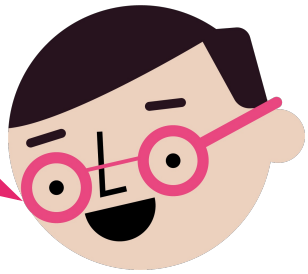
PYGAME ZERO 2 BUTTON CHALLENGE



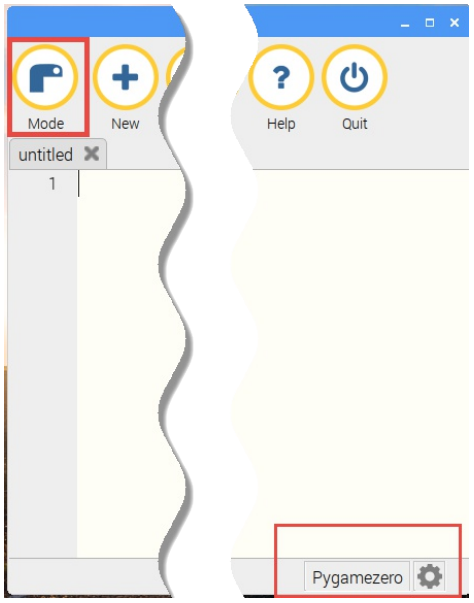
The Challenge...

The challenge is simple...

Create a game with PyGame Zero, but you may only use two buttons to play the game...

Our Editor

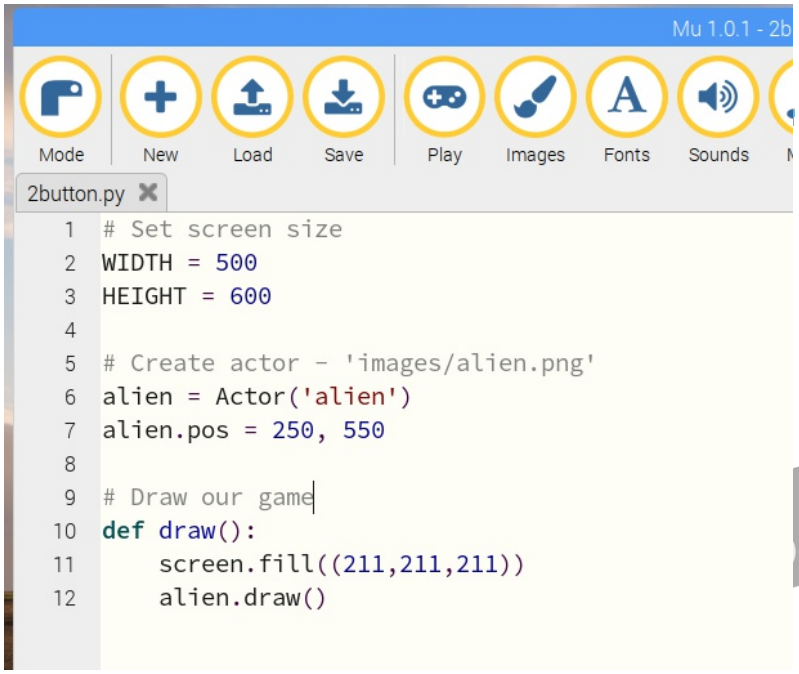
1. We will use the Mu Python editor to write our game. Not only is it a really nice editor; but it makes writing PyGame Zero so much easier! Open Mu from the programming menu and make sure that it is in "**Pygamezero**" mode.



Stage and Actors

2. First we need to create our screen and our actor. We will use the built-in "**alien**" actor for now; you can add your own characters later.

Type the following code, save it in the "**mu_code**" directory. Now run it by pressing the "**Play**" button. You should see an alien appear on a grey background.



