

Tutors Only

** Will be hard so not recommended **

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

$$y_t = y_0 - ((v_0 \times \sin(\theta) \times t) - (\frac{1}{2} \times g \times t^2))$$

Where:

y_t = the value of y at t seconds

y_0 = the value of y at 0 seconds

v_0 = 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:
 - y_0 ← The bird's original y value
 - 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
 $\text{bird.y} = y_0 + (0.5 \times g \times (t^2))$

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>
# start modules
import pgzrun
import sys

# create constants
WIDTH = 800
```

```
HEIGHT = 600
score = 0

# print welcome
print("The game is about to start!")
print('Click the mouse to "flap" upwards')
print("Dodge the pipes and the floor")
print("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
t = 0
y0 = bird.y
g = 9.81

# make pipes
topPipe1 = Actor("top")
topPipe1.x = 266
topPipe1.y = -10

bottomPipe1 = Actor("bottom")
bottomPipe1.x = 266
bottomPipe1.y = 800

topPipe2 = Actor("top")
topPipe2.x = 532
topPipe2.y = -200

bottomPipe2 = Actor("bottom")
bottomPipe2.x = 532
bottomPipe2.y = 560

topPipe3 = Actor("top")
topPipe3.x = 798
topPipe3.y = -120

bottomPipe3 = Actor("bottom")
bottomPipe3.x = 798
bottomPipe3.y = 710
```

```

# draw everything to screen
def draw():
    # draw background
    background.draw()
    # draw characters
    bird.draw()
    topPipe1.draw()
    bottomPipe1.draw()
    topPipe2.draw()
    bottomPipe2.draw()
    topPipe3.draw()
    bottomPipe3.draw()

# update everything
def update():
    global score, t
    t = t + 0.1
    # update bird
    bird.y = y0 + (0.5 * g * (t**2))
    # update pipes
    topPipe1.x = topPipe1.x - 1
    if topPipe1.x < -44:
        topPipe1.x = WIDTH
        score = score + 1

    bottomPipe1.x = bottomPipe1.x - 1
    if bottomPipe1.x < -44:
        bottomPipe1.x = WIDTH

    topPipe2.x = topPipe2.x - 1
    if topPipe2.x < -44:
        topPipe2.x = WIDTH
        score = score + 1

    bottomPipe2.x = bottomPipe2.x - 1
    if bottomPipe2.x < -44:
        bottomPipe2.x = WIDTH

    topPipe3.x = topPipe3.x - 1
    if topPipe3.x < -44:
        topPipe3.x = WIDTH
        score = score + 1

    bottomPipe3.x = bottomPipe3.x - 1
    if bottomPipe3.x < -44:

```

```

    bottomPipe3.x = WIDTH

# bird hits bottom of screen
if bird.y > HEIGHT:
    print("Game Over!")
    print("Your score was:", score)
    sys.exit()

# bird hits pipes
if bird.colliderect(topPipe1):
    print("Game Over!")
    print("Your score was:", score)
    sys.exit()

if bird.colliderect(bottomPipe1):
    print("Game Over!")
    print("Your score was:", score)
    sys.exit()

if bird.colliderect(topPipe2):
    print("Game Over!")
    print("Your score was:", score)
    sys.exit()

if bird.colliderect(bottomPipe2):
    print("Game Over!")
    print("Your score was:", score)
    sys.exit()

if bird.colliderect(topPipe3):
    print("Game Over!")
    print("Your score was:", score)
    sys.exit()

if bird.colliderect(bottomPipe3):
    print("Game Over!")
    print("Your score was:", score)
    sys.exit()

# moving
def on_mouse_down():
    global y0,t
    bird.y = bird.y - 50
    t = 0
    y0 = bird.y

```

```
# runs everything  
pgzrun.go()
```