

Hi,

Just wanted to share a simple one level game that shows a simple way to create walls/borders in your 2d game. The game has only 1 level, but it shouldn't be too hard to add more. I want to make another version using sprites, but time will determine that (free time that is).

I created this game to show an easy way to create borders or walls for your 2D games in BASIC. This game only has one level, but it should not be too hard to create more. I want to create a version using sprites, since the sprites commands do not have an easy way to create walls for your games I am attempting to come up with something that will work for me. I'd like to share what I come up with. I know this is very basic, but it's just a starting point. With the sprite version I want to try to add a more fluent movement to the sprite.

Note: Now updated so you can use your arrow keys to move the player

```
'basic 2d game level wall/border example
'using native graphic commands
'programmed by Ben Jimenez
'use your arrow keys to move the player
'July 20,2011
'upadted Oct 15,2011
```

```
nomainwin
```

```
WindowWidth = 470
WindowHeight = 500
```

```
open "Border/Wall Example" for graphics_nsb_nf as #main
print #main, "fill white; flush"
print #main, "font ms_sans_serif 0 16"
#main, "up;goto 20 30;size 3;down"
```

```
dim pxy$(22,22)
```

```
up$= chr$_VK_UP)
lft$=chr$_VK_LEFT)
rht$=chr$_VK_RIGHT)
dwn$=chr$_VK_DOWN)
```

```
'start postion of first wall piece
x=10
y=20
```

```
'draw border walls
  for r=1 to 22 'first row
    for c=1 to 22 'current row column loop
      read d$
      pxy$(r,c)=d$
      if d$="B" then 'if border piece then draw it
        #main,"up;goto ";x+20;" ";y+20;
        ";down;backcolor blue;color blue"
        #main,"boxfilled ";x;" ";y

      end if

      if d$="s" then 'save player start position
        px=x 'set current pen x position to player start x
        py=y ' set current pen y position to player start y
        ar=r 'remember current row location
        ac=c 'remember current column locaton
      end if

      x=x+20 'increase column x location by 20
    next c
    x=10
    y=y+20 'increase row y location by 20
  next r

  #main,"flush" 'save drawn board as segment

  'draw player
  #main,"up;goto ";px+10;" ";py+10;
  ";down;backcolor brown;color brown"
  #main,"circlefilled 6"

  #main,"when characterInput [checkkey]"

[main.inputLoop] 'wait here for input event
#main,"setfocus"
  wait

[checkkey]

key$ = left$(Inkey$, 2)
```

```
select case right$(key$,1)

'if wall/boarder found for next movement then no movement will be done
.

case dwn$ 'down

    if pxy$(ar+1,ac)<>"B" then
'if next location is not a wall/border then move player
        #main,"up;goto ";px+10;" ";py+10;
";down;backcolor white;color white"
        #main,"circlefilled 6"

        py=py+20
        #main,"up;goto ";px+10;" ";py+10;
";down;backcolor brown;color brown"
        #main,"circlefilled 6"

        ar=ar+1

    end if

case rht$

    if pxy$(ar,ac+1)<>"B" then
'if next location is not a wall/border then move player
        #main,"up;goto ";px+10;" ";py+10;
";down;backcolor white;color white"
        #main,"circlefilled 6"
        px=px+20
        #main,"up;goto ";px+10;" ";py+10;
";down;backcolor brown;color brown"
        #main,"circlefilled 6"
        ac=ac+1

    end if

case lft$

    if pxy$(ar,ac-1)<>"B" then
```

```
'if next location is not a wall/border then move player
    #main,"up;goto ";px+10;" ";py+10;
";down;backcolor white;color white"
    #main,"circlefilled 6"
    px=px-20
    #main,"up;goto ";px+10;" ";py+10;
";down;backcolor brown;color brown"
    #main,"circlefilled 6"
    ac=ac-1

    end if

case up$
    if pxy$(ar-1,ac)<>"B" then
'if next location is not a wall/border then move player
        #main,"up;goto ";px+10;" ";py+10;
";down;backcolor white;color white"
        #main,"circlefilled 6"
        py=py-20
        #main,"up;goto ";px+10;" ";py+10;
";down;backcolor brown;color brown"
        #main,"circlefilled 6"
        ar=ar-1

    end if

end select
#main,"flush"
#main,"segment n"
#main,"delsegment ";n-2
if pxy$(ar,ac)="e" then
notice "You escaped the maze!"
end if
goto [main.inputLoop]

'maze data
data  B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B
data  B,s,B,x,x,x,B,x,B,x,x,x,x,x,B,x,x,x,x,x,x,B
data  B,x,B,x,B,x,B,x,B,x,B,B,B,x,B,x,B,x,B,B,B
data  B,x,B,x,B,x,B,x,B,x,B,x,B,x,B,x,B,x,x,e
data  B,x,x,x,B,x,B,x,x,x,B,x,B,x,B,x,B,x,B,x,B
data  B,x,B,x,B,x,B,x,B,x,B,x,x,x,B,x,B,x,x,B
```

data B,x,B,x,B,x,x,x,B,x,B,x,B,B,B,B,x,B,B,x,B
data B,x,B,x,B,x,B,B,B,x,B,x,x,B,x,x,x,x,B,B,x,B
data B,x,B,x,B,B,B,x,B,x,B,B,x,B,x,B,B,B,x,x,B
data B,x,B,x,x,x,x,x,B,x,B,x,x,x,x,B,x,x,x,B,x,B
data B,x,B,B,B,x,B,x,B,x,B,B,B,B,x,B,B,B,x,B
data B,x,x,x,x,x,B,x,x,x,B,x,x,x,x,x,x,B,x,B,x,B
data B,x,B,x,B,x,B,B,x,B,B,B,B,B,B,B,x,B,x,B
data B,x,B,x,B,x,B,x,x,B,x,B,x,x,x,x,B,x,x,B,x,B
data B,x,B,x,B,B,B,B,B,x,B,B,x,B,x,B,x,B,x,B,B
data B,x,B,x,x,x,B,x,x,x,x,x,B,x,B,x,x,x,B,x,x,B
data B,x,B,B,B,B,B,B,B,B,x,B,x,B,B,B,B,B,x,B
data B,x,B,x,x,x,B,x,x,x,B,x,B,x,x,x,B,x,x,x,x,B
data B,x,x,x,B,x,x,x,B,x,x,x,B,x,B,x,x,x,B,x,B,B
data B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B,B