Tutors Only

Extension: Projectile motion!

The bird doesn't currently fall according to physics, so to make our game easier to play and look better we will be implementing better motion

Task 1.1: The projectile motion formula

$$yt = y0 - ((v0 \times sin(\theta) \times t) - (\frac{1}{2} \times g \times t^2))$$

Where:

yt = the value of y at t seconds

$y0$ = the value of y at 0 seconds

$v0$ = the velocity at 0 seconds

$sin(\theta)$ = sin of the initial angle the projectile is "thrown at"

t = the time in seconds

g = gravity

Task 1.2: Using the formula!

Let's implement this

1. Go to directly under where the bird is created
2. Make a variable for:
   - y0 ← The bird's original y value
   - v0 ← Should be 80 (You can play around with this after it works)
   - t ← Should start at 0
   - g ← 9.81
3. Inside the update function, after you make these variables global, you will need to increase t by 0.1 (the function is run roughly 10 times per second). Where you're currently updating bird's y value you will need to substitute in the formula
   bird.y = y0 + (0.5 x g x (t^2))

The formula has changes a bit from the original as the bird falls at an angle of 0° and sin(0) is 0 so z bit of the formula cancels out

Hint

To use exponents (powers) in python it should look like:

```
squared = x**2
```

Remember to make a variable global in a function, you need to write this at the top of the function:

```
global myVar
```

Task 1.3: Using the formula pt. 2!

Now that the bird is falling more gracefully test the code. Hm.. something looks wrong

This is because we're not resetting any of the values we need to when the bird moves upwards when we click. In the on_mouse_down() function you need to reset t to 0 and you need to set y0 to the bird's y coordinate after it has moved upwards

★ BONUS 7.4: Faster and Faster!

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

Now that you have mastered basic projectile motion, play around with some of the values and make the bird's descent faster or slower. You could even try importing the math module to use sin and change the initial projectile angle

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

# 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
y0 = bird.y
t = 0
g = 9.81

# 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 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, t
    if gameOver == False:
        t = t + 0.1
        # update bird
        bird.y = y0 + (0.5 * g * (t**2))

        # 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 t,y0
    bird.y = bird.y - 50
    t = 0
    y0 = bird.y

# runs everything
pgzrun.go()
```