Tutors Only

Extension: Wait to start!

Making it so the game won't start until the player has clicked the screen

Task 1.1: Setup

Let's make a variable that checks if the person has clicked

1. Make a boolean or "flag" in your create constants section called start
2. Set it to False

Hint

To make a flag it looks like:

```
myFlag = False
```

Task 1.2: Checking the flag

Now we need to check whether the player has clicked the game in the draw and update functions

1. At the top of your draw function, check if the flag is false
2. If it is, draw the background and some text that says "Click the game to start!"
3. Everything else should go in the else statement
4. At the top of your update function, check if the flag is false
5. If it is, for now just write the line `pass` as this will stop it from erroring before we put something there
6. Everything else should go in the else statement

Hint: text on the screen

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))
```

Hint: If else statements

To write an if statement with an else statement it should look like:

```
if myNum > 5:
    print("My number is greater than 5")
else:
    print("My number is lower than 5")
```

Task 1.3: Testing for clicks!

Now that you're displaying a screen waiting for a click.

1. Go to your on_mouse_down() function
2. Make the start flag global at the top of the function
3. Then inside the function you need an if statement to check if start is False. If it is, switch it to True.
4. Put everything else already in your function into the else statement

★ BONUS 1.4: Waiting animation!

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

Now that you have a basic screen that waits for the user to click it, let's make it cooler

1. In your create constants section create a count variable that starts at 0
2. In your draw function above where you draw the text, draw the bird
3. In your update function, where you've written pass, delete that
4. Add one to count and modulus(%) it by 200 (this finds the remainder when it's divided by 200)
5. Make another if statement to test if count is less than 50 or greater than 150. If it is, increase bird's y value by one

6. Otherwise, decrease bird's value by 1 (this will make bird look like it's floating up and down)

7. In your on_mouse_down() function after you make start True, set bird's y value to 300 so it starts at the right height.

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
start = False

# 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)

    def updatePipes(self,bird):
            global score, gameOver
            self.top.x = self.top.x - 1
            self.bottom.x = self.bottom.x - 1
            if self.top.x < -44:
                self.top.x = 844
                self.bottom.x = 844
                gap = randint(160,260)
                y = randint((300-250+(gap//2)),(850-300-(gap//2)))
                self.top.y = y - (300 + (gap//2))
                self.bottom.y = y + 300 + (gap//2)
                score = score + 1

            if bird.colliderect(self.top) or
            bird.colliderect(self.bottom):
                print("Game Over!")
                print(f"Your score was {score}")
                gameOver = True

    def drawPipes(self):
        self.top.draw()
        self.bottom.draw()

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 start == False:
        background.draw()
        screen.draw.text("Click the screen to start!", center =
(400,300), color = (255,255,255), fontsize = 60)
    else:
        if gameOver == True:
            screen.fill((0,0,0))
            screen.draw.text(f"Game Over!\n Your score was {score}",
center = (400,300), fontsize = 60)
        else:
            # draw background
            background.draw()

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

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

            # update pipes
            for pipe in pipes:
                pipe.updatePipes(bird)
            # bird hits bottom of screen
            if bird.y > HEIGHT:
                print("Game Over!")
                print(f"Your score was: {score}")
```

```
                gameOver = True

# moving
def on_mouse_down():
    global start
    if start == False:
        start = True
    else:
        bird.y = bird.y - 50

# runs everything
pgzrun.go()
```