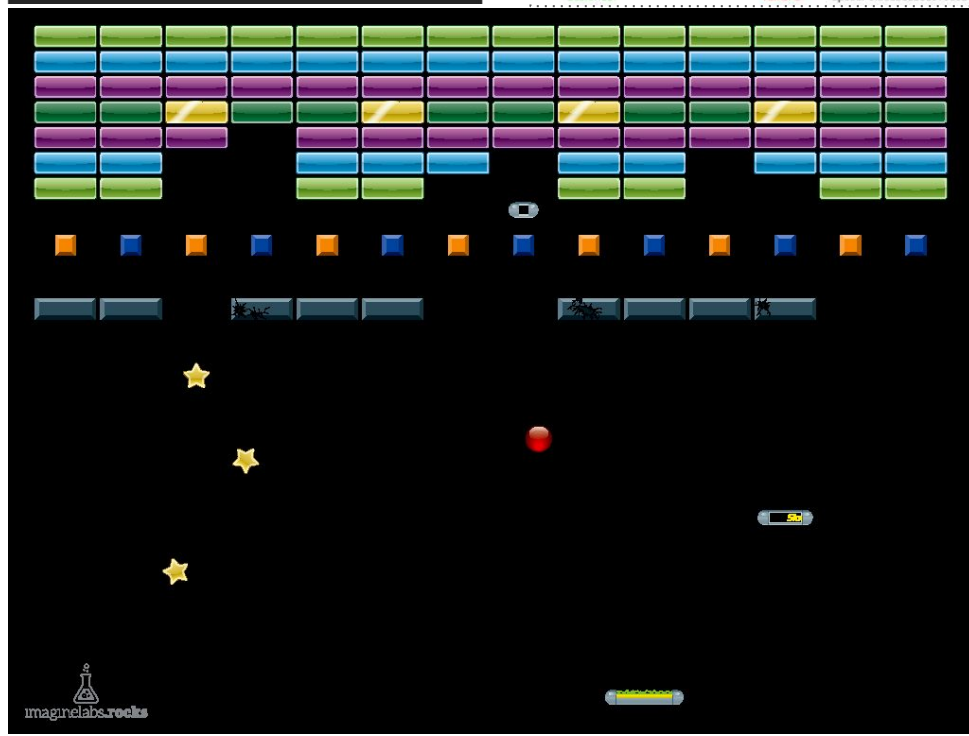
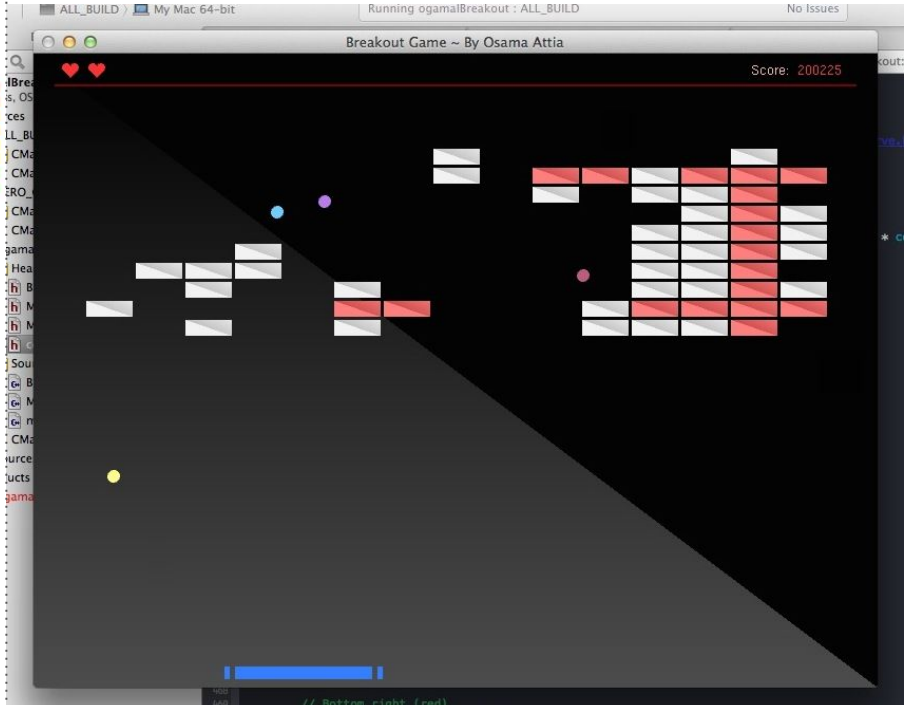
