Girls' Programming Network

Flappy Bird!

Book 2

FOR TUTOR EYES ONLY

This project was created by GPN Australia for GPN sites all around Australia!

This workbook and related materials were created by tutors at:

Sydney, Canberra and Perth

Girls' Programming Network

If you see any of the following tutors don't forget to thank them!!

Writers

Sasha Morrissey
Renee Noble
Courtney Ross
Joanna Elliott
Amanda Meli
Adele Scott
Sheree Pudney
Caitlin Bathgate
Alex McCulloch

Testers

Emily Aubin
Samantha Rampant
Leanne Magrath
Melody Wong
Vivian Dang
Annie Liu
Rosey Stewart
Natalie Bartolo
Jeannette Tran
Gaya Pilli
Cindy Chung
Stephanie Chant
Sam Criddle

Part 1: Move the Pipes

The original Flappy Bird game had the bird only going up and down and the pipes moving across the screen so let's change our game to do that!

Task 1.1: Stop that bird!

We need to stop the bird from moving left and right, so let's delete or comment out the lines of code that make the bird move left and right.

We can also delete the code that checks if the bird has gone off the edge, since that's impossible now

Hint

To make a line of code stop running we can add a # to the front

Task 1.2: Pipes on the move

Now that our bird only goes up and down we want to make the pipes move across the screen for the bird to dodge

After we update the display, make the pipe_x variable be itself minus 2. Make sure you do this for all 3 of your pipes

Task 1.3: Why slow?

You might notice that the pipes slow down when you move the bird. This is because when python prints out "Going up" or "Going down" it takes some time and slows down the game. Let's remove those print statements to keep our game nice and smooth.

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 2:

□ Your bird doesn't move left and right

□ Your pipes all move across the screen from right to left

□ You still lose the game if the bird hits a pipe

□ The game doesn't print "Going up" or "Going down" anymore

TUTOR TIPS

The code should look like this:

```
from pygame import *
init()

print("The game is about to start!")

screen = display.set_mode((800, 600))

background_image = image.load("bg.png")
bird_image = image.load("bird.png")
pipe_image = image.load("pipe.png")
flipped_pipe_image = transform.flip(pipe_image, False, True)

bird_x = 10
bird_y = 250

pipe_x = 200
pipe_y = 250
pipe2_x = 450
pipe2_y = 100
pipe3_x = 700
pipe3_y = 400

while True:
    new_event = event.poll()
    if new_event.type == KEYDOWN and new_event.key == K_UP:
        bird_y = bird_y - 50
    if new_event.type == KEYDOWN and new_event.key == K_DOWN:
        bird_y = bird_y + 50
    #Deleted the extra left and right key events and the prints

    #Added the pipe movement
    pipe_x = pipe_x - 2
    pipe2_x = pipe2_x - 2
    pipe3_x = pipe3_x - 2

    background = screen.blit(background_image, (0, 0))
    bird = screen.blit(bird_image, (bird_x, bird_y))
    pipe = screen.blit(pipe_image, (pipe_x, pipe_y))
    pipe2 = screen.blit(pipe_image, (pipe2_x, pipe2_y))
    pipe3 = screen.blit(flipped_pipe_image, (pipe3_x, -pipe3_y))
    display.update()

    if bird.colliderect(pipe):
        print("Game Over!")
        quit()
        break
    if bird.colliderect(pipe2):
        print("Game Over!")
        quit()
        break

    if bird.colliderect(pipe3):
        print("Game Over!")
        quit()
        break

    #Deleted the bird reaching the other side win condition
```

Part 2: More pipes

Having only 3 pipes makes the game very short - let’s make the game run for longer by looping the pipes around!

Task 2.1: Bye pipe!

When the pipe disappears off the screen we want it to move back to the far right of the screen!

Add an if statement after the collision if statements that checks if the pipe_x is off the left edge of the screen. If it is, change pipe_x so that it’s 800 (that will put the pipe just off the screen on the right so it’s ready to move back across the screen)

Do the same thing for all 3 of your pipes

Hint

The pipes are 87 pixels wide, so when the pipe is off the left of the screen pipe_x will be -87

Random

Task 2.2: That’s so random!

The pipes are always the same over and over again - that’s a bit boring! It would be more interesting if they picked a random height to be.

First we need to import random at the top of our code where we import pygame

Hint

To import random we can use this code:

```
from random import *
```

Task 2.3: Pick a number, any number!

Now where we set our pipe_x to 800 let’s set pipe_y to a random number between 100 and 500

Hint

To pick a random number between 1 and 100 you can use this code:

```
number = randint(1, 100)
```

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 3:

□ When a pipe goes over the left edge it comes back on the right edge

□ When the pipe reappears on the right edge it's a random height

□ Try running your code!

