

# Tutors Only

## Extension: Play Again!

Making a play again button for on our game over screen

### Task 1.1: The button

Let's make a button for the player to click!

1. Make a rectangle variable and set it to a height, width and position where you want it to be.
2. In your draw function, draw the rectangle with whatever colour you want it to be

### Hint

To set up a rectangle you need code that looks like this:

```
myRect = Rect((x,y), (width,height))
```

To then draw that to the screen you need code that looks like:

```
screen.draw.filled_rect(myRect, (r,g,b))
```

### Task 1.2: Adding some text

Now we need the button to say something like "Play again?"

1. In your draw function, directly under where you've drawn the rectangle onto the screen, draw some text to the screen in the middle of the button that says something like play again?.
2. To make this text centered on the button make the x value the button's x value plus half its height and the y value the button's y value plus half its width

### Hint

To write text onto the screen you need code that looks like this:

```
screen.draw.text("My text", center = (x,y) color = (r,g,b))
```

### Task 1.3: Testing for clicks!

Now that you're displaying a button you need to test if it has been clicked.

1. Go to your `on_mouse_down()` function
2. You need to put what's called a "parameter" inside the two brackets. It needs to be called `pos`
3. Then inside the function you need an if statement to check whether `pos` is inside the button rectangle. If it is you should reset all of the values that we originally set at the beginning of the game. This includes bird's x and y values, all the pipe's x and y values, `gameOver`, and `score`, and any other values that need to be reset after a game.

### Hint

To check if a point is inside a rectangle you need to use this code:

```
if myRect.collidepoint(pos):
```

### ★ BONUS 1.4: Pretty buttons!

**Waiting for the next lecture? Try adding this bonus feature!!**

Now that you've gotten the basic button code working, you can play around with it to make it look better. For this you could work with colours or the text. You could also experiment with making the button an image.

### ★ BONUS 1.5: Pretty code

**Waiting for the next lecture? Try adding this bonus feature!!**

Now that you have a button and it resets the game when clicked, let's make a function so that its easier to read.

1. Make a new function called `restart()`

2. Copy everything that you reset from your `on_mouse_down()` function into this new function
3. In the section where you were resetting all of those values, call the `restart()` function

## TUTOR TIPS

The code should look like this:

```
# <The student's name>
import pgzrun
import sys
from random import *

# create constants
WIDTH = 800
HEIGHT = 600
score = 0
gameOver = False
button = Rect((150,375),(500,100))

# print welcome
print('''The game is about to start!
Click the mouse to "flap" upwards
Dodge the pipes and the floor
Good luck and have fun!''')

# make background
background = Actor("bg")
background.x = 400
background.y = 300

# make bird
bird = Actor("bird")
bird.x = 160
bird.y = 300

# make pipes
```

```

class Pipes():
    def __init__(self, x):
        gap = randint(160,260)
        y = randint((300-250+(gap//2)),(850-300-(gap//2)))

        self.top = Actor("top")
        self.top.x = x
        self.top.y = y - (300 + (gap // 2))

        self.bottom = Actor("bottom")
        self.bottom.x = x
        self.bottom.y = y + 300 + (gap // 2)

pipes = []
pipes1 = Pipes(266)
pipes2 = Pipes(532)
pipes3 = Pipes(798)
pipes.append(pipes1)
pipes.append(pipes2)
pipes.append(pipes3)

# draw everything to screen
def draw():
    if gameOver == True:
        screen.fill((0,0,0))
        screen.draw.text(f"Game Over!\n Your score was {score}", center
= (400,300), fontsize = 60)
        screen.draw.filled_rect(button,(255,0,0))
        screen.draw.text("Play again?", center = (400,425), fontsize =
60)
    else:
        # draw background
        background.draw()

        # draw characters
        bird.draw()
        for pipe in pipes:
            pipe.top.draw()
            pipe.bottom.draw()

```

```

# update everything
def update():
    global score, gameOver
    if gameOver == False:
        # update bird
        bird.y = bird.y + 1

        # update pipes
        for pipe in pipes:
            pipe.top.x = pipe.top.x - 1
            pipe.bottom.x = pipe.bottom.x - 1
            if pipe.top.y < -44:
                pipe.top.x = 844
                pipe.bottom.x = 844
                gap = randint(160,260)
                y = randint((300-250+(gap//2)),(850-300-(gap//2)))
                pipe.top.y = y - (300 + (gap//2))
                pipe.bottom.y = y + 300 + (gap//2)
                score = score + 1

        # bird hits bottom of screen
        if bird.y > HEIGHT:
            print("Game Over!")
            print(f"Your score was: {score}")
            gameOver = True

        # bird hits pipes
        for pipe in pipes:
            if bird.colliderect(pipe.top) or
bird.colliderect(pipe.bottom):
                print("Game Over!")
                print(f"Your score was: {score}")
                gameOver = True

# moving
def update(pos):
    global score, gameOver, pipes
    if gameOver:
        if button.collidepoint(pos):

```

```
    score = 0
    gameOver = False

    bird.y = 300

    pipes1 = Pipes(266)
    pipes2 = Pipes(532)
    pipes3 = Pipes(798)
    pipes = [pipes1,pipes2,pipes3]
else:
    bird.y = bird.y - 50

# runs everything
pgzrun.go()
```