

VIC BRICK 80

by S. "Italianlawyer" Gambino

for 10liners basic challenge - cat. PUR80

1. Yeah, another brick game!

Yes, as you read this paper you may ask why another brick game.

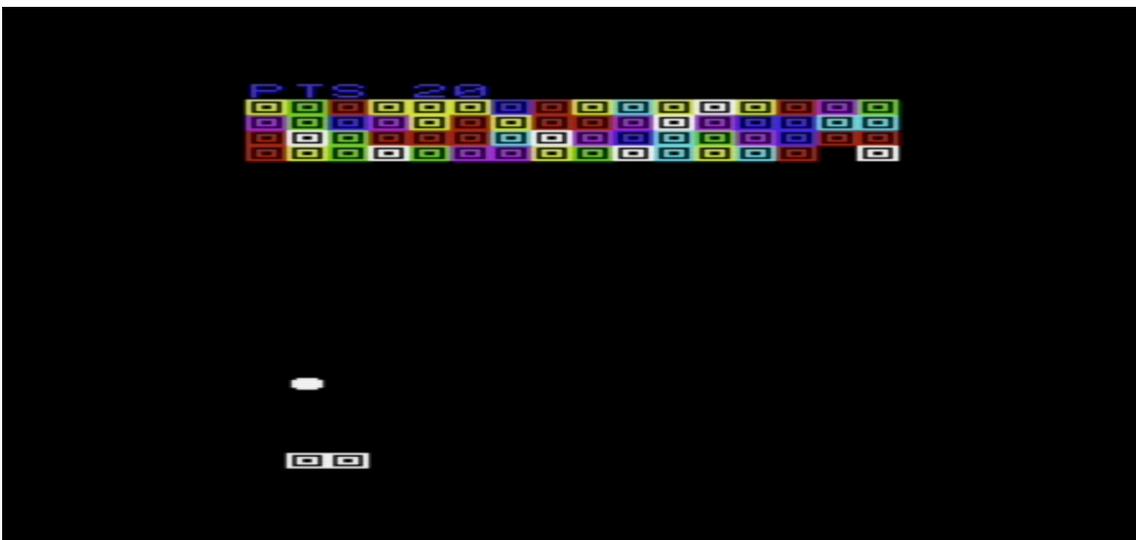
I could reply that is simple to make a brick game, less simple to fit it in just 10 lines of basic code, especially for me! (Doh!)

In particular, it's difficult to make it enough fast to give an arcade impression since BASIC is so slooow :(so I though for a little trick to be explained in part 3.

2. how to play

I really need to tell you how to play this one? ;P

You just need to move the racket using the "a" or the "d" keys in order to bounce the ball. The ball will brake some coloured bricks.



The ball usually bounce forward, BUT there's also a spike effect if you catch the ball with the corner. In this case the ball will bounce back.

If you don't catch the ball is game over, it's just one life. That's all folks.

The gameplay is really simple, but in the future I could add some other features.

3. TECH Stuff

Here is the code with a brief explanation:

```
0pO36866,(pE(36866)aN128)or16:pO36867,(pE(36867)aN129)or(2*26):pO36864,18
1pO36865,35:pO52,28:pO56,28:cL:dA255,129,189,165,165,189,129,255
2fOI=7168to7679:pOI,pE(I+25600):nE:fOc=7168to7175:rEa:pOc,a:nE:pO36869,255
3pO36879,8:s=8070:d=8071:p=7680:b=36876:y=10:x=10:dx=1:dy=1:pt=0:pO36878,15:?"{cl
ear}"
4fOz=.to63:pO7696+z,0:pO38415+z,int(rN(1)*7+1):nEz
5pp=p+x+16*y:pOs,32:pOd,32:gEa$:ifa$="a"tHifs>8064tHs=s-1:d=d-1:ifpp=dtHdx=-dx
6?"{up}pts";pt:ifa$="d"tHifs<8078tHs=s+1:d=d+1:ifpp=stHdx=-dx
7pOs,0:pOd,0:fOt=.to15:nE:ify<=1orpE(pp)<>32tHdy=-dy:pt=pt+10:pOb,230
8pOpp,209:fOt=.to15:nE:pOpp,32:ifx<=0orx>=15tHdx=-dx:pOb,220
9x=x+dx:y=y+dy:pOb,0:on-(y<25)gO5:?"{down}{down}{down}{left}{left}game
over":fOnn=.to500:nE:gO0
```

Line 0: This is the little trick I mentioned before. The Vic20 has just 506 characters on screen (23x22). It is already better than c64 because this need to move more characters on the screen having a 40x25 matrix in text mode, obviously if we do not consider hardware sprites. ;)

In this case the game window is further reduced to 16x26

characters using the 36866 and 36867 registry, in my opinion with some small improvements in speed.

Line 1 and 2 is the ordinary routine which allows you to redefine characters. The data includes the brick and racket.

Line 3 and 4 is the program variables and the random color wall generator.

Line 5 to 9 is the main loop.

Line 5 and 6 are used to control the racket. Two controls are included: the first is the field control and the second is the the abovementioned spike effect (the second if statement). The ball position is updated.

Line 7 and 8 racket and ball draw and some collision control. Please take note: if you are feeling the ball and or racket is flickering too much you could modify the values in the for cycle. I used 15 but you could increase to 25 for example. At 5 the game is fast, but I couldn't see nothing.

Line 9 updates x and y used to move the ball. I think it's the first time I used the On - goto statement instead of if -then (hahaha).

4. Greetings

The 10 liners is always a lot of fun, I'm glad to participate. A special greeting goes to the retroprogramming group Italy, without which I would never have known about this event. W l'Italia, abbasso il coronavirus!