

## Catepill#

### General Information

Author: Carsten Strotmann

Language: ACTION!

Compiler/Interpreter: ACTION / Bibo Assembler

Published: 14.06.2006

(c) 1990, Carsten Strotmann

unfinished Game with Level Editor

written in ACTION!

## Game Idea:#

Catapill is Sokoban on steroids: you drive the small caterpillar around in a huge warehouse and must complete missions

There are several good in the warehouse, and rules that must be observed:

- containers with base (L) and Acid (S) should never be stored side-by-side, else.....
- containers including magnetics (positive and negative available) should never stored next to iron boxes, else they stick together
- some containers contain explosives
- some contain hidden time-bombs, that are activated once the container is moved
- the caterpillar needs to refill gas from time to time
- the caterpillar can be damaged, so it is running slower, or can only turn in one direction
- there was also the idea to include some ideas from the boardgame "roboraally" in this game

## How to use#

boot attached Disk, on the DOS prompt, start "CTG2.COM" for a shortly playable version with Splash-Screen, or first load "CATAPILL.COM" and then "C.COM" for the latest binary of the game. start "CEDIT.COM" for the Level Editor.

```
*****
;**                                     **
;** Phoenix SoftCrew ACTION!          **
;** Programme und Tips f. 8Bit        **
;**                                     **
;** Carsten Strotmann                 **
;** An der Kreuzbreite 20             **
;**                                     **
;** D- 4410 Warendorf 1              **
;** (02581) 8920                      **
;**                                     **
*****

; Programmname:CATAPILL The Game
; Programmierer:PSC/Carsten Strotmann
; Filename:TG.ACT
; erste Version:02.07.90
; letzte Aenderung:19.03.93
; Zweck:
```

; Bemerkung:

;

;

INCLUDE "D:SYSTEM.ACT"

MODULE

BYTE sflg=\$03C6,  
    phase,  
    direc, ; Richtung Joystick  
    px=\$3DA,py=\$3DB, ; Playerposition  
    ax=\$3DC,ay=\$3DD, ; Absolute Position  
    dx=\$3DE,dy=\$3DF, ; Richtungen  
    hx=\$3E0,hy=\$3E1, ; Abweichung zum Zentr.  
    pp,sti,str,player,  
    consol=\$D01F

CARD hpixz=\$3CA,vpixz=\$3CC,  
    svscrol=\$3C0, shscrol=\$3C1,  
    plf=[\$2003], rtclok=\$12,  
    points,copadr=\$3C2

BYTE ARRAY raupe (\$100),  
    raupe1(\$100),  
    raupe2(\$100),  
    boom (\$100),  
    cols(3),  
    color(3)=\$3CE,  
    save1(\$21),  
    save2(\$21)

INCLUDE "D:TGINC.ACT"

PROC Count (BYTE xx,yy)

BYTE c

    c=Look (xx,yy)

    IF c#0 THEN

        FOR c=1 TO 20

            DO

                Sound (0,192-c,14,14-(C/2))

                Pause (1)

            OD

        SndRst ()

    FI

    Restaur (xx,yy,9)

RETURN

BYTE FUNC ChkLS (BYTE xx,yy,u)

BYTE res,z

    res=0

```

z=Look (xx+1,yy)
IF z+u=222 THEN
  res=1
FI
z=Look (xx-1,yy)
IF z+u=222 THEN
  res=1
FI
z=Look (xx,yy+1)
IF z+u=222 THEN
  res=1
FI
z=Look (xx,yy-1)
IF z+u=222 THEN
  res=1
FI

```

```
RETURN (res)
```

```
PROC BoomK (BYTE xx,yy)
```

```

Sound (0,6,4,10)
Restaur (xx,yy,93)
Restaur (xx+1,yy,93)
Restaur (xx+1,yy+1,93)
Restaur (xx+1,yy-1,93)
Restaur (xx-1,yy,93)
Restaur (xx-1,yy+1,93)
Restaur (xx-1,yy-1,93)
Restaur (xx,yy+1,93)
Restaur (xx,yy-1,93)

```

```

Pause (100)
SndRst ( )