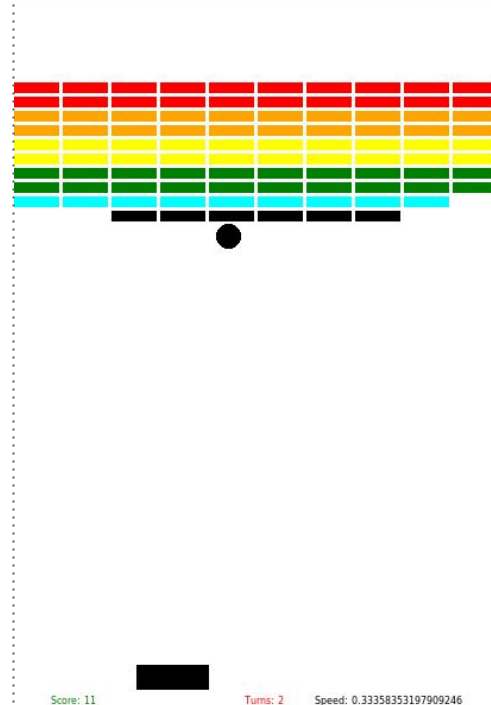
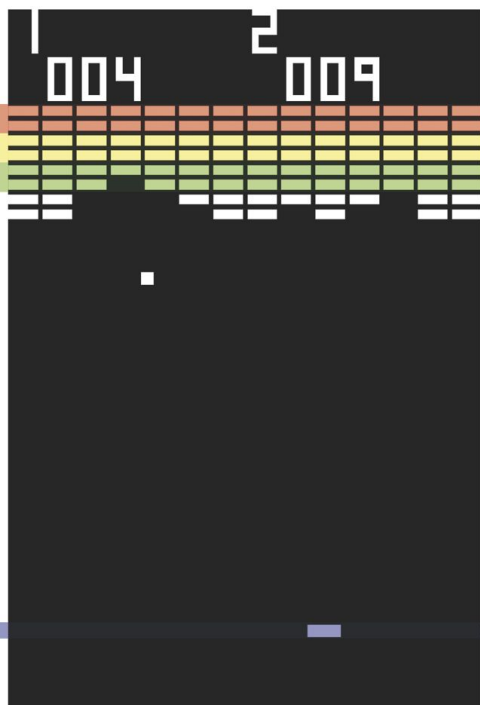


