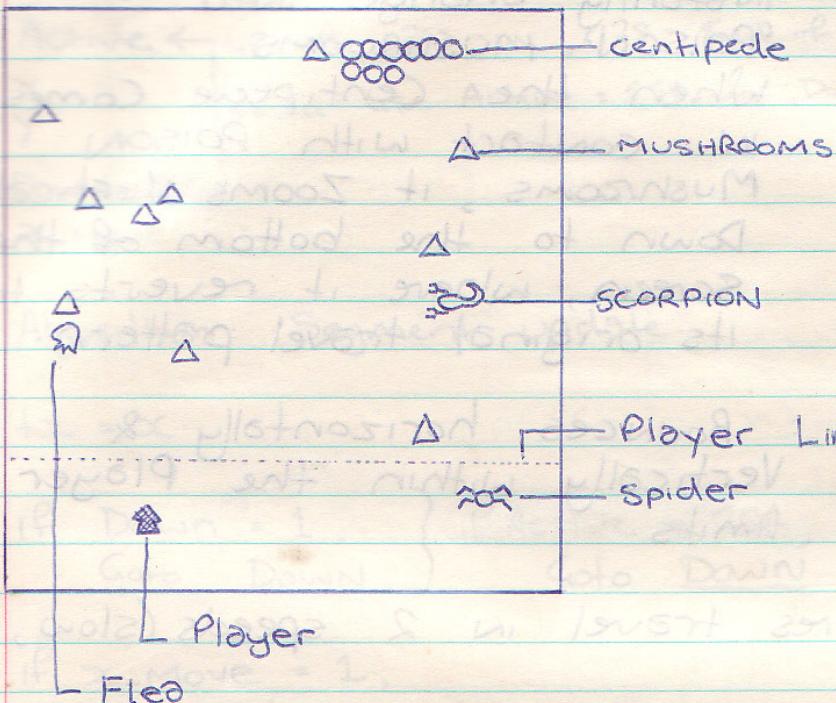


ARCADE - Type Game Clones:

(i) Centipede:



Movement of Objects:

Centipede : Moves horizontally until it either (a) Reaches the Edges of the Screen, or (b) Mushroom Where it Descends Down 1 Line, until it Reaches the Bottom, where it Returns Back to the top of the Screen.

Flea : Drops Vertically, Randomly leaving Mushrooms behind.

SCORPION : Travels Horizontally across the Screen. Any Mushrooms it Comes in Contact with Instantly Change into 'POISON' MUSHROOMS When the Centipede Comes in contact with POISON Mushrooms, it Zooms Vertically Down to the bottom of the Screen, where it reverts to its original travel pattern.

SPIDER : Bounces horizontally & Vertically within the Player limits.

All creatures travel in 2 speeds (slow, fast)

Speed of creatures go fast after certain point values.

Values:

- 1 Flea on screen at once.
- 1 Scorpion on SCREEN at once.
- 1 Spider on screen at once.
- 8 Centipedes on screen at once.

Player:

Can only move within limit.

Fire :

Maximum 1 Bullet on Screen.

Centipede Control:

Each Segment has following Data:

Active \leftarrow (0-off, 255-on, 128-End of Centipede)
 X
 Y could use Active = 254 for Down-Flg.
 Down-Flg
 X-Move.

Algorithm: Segment update.

$fx = x$

If $Down^{Flg} = 1$, } If Active = 254,
 Goto Down Goto Down

If $x_Move = 1$,

$x = x + 1$,

Graphic = Segment_Rt

If $x_Move = -1$

$x = x - 1$,

Graphic = Segment_Lt

If Screen(x,y) = MUSHROOM,

Goto Down

If Screen(x,y) = Poison Mushroom,

Down-Flg = 1, } Active = 254

Goto Down

If $x > 31$,
Goto Down

If $x < \emptyset$,
(Goto Down)

Exit.

Down

$x = tx$

$x_Move = -(x_Move)$

Graphic = Segment_Down,

$y = y + 1$

If $y = 2\emptyset$, } If $y = 2\emptyset$, }
DOWN-FIG = \emptyset } Active = 255

If $y > 23$,

$y = \emptyset$

If Screen ($x + x_Move$, y) = MUSHROOM/Poison,
 $x_Move = -(x_Move)$

Exit.

Algorithm : Centipede Control.

Pointer = First Segment

START

If Active (Pointer) = 128,
Exit

If Active (Pointer) = \emptyset ,
Goto NEXT

PRINT (x, y) = " "

CALL SEGMENT UPDATE

PRINT (x, y) = Graphic

] or, to eliminate any Flicker:

$Tx = x, Ty = y$

CALL SEGMENT UPDATE

PRINT (Tx, Ty) = " "

PRINT (x, y) = Graphic

Next

Pointer = Next-Segment

Goto START

Algorithm: Shot Segment Detection.

(Centipede + Creatures put to screen,
then players bullet (Ship is put first.))

Pointer = First Segment

Loop

IF Active (Pointer) = 128,
 Exit Down.