```

```

ClearK (xx+1,yy)
ClearK (xx+1,yy+1)
ClearK (xx+1,yy-1)
ClearK (xx-1,yy)
ClearK (xx-1,yy+1)
ClearK (xx-1,yy-1)
ClearK (xx,yy+1)
ClearK (xx,yy-1)
ClearK (xx,yy)

```

```
RETURN
```

```
PROC MoveBox (BYTE xx,yy)
```

```
BYTE z,u,sx,sy,sc
```

```
z=-1
```

```
DO
```

```
z==+1
```

```
u=Look (xx,yy)
```

```
xx==+dx
```

```
yy==+dy
```

```
UNTIL u=0 OR u=1 OR u=9 OR u=21 OR u=13 OR u=17 OR u=25 OR z>3
```

OD

sc=u  
sx=xx-dx  
sy=yy-dy

xx== -dx  
yy== -dy

IF z<4 AND u=0 OR u=9 AND z>0 THEN

Sound (0,50,12,9)

FOR u=1 TO z

DO

xx== -dx

yy== -dy

MoveK (xx,yy,xx+dx,yy+dy)

OD

ClearK (xx,yy)

SndRst ( )

u=Look (xx+dx,yy+dy)

IF u=113 OR u=109 THEN

u=ChkLS (xx+dx,yy+dy,u)

IF u=1 THEN

BoomK (xx+dx,yy+dy)

FI

FI

FI

IF z>0 THEN

IF sc=9 THEN

Count (sx,sy)

FI

IF sc=21 AND dx=0 THEN

BLft (sx,sy-dy)

FI

IF sc=13 AND dy=0 THEN

BUp (sx-dx,sy)

FI

IF sc=17 AND dy=0 THEN

BDwn (sx-dx,sy)

FI

IF sc=25 AND dx=0 THEN

BRht (sx,sy-dy)

FI

FI

RETURN

PROC PosR ( )

BYTE U

CARD xx,yy

ax=0

ay=0

xx=0

yy=0

```

DO
  DO
    u=Look(xx,yy)
    xx==+1
  UNTIL xx=40 OR u=29
  OD
  IF xx=40 THEN xx=0 FI
  yy==+1
  UNTIL yy=24 OR u=29
  OD

  ax=xx-1
  ay=yy-1

  xx==*8-hx
  yy==*16-hy

  IF xx>152 THEN
    px==+xx-152
    xx=152
  FI

  IF yy>160 THEN
    py==+yy-160+1
    yy=160
  FI

  DO

    IF xx>0 THEN
      sflg==%4
      xx==-1
      vpixz==+1
    FI

    IF yy>0 THEN
      sflg==%1
      yy==-1
      hpixz==+1
    FI

    DO
      UNTIL sflg=0
    OD

  UNTIL xx=0 AND yy=0
  OD

RETURN

PROC Blend ()

BYTE u,c

color(0)=cols(0)&$F0
color(1)=cols(1)&$F0
color(2)=cols(2)&$F0

```

```

FOR u=0 TO $F
DO
  FOR c=0 to 3
  DO
    IF color(c)<cols(c) THEN
      color(c)==+1
    FI
  OD
  Pause (3)
OD

RETURN

PROC Change ()

BYTE c

  P_Clear(2)
  P_Clear(3)
  FOR c=1 TO 20
  DO
    Sound (0,100+c,12,C/2)
    Pause (1)
  OD

  Restaur (ax,ay,29)

  IF player = 0 THEN
    MoveBlock (save1,$3C0,$21)
    MoveBlock ($3C0,save2,$21)
  ELSE
    MoveBlock (save2,$3C0,$21)
    MoveBlock ($3C0,save1,$21)
  FI

  ClearK (ax,ay)

  player == ! 1
  ARaupe ()
  SndRst ()
  Pause (2)

RETURN

PROC ShowTime ()

BYTE hpos1=$3D7, hpos=$D000,
      t1=$12, t2=$13, xv, yv
CARD pmadr=$2D5, adr

IF t2>$1 OR hpos1=0 THEN

  t2=0
  t1==+1

  IF t1>19 THEN
    t1=0
    Change ()
  FI