Score: 0

Turns: 3



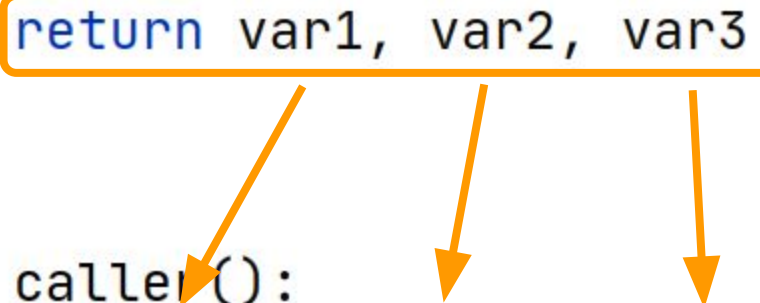


Plan


- multiple return values
- special bricks
- enhanced bouncing
- increasing speed
- cheats
- score
- lives
- paddle with varying bounce angle

```
def function():  
    var1 = 3  
    var2 = 5  
    var3 = 17
```

```
return var1, var2, var3
```

The return statement is enclosed in an orange rounded rectangle. Three orange arrows originate from the variables 'var1', 'var2', and 'var3' and point downwards to the arguments 'someval', 'otherval', and 'difval' in the caller function's assignment statement.

```
def caller():  
    someval, otherval, difval = function()
```