IF Active (Pointer) = \emptyset ,
 Goto NEXT

IF Screen (x,y) = Bullet,

 PRINT (x,y) = MUSHROOM,

 Active (Pointer) = \emptyset ,

 Player Bullet Active = \emptyset ,

 Score = Score + n points, (x, y) -> S

 Exit.

Next

Pointer = Next_Segment

Goto Loop

Note: Some Collision Technique is used
for all other Creatures.

Mushrooms are Detected in the Bullet
Control algorithm.

Appearance of Other Creatures:

Simplest method: use timing table:

Flea : Count, Max

Spider : Count, Max

Scorpion : Count, Max

Each Loop, INC Count
If = max, set Creature.(Co-ords / Active.)

Each Creature: only one on screen at a time.

Active - Ø-off, 255-on

x,y - Co-ordinates

each has own system for movement:

Flea: If Active = Ø, Exit.

Print (x,y) = "

If Mushroom (i.e. Random Value)] i.e Randomly
Print(x,y)mushroom.] Decide to leave
mushroom behind.

y = y + 1

If y > 23,

Active = Ø.

Print (x,y) = " @"

Exit.

Scorpion / Spider : Both have Δx -Move values.
(Spider Bounces, ∴ May Need y-move value.)

Scorpion : Add in $2/n$ Bytes to store Background
(Buffer)

Scorpion : If Active = \emptyset ,
Exit

Print $(x, y) = \text{Buffer}$ } or $Tx = x$

If x -Move = \emptyset ,

$x = x - 1$,
Graphic = Left-Face

If x -Move = 255,

$x = x + 1$

Graphic = Right-Face

If $x < \emptyset$,
Active = \emptyset , Exit

If $x > 31$,

Active = \emptyset , Exit

Read $(x, y) = \text{Buffer}$

Print $(x, y) = \text{Graphic}$.

If Buffer = Mushroom,

Change = Poison Mushroom.

Exit.

Spider Control:

if Active = \emptyset , Exit.

$tx = x$

$ty = y$

Print $(tx, ty) = " "$

if $x_move = \emptyset$

$x = x - 1$

if $x_move = 255$

$x = x + 1$

if $y_move = \emptyset$

$y = y - 1$

if $y_move = 255$

$y = y + 1$

if Screen $(x, y) = \text{mushroom}$,

$x = tx$

$y = ty$

if $x < 0$, Active = \emptyset , Exit

if $x > 31$, Active = \emptyset , Exit

if $y = 23$, $y_move = -(y_move)$

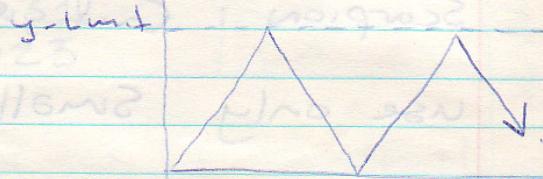
if $y = y_limit$, $y_move = -(y_move)$

Print $(x, y) = \text{Spider } " \wedge "$

Exit.

Note: Every spider appears from the opposite side of the screen as the last spider.

e.g. if last was from Right, the next appears from to Left.



(INSERT)

Speed Controller:

Each 4 creatures have timer (similar to Appearance table.)

Centipd : Count, max

Spider : Count, max

Flea : Count, max

Scorpion : Count, max

use only small #'s for max's.

i.e. Each Loop, inc count
if = max, call movement Routine

Difficulty:

Each Level Completed, the following happens :

- (a) INCREASE IN CENTIPEDES (Max 8 -
8 segments. Each max!)
Until maximum (stays at 8)
- (b) Speed increase (Starts at 2/4)
After ~ levels, Becomes & faster.
- (c) Appearance increases every ~
Levels.

Mushrooms:

Printed: $x \rightarrow 1 \rightarrow 3\phi$ (1 wide at each side)
 $y \rightarrow \phi \rightarrow 23$

Player has: $x \rightarrow 1 \rightarrow 3\phi$ } Also, spider
 $y \rightarrow 18 \rightarrow 23$

All creatures: $x \rightarrow \phi \rightarrow 31$
 $y \rightarrow \phi \rightarrow 23$

Graphics:

All sprites put in the following order:

Centipede PLAYER
Frogs Creatures
Spider Bullet.

Collision ~~as per~~:

Bullet move: If Active = ϕ , Exit

If Screen (x, y) \leq Mushroom / space, jump

Print at $x, y = " "$

$y = y - 1$

If Screen (x, y) = Mushroom / Poison.

Active = ϕ , Score = score + 1

Print (x, y) = " ", Exit

If $y < \phi$, Active = ϕ , Exit

Print at $x, y = "\uparrow"$

Exit.

When Press Fire:

If Active $\neq \emptyset$, Exit
 Active \neq $x = \text{Player } x - 1$
 $y = \text{Player } y - 1$

If Screen(x,y) = Mushroom/Poison Mushroom
 Score = Score + 1
 Print(x,y) = "
 Exit.

Print(x,y) = " \uparrow "
 Active = 1
 Exit.

