren nappulaa. Jos käytät tätä komentoa funktiona, palautetaan seuraavasti:

- **1** = vasen nappula
- **2** = oikea nappula
- **3** = keskimmäinen nappula

KÄYTTÖ

```
WAITMOUSE  
tai  
WAITMOUSE ()
```

Katso myös: [WAITKEY](#)

ESIMERKKI

```
ClearMouse  
  
Print "Press mouse..."  
  
WaitMouse  
  
Print "OK, press again..."  
  
button=WaitMouse() 'wait and read  
  
Select button  
    Case 1  
        Print "LEFT BUTTON"  
    Case 2  
        Print "RIGHT BUTTON"  
    Case 3  
        Print "MIDDLE BUTTON"  
    Default  
        Print "SPECIAL BUTTON"  
End Select  
  
'wait a key and close the program  
Print "Press any key to quit..."  
WaitKey
```

WHILE

KUVAUS

Muodostaa silmukan, jota toistetaan niin kauan, kun alkuehto on TOSI. Ero **Repeat**-silmukkaan, on se, että ehto esitetään jo alussa While-sanan yhteydessä, joten silmukkaa ei välttämättä suoriteta ollenkaan, jos alkuehto on jo valmiiksi epätosi. Repeat-silmukka käydään aina vähintään yhden kerran läpi, koska ehtotarkastus on vasta silmukan jälkeen.

Normaalisti kaikissa ohjelmissa on jonkin sortin pääsilmukka, jonka sisällä kaikki pelin toiminnot päivitetään.

Silmukasta voidaan rikkoutua kesken kaiken pois komennolla **Exit**.

While silmukka päätetään komentoon WEND.

KÄYTTÖ

```
WHILE ehto
    'koodia
WEND
```

Katso myös: [TOINEN SELITYS](#)

ESIMERKKI

```
i=1

While Not KeyHit(cbkeyreturn)

    Print "Press RETURN to QUIT."

    'If 10 000 iterations have passed break free
    i=i+1
    If i>10000 Then Exit

Wend
```

WRAPANGLE

KUVAUS

Pitää kulman välillä 0-360. Esim. **-48 -> 312**.

Funktion hyöty on lähinnä siinä, että taulukko-sisältää-käännetyt-spritet-tekniikka ei lue taulukon yli.

KÄYTTÖ

WRAPANGLE (kulma)

- kulma** = Desimaaliluku tai kokonaisluku.

ESIMERKKI

Print WrapAngle(-48.0)

WaitKey

WRITE

KUVAUS

Kirjoittaa tekstiä ruudulle. Fontti pysyy aina samana. Komento kirjoittaa tekstin suoraan näytön etupuskuriin, joten se ilmestyy ruudulle välittömästi ilman DrawScreeniä. Huomaa, että toisin kuin Print-komennossa, tässä rivi ei vaihdu, joten voit kirjoittaa edellisen tekstin "perään".

Voit asettaa tekstin sijainnin käyttämällä **Locate**-komentoa.

Emme suosittele tämän komennon käyttämistä, koska se saattaa aiheuttaa ongelmia muun grafiikan kanssa. Käytä paljon mieluummin Textiä tai AddTextiä.

KÄYTTÖ

```
WRITE teksti
```

- **teksti** = Tulostettava kokonaisluku, desimaaliluku tai merkkijono.

Katso myös: [PRINT](#)

ESIMERKKI

```
'Set the printing coordinates
Print
Locate 100,80

'make a "loading" panel
For i=1 To 40

    Write "|"
    Wait 50

Next i

WaitKey
```

WRITEBYTE

KUVAUS

Kirjoittaa yhden tavun tiedostoon. Samalla tiedosto-osoitin siirtyy yhden askeleen eteenpäin.

Tavun arvo on kokonaisluku väliltä 0-255.

KÄYTTÖ

```
WRITEBYTE tiedostomuuttuja, arvo
```

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.
- **arvo** = Mielellään kokonaisluku väliltä 0-255.

Katso myös: [READBYTE](#), [WRITESHORT](#), [WRITEINT](#), [WRITEFLOAT](#), [WRITESTRING](#), [WRITELINE](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
  lifes=8

  WriteByte f,8
  WriteShort f,lifes*9
  WriteInt f, -8901
  WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

  mybyte=ReadByte(f)
  myshort=ReadShort(f)
  myint=ReadInt(f)
  myfloat=ReadFloat(f)

  Text 10,10,mybyte
  Text 10,30,myshort
  Text 10,50,myint
  Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```


WRITEFLOAT

KUVAUS

Kirjoittaa tiedostoon 32-bittisen desimaaliluvun. Samalla tiedosto-osoitin siirtyy neljä askelta eteenpäin.

KÄYTTÖ

```
WRITEFLOAT tiedostomuuttuja, arvo
```

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.
- **arvo** = Mielellään desimaaliluku.