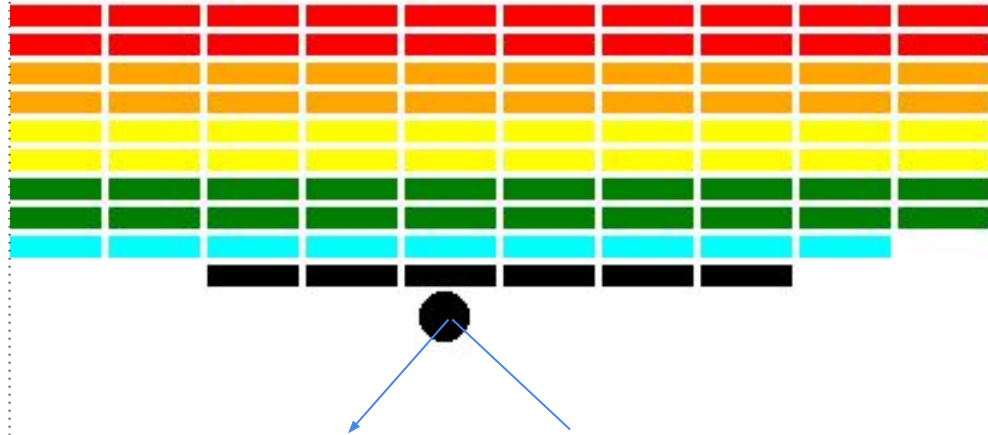
The caller function call is enclosed in an orange rounded rectangle. The return values from the function above are passed to the arguments 'someval', 'otherval', and 'difval' in the assignment statement.

- multiple return values
- special bricks
- enhanced bouncing
- increasing speed
- cheats
- score
- lives
- paddle with varying bounce angle

Special Bricks

- permanent bricks
- bricks with a counter
- game change (speed, angle, lives)
- game-changing “brick” anywhere

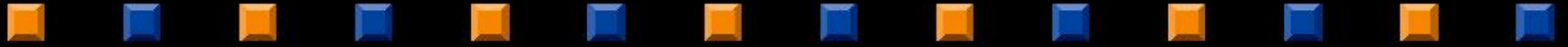
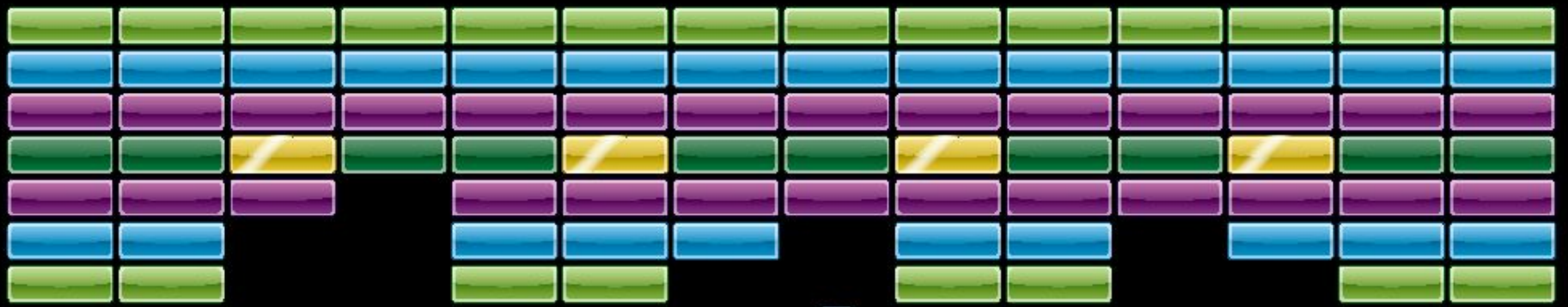
How to implement this?