```

```

IF t1=10 THEN Change ( ) FI

IF t1<5 OR t1>14 THEN
  yv=0
ELSE
  yv=$B
FI

IF t1<10 THEN
  xv=7
ELSE
  xv=0
FI

adr=pmadr+$11D+yv
Zero (pmadr+$11D,$18)
sflg=$10
DO UNTIL sflg=0 OD
hpos1=$BE+xv
hpos=hpos1
sflg=$10
DO UNTIL sflg=0 OD
MoveBlock (adr,timpl+$C*t1,$C)
FI

RETURN

PROC BoomBox (BYTE xx,yy)

BYTE u,x,z
BYTE ARRAY hpos=$3D2

  z=0
  FOR u=0 TO 7
  DO
    FOR x=0 TO 10
    DO
      Sound (0,z,0,15)
      z==+1
    OD
    Animate (0,px+(dx*8),py+(dy*16),u,boom)
    IF u=4 THEN ClearK (xx,yy) FI
    Pause (2)
  OD
  SndRst ( )

RETURN

PROC MainInit ( )

BYTE chsalt=$26B,
  chbas =$2F4,
  dmactl=$22F,
  nmien=$D40E,chr,
  crsinh=$2F0

CARD savmsc=$58

```

```
BYTE ARRAY file (20),scolor=$2C4

hx=$20
hy=$40

px=47+hx
py=61+hy

PM_Init ()
MPA_Set ()

chsalt=Set_Ramtop (8)
dmactl=0
Font_Load ("D1:CATAPILL.FNT",chsalt)
nmien==%$C0

chbas=chsalt+4
Font_Load ("D1:TOPLINE.FNT",chbas)

Close (1)
Open (1,"D1:TOPLINE.SCR")
BGet (1,savmsc,160)
Close (1)

SCopy (file,"D1:LEVELDAT.SCR")
Screen_Load (file)

MPA_Load (raupe1,"D:RAUPE.MPA")
MPA_Load (raupe2,"D:RAUPE2.MPA")
MPA_Load (boom,"D:BOOM.MPA")
MoveBlock (raupe,raupe1,$100)

dmactl=34
PM_Set ()
PM_Col (2,0,6)
PM_Col (3,2,10)
PM_Col (0,0,7)
PM_Col (1,0,12)

crsinh=1
Dspl ()
Init ()

scolor(0)=$C4
scolor(1)=$1A
scolor(2)=$86
scolor(4)=$0

points=0

sflg=$F0
DO
UNTIL sflg=0
OD

dx=1
dy=0
direc=2
phase=0
```



```

Blend ( )
PosR ( )
ClearK (ax,ay)
Dreh(0)

player=0
MoveBlock (save2,$3C0,$21)

RETURN

PROC Main ( )

  BYTE st,chr

  MainInit ( )

  ARaupe ( )
  rtclok=18

  DO

    ShowTime ( )
    st=Stick(player)!$F
    str=Strig(player)
    dx=0
    dy=0

    IF st=1 THEN sti=0 dy=-1      ; OBEN
    ELSEIF st=2 THEN sti=1 dy=1  ; UNTEN
    ELSEIF st=4 THEN sti=2 dx=-1 ; LINKS
    ELSEIF st=8 THEN sti=3 dx=1  ; RECHTS
    FI

    chr=Look (ax+dx,ay+dy)
;   PRINTB(CHR)

    IF sti=dirac THEN
      IF chr=0 OR (chr>12 AND chr<26) THEN
        IF st=1 THEN
          Up ( )
        FI
        IF st=2 THEN
          Down ( )
        FI
        IF st=4 THEN
          Left ( )
        FI
        IF st=8 THEN
          Right ( )
        FI
      ELSEIF chr#29 THEN
        IF str THEN
          MoveBox (ax+dx,ay+dy)
        ELSEIF chr>25 THEN
          BoomBox (ax+dx,ay+dy)
        FI
      FI
    FI
  
```

```
ELSEIF st#0 THEN
  Dreh (sti)
  Pause (5)
FI
OD
```

```
RETURN
```