Katso myös: [READFLOAT](#), [WRITEBYTE](#), [WRITESHORT](#), [WRITEINT](#), [WRITESTRING](#), [WRITELINE](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
  lifes=8

  WriteByte f,8
  WriteShort f,lifes*9
  WriteInt f, -8901
  WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

  mybyte=ReadByte(f)
  myshort=ReadShort(f)
  myint=ReadInt(f)
  myfloat=ReadFloat(f)

  Text 10,10,mybyte
  Text 10,30,myshort
  Text 10,50,myint
  Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```

WRITEINT

KUVAUS

Kirjoittaa tiedostoon 32-bittisen kokonaisluvun. Samalla tiedosto-osoitin siirtyy neljä askelta eteenpäin.

Kirjoitettu arvo on kokonaisluku väliltä -2 147 483 647 ja 2 147 483 647.

KÄYTTÖ

```
WRITEINT tiedostomuuttuja, arvo
```

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.
- **arvo** = Mielellään kokonaisluku.

Katso myös: [READINT](#), [WRITEBYTE](#), [WRITESHORT](#), [WRITEFLOAT](#), [WRITESTRING](#), [WRITELINE](#)

ESIMERKKI

```
x=200
y=150

AddText "Use arrows to move"
AddText "F1=save game"
AddText "F2=load game"

Repeat

    If KeyHit(cbkeyf1) Then GoSub savegame
    If KeyHit(cbkeyf2) Then GoSub loadgame

    If LeftKey() Then x-1
    If RightKey() Then x+1
    If UpKey() Then y-1
    If DownKey() Then y+1

    Box x,y,10,10

    DrawScreen

Until EscapeKey()
End

'This sub-program will save game
savegame:

    f=OpenToWrite("Media\simpleSave.dat")

    WriteInt f,x
    WriteInt f,y

    CloseFile f

Return

'This sub-program will load game
loadgame:

    f=OpenToRead("Media\simpleSave.dat")

    x=ReadInt(f)
    y=ReadInt(f)

    CloseFile f

Return
```

WRITELINE

KUVAUS

Kirjoittaa tekstirivin tiedostoon. Käytä tätä komentoa luodaksesi selkokielisiä teksti- ja asetustiedostoja (.ini).

KÄYTTÖ

```
WRITELINE tiedostomuuttuja, jono
```

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.
- **jono** = Kirjoitettava merkkijono.

Katso myös: [READLINE](#)

ESIMERKKI

```
'WRITE FILE

var$="This is the second line"

line1$="Two lines"
line2$="merged together"

f=OpenToWrite("Media\textfile.txt")
    WriteLine f,"Hello, this is the first line"
    WriteLine f,var$
    WriteLine f,1+2+3+4+5+6
    WriteLine f,line1$+" "+line2$
CloseFile f

'READ FILE

f=OpenToRead("Media\textfile.txt")
    For i=1 To 4
        Print ReadLine(f)
    Next i
CloseFile f

WaitKey
```

WRITESHORT

KUVAUS

Kirjoittaa tiedostoon 16-bittisen kokonaisluvun. Samalla tiedosto-osoitin siirtyy kaksi askelta eteenpäin.

Kirjoitettu arvo on kokonaisluku väliltä 0-65536.

KÄYTTÖ

```
WRITESHORT tiedostomuuttuja, arvo
```

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.
- **arvo** = Mielellään kokonaisluku väliltä 0-65536.

Katso myös: [READSHORT](#), [WRITEBYTE](#), [WRITEINT](#), [WRITEFLOAT](#), [WRITESTRING](#), [WRITELINE](#)

ESIMERKKI

```
'Create a file
f=OpenToWrite("Media\temp.dat")
  lifes=8

  WriteByte f,8
  WriteShort f,lifes*9
  WriteInt f, -8901
  WriteFloat f,6.0

CloseFile f

'Now read it
f=OpenToRead("Media\temp.dat")

  mybyte=ReadByte(f)
  myshort=ReadShort(f)
  myint=ReadInt(f)
  myfloat=ReadFloat(f)

  Text 10,10,mybyte
  Text 10,30,myshort
  Text 10,50,myint
  Text 10,70,myfloat

CloseFile f

DrawScreen

WaitKey
```

WRITESTRING

KUVAUS

Kirjoittaa tiedostoon merkkijonon.

VINKKI: Käytä mieluummin **WriteLine**-komentoa, jos haluat kirjoittaa "selkokielisiä" tekstitiedostoja.

KÄYTTÖ

```
WRITESTRING tiedostomuuttuja, arvo
```

- **tiedostomuuttuja** = Se muuttuja, johon tiedosto avattiin OpenToxxxx-komennolla.
- **arvo** = Mielellään merkkijono.

Katso myös: [READSTRING](#), [WRITELINE](#)

ESIMERKKI

```
'We recommend that you open the file into a text editor and  
'see the contents.  
  
f=OpenToWrite("Media\strings.dat")  
    WriteString f,"Don't you worry, babe. I'm an expert."  
    WriteString f,"Hastalavista, baby!"  
CloseFile f  
  
'Also try to do the same with WriteLine. Study the difference
```