Score: 11

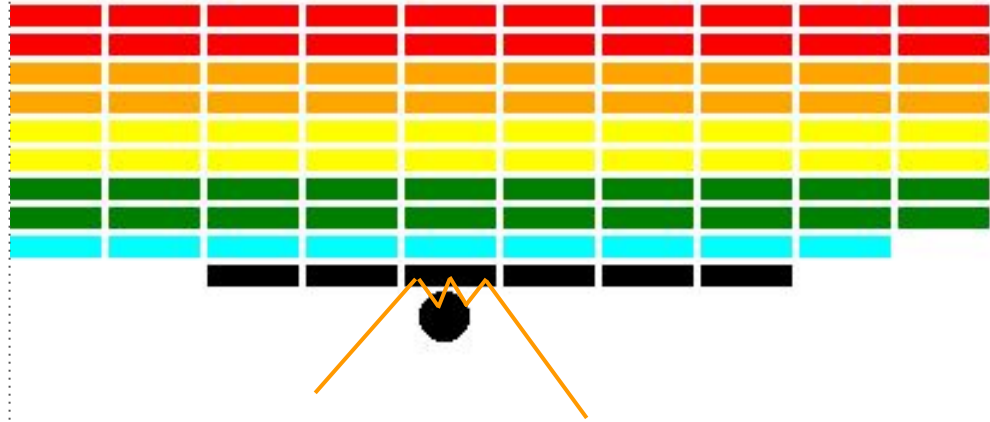
Turns: 2

Speed: 0.33358353197909246



Permanent Bricks

Sticky brick problem



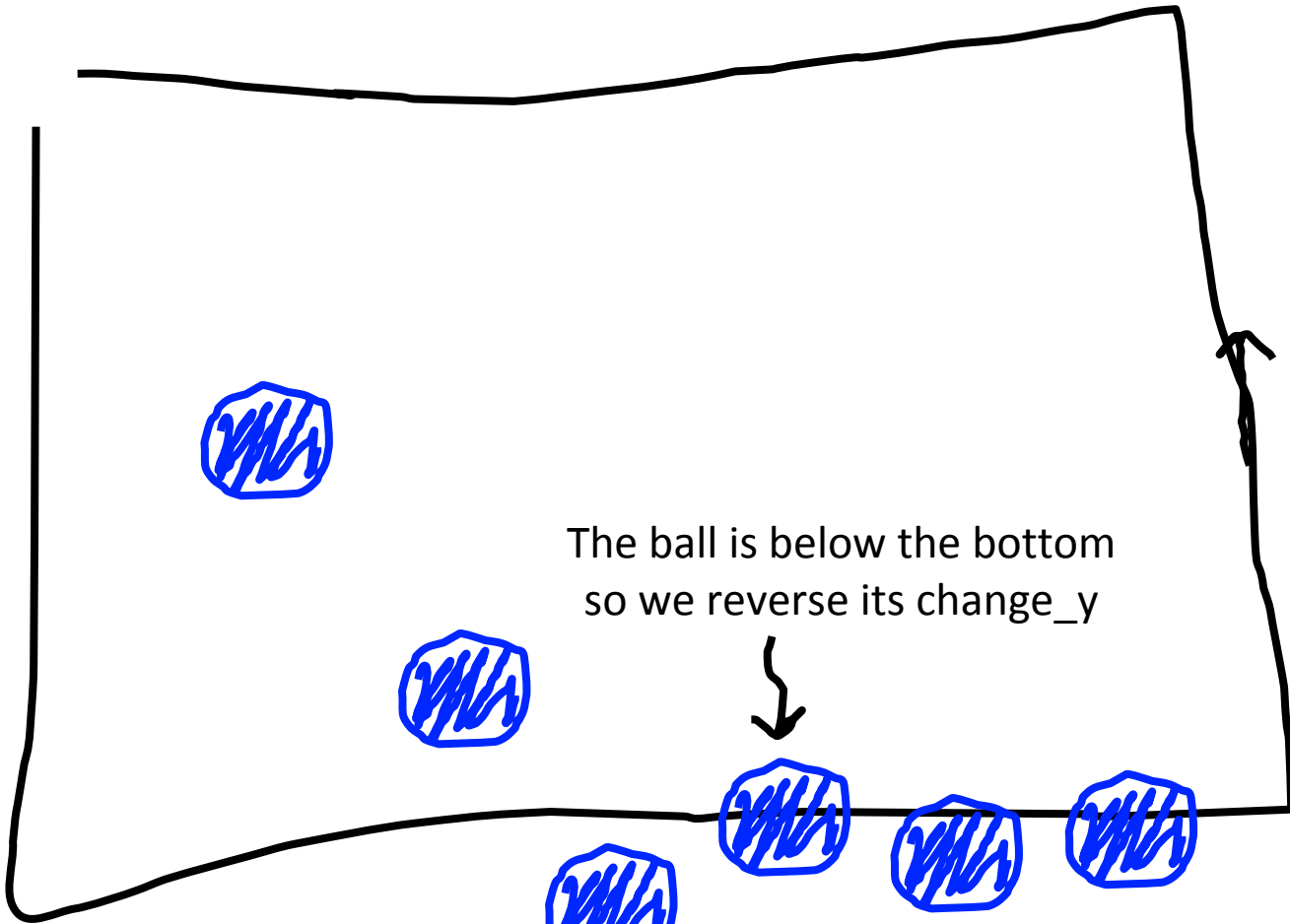
Score: 11



Turns: 2