Each Loop:
 If Active = \emptyset , Exit
 Print at(x,y) = "
 $y = y - 1$

If Screen(x,y) = Mushroom/Psn
 Print(x,y) = " \uparrow "
 Active = 0, Score = Score + 1
 Print(x,y) = "
 Exit

When press Left/Right/up/Down:

$tx = x$
 $ty = y$

Print(x,y) = " " } ALSO CHANGE
 (i.e. Print tx,ty)

If Left, $x = x - 1$

If Right, $x = x + 1$

If Up, $y = y - 1$

If Down, $y = y + 1$

If Sc

If Screen(x,y) = Mushroom/Poison Mushroom

$x = tx$, $y = ty$

If $x < 1$, $x = 1$

If $x > 30$, $x = 30$

If $y > 23$, $y = 23$

If $y < 18$, $y = 18$
If screen(x, y) \neq Blank, Goto Explode Player.
Exit: Print at (x, y) = "X"

Exit.

Explode Player: Lose Life/etc...

Scoring:

Centipede: 10 pts per Segment

Flea : 15 points

Spider : 50 points

Scorpion : 20 Points

Mushroom : 1 points

Attributes: Centipede: Yellow ink.

Flea : Red

Spider : White (Bright)

Scorpion : Purple

Mushroom : Cyan / Green

Player : White

Background: Swaps : Black / Blue

Note: Could change colours per level.

(i.e. have table of Attr's & swap
Randomly / Set.)

Game over: When player lives = 0

Level Complete: When all Centipedes Destroyed

When Level is Complete, Add \approx 5 New mush
rooms, Swap Colours

Complete Game algorithm:

Cls

Start: Dump Random Mushrooms
Wait (Print "Get Ready!")

Loop: Dump Player

Dump Centipede

Dump Flea

Dump Spider

Dump Scorpion

Dump Bullet

Check Collision Player

Check Collision Bullet/creatures

update Score.

JR Loop.

Stats: Score

Lives

Level

High Score

SCORE: Δ LVL HIGH :
i.e. 000000 0 0 000000

As For Code:

Initialise

Loop: Dump Screen (i.e. Mushrooms)

Loop: Print 'Get Ready!' - Timer

Game: Call Appearance Controller

Call player move

Call Creatures Dump (i.e. Speed, Print, etc...)

Call Bullets Dump

Goto Game.

Other Routines: Titles, Game over, Menu (i.e. Options)
Music + Fx, High Score table.

Initialisation:

2 Tables for Centipede : One = Temporary.
other : Set Table

(for initialisation)

Also, All other starting variables pre-stored.
& copied.

Updating Centipede (i.e. adding in Extra's)

Centipede stored in sets of 8 segments

for 1 Centipede : Set Active (9th) = 128
2 etc... : " " (18th) = 128

Other Flags/etc...

Segment_Max, Segment_Cnt = constant 2500
Each Centipede Segment Hit
INC Segment Cnt
if Segment_Cnt = Segment_Max,
Complete level.

Possible Sound Fx:

Centipede move (click)

Spider move (INC/Dec tone) (i.e. Via fence)

Scorpion ♂ (Click.)

Flea ♂ (Ascending Tone)

Fire (zip.) / HIT (Pop.)

Player HIT (Explosion (+ Animation))

(Note: too many will decrease speed!)
(unless : interrupt driven!)

OVERPM: ld A, (ix + 1) : - Read x ~~MEMORE~~ to pinball
 AND 224 : - Is x in Range $\emptyset \rightarrow 31$?
 RET Z ~~exit~~ - Exit if F true

DOWN: ld A, (Tx)
 ld (ix + 1), A : - Reset x
 ld A, (ix + 3) }
 CPL } Reverse x-Move
 INC A }
 ld (ix + 3), A : -
 ld HL, Graph_DN : Set Graphic for Segment_Down
 ld (Graph), HL : Set Graphic for Segment_Down

ld A, (ix + 2)
 INC A : - Y = Y + 1
 CP 24 : - Is Y > 23
 JR NZ, NEXT : - No, SKIP
 XOR A : - Else Y = \emptyset
 NEXT: ld (ix + 2), A
 CP 20 : - Is Y = 20
 JR NZ, CHK_N : - No, Skip
 ld A, ZSS
 ld (ix + \emptyset), A : - Reset Active to Not_Down

CHK_N: ld A, (ix + 3)
 ld C, A
 ld A, (ix + 1)
 Add A, C
 CALL CHKSC : - Check x + X-Move
 CP 'MUSHROOM' : - Is it Mushroom
 JR Z, RESET : - Yes, Jump
 CP 'Poison_Mushroom'

RET NZ

- Exit if Not Poison Mushroom

RESET: ld A,(ix+3)

CPL

INC A

ld (ix+3),A

RET

} Reverse X-MOVE

- Exit.

TX: Defb Ø

CHKSC: ld C,A

XOR A

ld B,A - BC = X

ld H,A

ld L,(ix+2) - HL = Y

Add HL,HL

Add HL,HL

Add HL,HL

Add HL,HL

Add HL,HL - HL = 32 * Y

ld DE,22528

Add HL,DE

Add HL,BC

- HL = Attribute Address at x,y

ld A,(HL)

- A = Value of Attribute

RET

- Exit