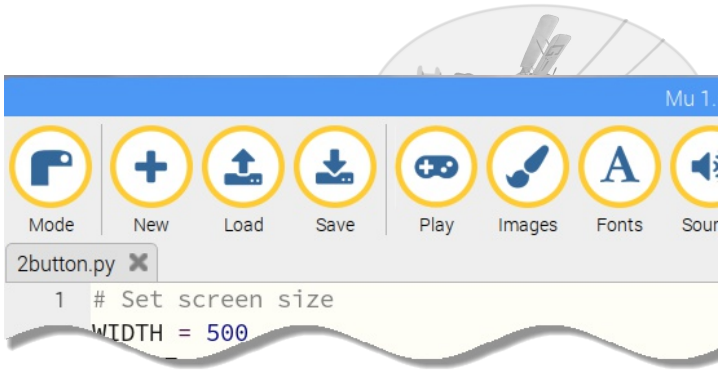

```
# Set screen size
WIDTH = 500
HEIGHT = 600

# Create actor - 'images/alien.png'
alien = Actor('alien')
alien.pos = 250, 550

# Draw our game
def draw():
    screen.fill((211,211,211))
    alien.draw()
```

Moving our Actor

3 Now we have an actor on our screen we need to make it move. Add the following two function definitions to the end of your file, save it and press play.



If PyGame Zero finds an update function, it executes this every time the game goes round its loop. We are telling the update function to call our checkKeys function. Hopefully you can see what this is doing!

```
12      screen.draw()
13
14  # Check for updates
15  def update():
16      checkKeys()
17
18  # checkKeys function
19  def checkKeys():
20      if keyboard.left:
21          if alien.x > 40:
22              alien.x -= 5
23      if keyboard.right:
24          if alien.right < 480:
25              alien.x += 5
```

Falling blocks!

4 We will now add a new actor to our screen; a block. Press "**Shift + Ctrl + F**" to open the file search tool. Search for "**blockerMad.png**" making sure you are looking in sub-directories too. Right click and copy the file you find. Now back in the Mu editor, click the "**Images**" button. In the window that opens, right click and paste your new actor file. You now need to rename the file so that it is all lower case, as PyGame Zero will not load a sprite with upper case letters!


```
9   alien = Actor('alien')
10  alien.pos = 250, 550
11  # Create actor - 'images/blockermad.png'
12  block = Actor('blockermad')
13  block.pos = (random.randint(20, WIDTH - 20), 10)
14
15  # Draw our game
16  def draw():
17      screen.fill((211,211,211))
18      alien.draw()
19      block.draw()
```

Adjust your program to match this and save it and play it. Each time you play the game our block should appear at a different point.

```
25  # checkKeys function
26  def checkKeys():
27      if keyboard.left:
28          if alien.x > 40:
```

Falling rocks!

5

Now we are going to make the block fall down the screen. To do this we will define a new function called move_block() and call it in the draw loop.

```
14
15  # Draw our game
16  def draw():
17      screen.fill((211,211,211))
18      alien.draw()
19      move_block() # move our block down each draw loop
20      block.draw()
21
```

So now our block drops down the screen and we can try and dodge it; however it would be nice for another block to appear once one drops off the bottom of the screen. It would also be good if the game ended if we were hit by a block...

```
25
26  # move_block function
27  def move_block():
28      block.pos = (block.x, block.y+1)
29
30  # checkKeys function
```

We will create a function called detect_dodge() which will handle all of this. We will also need to add a game_status variable to indicate when the game is over and adjust our draw loop slightly.

```
5   WIDTH
6   HEIGHT = 600
7
8   game_status = 1
9
10  # Create actor - 'images/alien.png'
```

```
16
17  # Draw our game
18  def draw():
19      if game_status == 1:
20          screen.fill((211, 211, 211))
21          alien.draw()
22          move_block()  # move our block down each draw loop
23          block.draw()
24          detect_dodge()
25
```

Over to you!

6

Make the game your own!

Can you add a score?

How about more objects?

How about random falling speeds?

Change the graphics...

```
32      block.pos = (block.x, block.y+1)
33
34  # detect_dodge function
35  def detect_dodge():
36      global game_status
37      if block.y >= HEIGHT - 10:
38          # block got to the bottom, draw another
39          block.pos = (random.randint(20, WIDTH - 20), 10)
40      if alien.collidepoint(block.pos):
41          # block has hit the alien
42          screen.clear()
43          screen.draw.text("Game Over!", (100, 250), color="red", fontsize=32)
44          game_status = 0
45
46  # checkKeys function
47  def checkKeys():
```