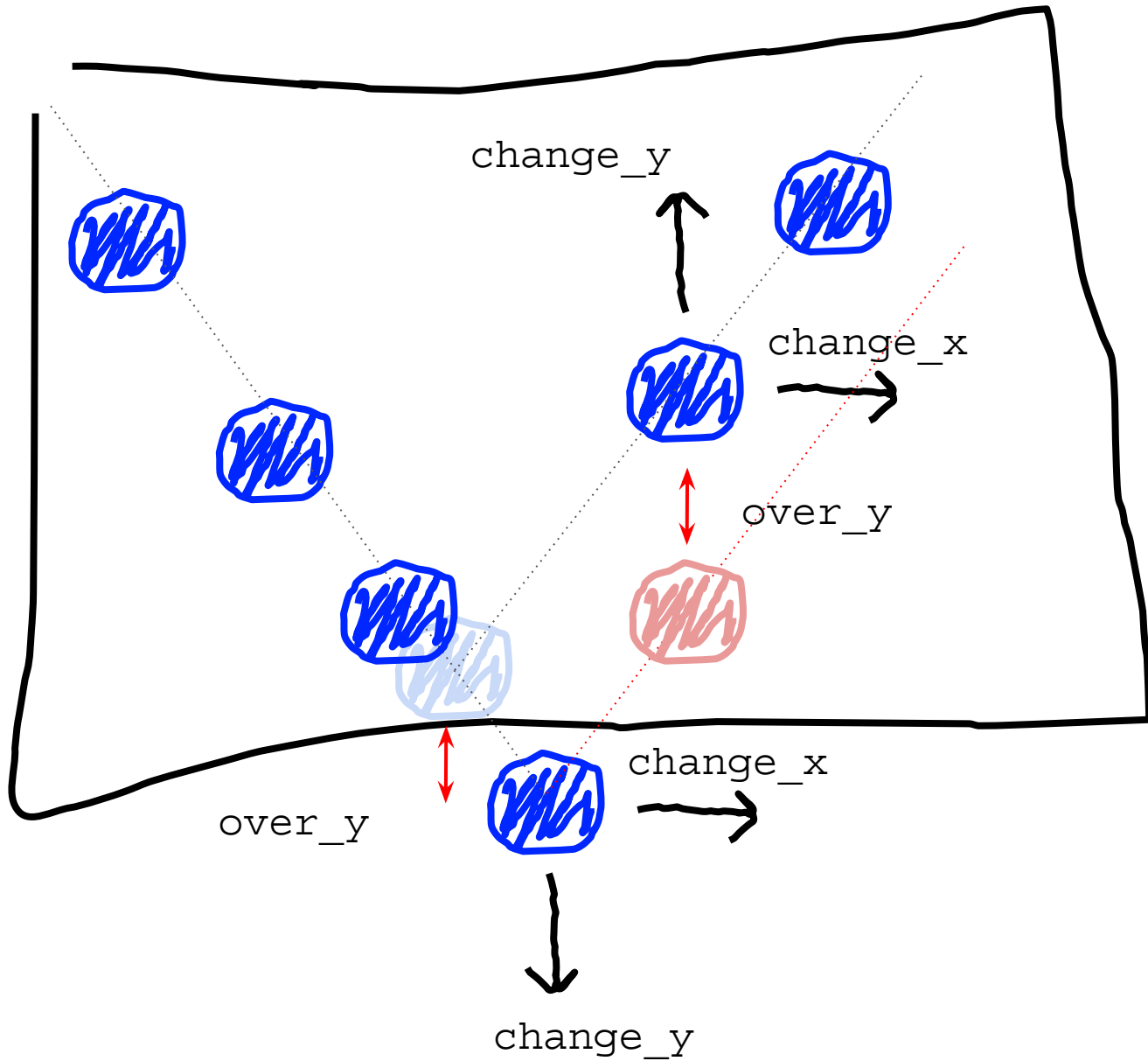
Speed: 0.33358353197909246

- multiple return values
- special bricks
- enhanced bouncing
- increasing speed
- cheats
- score
- lives
- paddle with varying bounce angle

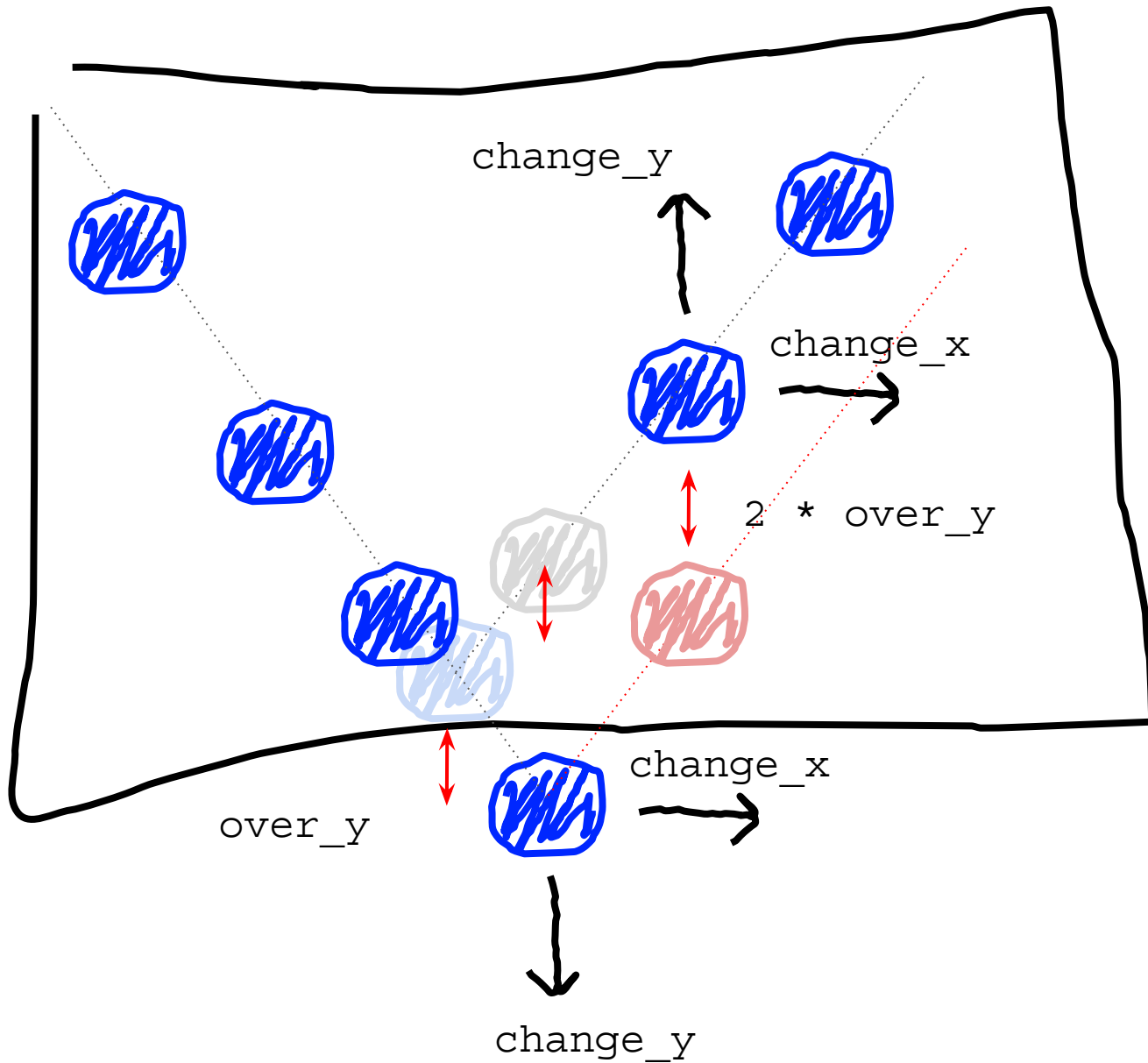


The ball is below the bottom
so we reverse its change_y

The ball is below the bottom
so we reverse its change_y



```
canvas.move(ball, change_x, change_y + 2 *  
over_y)
```



- multiple return values
- special bricks
- enhanced bouncing
- increasing speed
- cheats
- score
- lives
- paddle with varying bounce angle



Score: 0

Turns: 3

- multiple return values
- special bricks
- enhanced bouncing
- increasing speed
- cheats
- score
- lives
- paddle with varying bounce angle

- decrease the difficulty (speed)
- add turn
- infinite turns
- sticky paddle
- multiple balls or paddles

```
keys = canvas.get_new_key_presses()
for key in keys:
    if key.keysym == 'a':
        # do something
    elif key.keysym == 'z':
        # do something
```



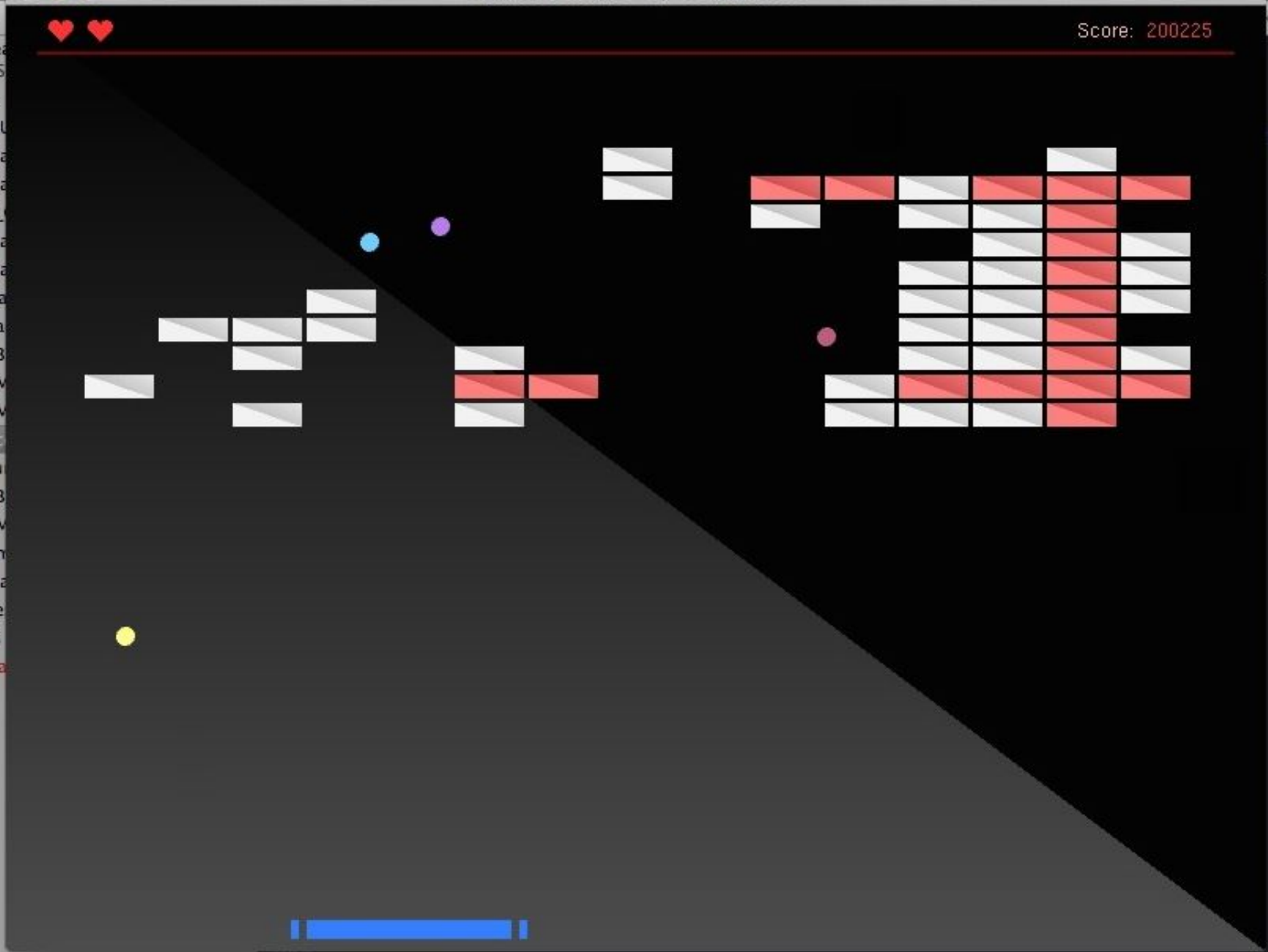
Score: 0

Turns: 3

Breakout Game ~ By Osama Attia



Score: 200225



// Bottom right (red)

ilBre
ts, OS
rces
LL_BU
CMA
CMA
ERO_
CMA
CMA
gama
Hea
h B
h M
h M
h
Sou
G+ B
G+ M
G+ n
CMA
ource
lucts
gama

kout:
vb.
* C

- multiple return values
- special bricks
- enhanced bouncing
- increasing speed
- cheats
- score
- lives
- paddle with varying bounce angle



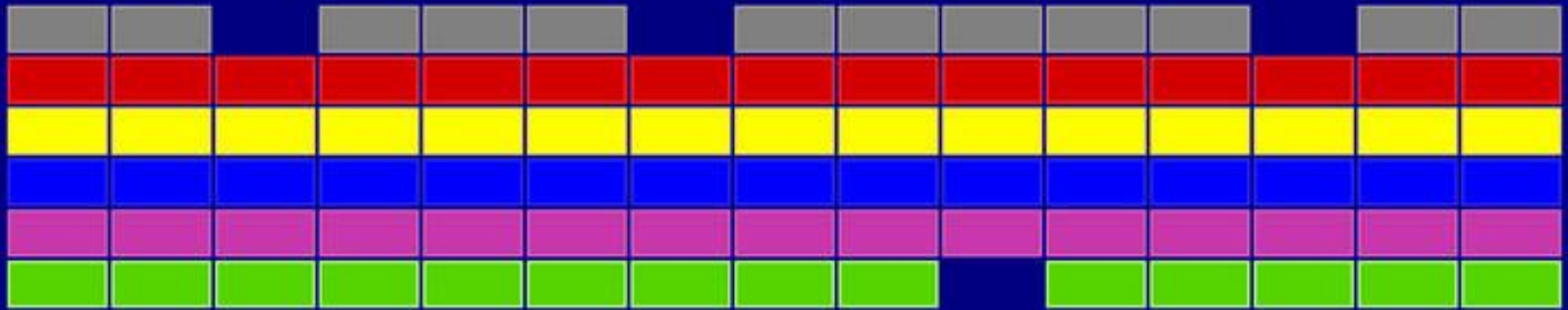
Score: 0

Turno: 3

- multiple return values
- special bricks
- enhanced bouncing
- increasing speed
- cheats
- score
- lives
- paddle with varying bounce angle

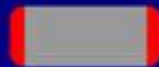


Level 5



200

dreamstime.



score:
12500



Score: 0

Turno: 3



#229660418