TUTOR TIPS

The code should look like this:

```
from pygame import *
#Added random import
from random import *

init()

print("The game is about to start!")

screen = display.set_mode((800, 600))

background_image = image.load("bg.png")
bird_image = image.load("bird.png")
pipe_image = image.load("pipe.png")
flipped_pipe_image = transform.flip(pipe_image, False, True)

bird_x = 10
bird_y = 250

pipe_x = 200
pipe_y = 250
pipe2_x = 450
pipe2_y = 100
pipe3_x = 700
pipe3_y = 400

while True:
    new_event = event.poll()
    if new_event.type == KEYDOWN and new_event.key == K_UP:
        bird_y = bird_y - 50
    if new_event.type == KEYDOWN and new_event.key == K_DOWN:
        bird_y = bird_y + 50

    pipe_x = pipe_x - 2
    pipe2_x = pipe2_x - 2
    pipe3_x = pipe3_x - 2

    background = screen.blit(background_image, (0, 0))
    bird = screen.blit(bird_image, (bird_x, bird_y))
    pipe = screen.blit(pipe_image, (pipe_x, pipe_y))
    pipe2 = screen.blit(pipe_image, (pipe2_x, pipe2_y))
    pipe3 = screen.blit(flipped_pipe_image, (pipe3_x, -pipe3_y))
    display.update()

    if bird.colliderect(pipe):
        print("Game Over!")
        quit()
        break

    if bird.colliderect(pipe2):
        print("Game Over!")
        quit()
        break

    if bird.colliderect(pipe3):
        print("Game Over!")
        quit()
        break

    #Added ifs to move the pipe back to the start & randomise the
y
    if pipe_x < -87:
        pipe_x = 800
        pipe_y = randint(100, 500)

    if pipe2_x < -87:
        pipe2_x = 800
        pipe2_y = randint(100, 500)

    if pipe3_x < -87:
        pipe3_x = 800
        pipe3_y = randint(100, 500)
```

Part 3: Flip it!

Task 3.1: Turn the pipe around!

Let's also randomise whether the pipe is facing up or facing down.

Where we set the pipe_x and pipe_y variables at the start of our code, write a new pipe_flipped variable for each pipe. Pipe1 and pipe2 should be `False` (because they are not flipped) and pipe3 should be `True`

Then, in the same place that we set the pipe_y randomly let's also set pipe_flipped to be a random choice between `True` and `False`

Make sure you do this for all 3 of your pipes

Hint

To choose randomly out of a choice of Cat or Dog I could use this code:

```
pet = choice(["Cat", "Dog"])
```

Task 3.2: Now what?

So now that we are randomly flipping the pipes, we need to write some if statements that check if the pipe is flipped or not and show the right image.

Where we are blitting the images of the pipes, before we update the display let's add an if statement that checks if the pipe is flipped. If it is we should blit the pipe_image, otherwise we should blit the flipped_pipe_image

Do this for all 3 of your pipes and remember to not blit any of the pipe images outside of the if statements

Hint

Remember that to do something when an if is not true we can use an else like this:

```
name = "Alex"
if name == "Alex":
    print("That's my name!")
else:
    print("That's a nice name too")
```

Task 3.3: Pipes are too low!

Remember how we were using -pipe3_y (notice the minus sign) to keep it coming down from the ceiling instead of poking up from the floor? Let’s update our flipped blit so that instead of blitting (pipe_x, pipe_y) it does (pipe_x, -pipe_y) - Notice the minus sign in front of the pipe_y! This will make the pipe start above the screen and hang down from the top so only do this for the flipped pipes.

☑ CHECKPOINT ☑

If you can tick all of these off you can go to Part 4:

□ Some of the pipes are flipped upside down randomly

□ The flipped pipes hang down from the top of the screen

□ Run your code!

TUTOR TIPS

The code should look like this (no bonuses):

```
from pygame import *
from random import *

init()

print("The game is about to start!")

screen = display.set_mode((800, 600))

background_image = image.load("bg.png")
bird_image = image.load("bird.png")
pipe_image = image.load("pipe.png")
flipped_pipe_image = transform.flip(pipe_image, False, True)

bird_x = 10
bird_y = 250

#Added pipe_flipped for each pipe
pipe_x = 200
pipe_y = 250
pipe_flipped = False
pipe2_x = 450
pipe2_y = 100
pipe2_flipped = False
pipe3_x = 700
```

```
pipe3_y = 400
pipe3_flipped = True

while True:
    new_event = event.poll()
    if new_event.type == KEYDOWN and new_event.key == K_UP:
        bird_y = bird_y - 50
    if new_event.type == KEYDOWN and new_event.key == K_DOWN:
        bird_y = bird_y + 50

    pipe_x = pipe_x - 2
    pipe2_x = pipe2_x - 2
    pipe3_x = pipe3_x - 2

    background = screen.blit(background_image, (0, 0))
    bird = screen.blit(bird_image, (bird_x, bird_y))

    #Added if statements to choose whether the pipe is flipped
    if pipe_flipped:
        pipe = screen.blit(flipped_pipe_image, (pipe_x, -pipe_y))
    else:
        pipe = screen.blit(pipe_image, (pipe_x, pipe_y))
    if pipe2_flipped:
        pipe2 = screen.blit(flipped_pipe_image, (pipe2_x, -pipe2_y))
    else:
        pipe2 = screen.blit(pipe_image, (pipe2_x, pipe2_y))
    if pipe3_flipped:
        pipe3 = screen.blit(flipped_pipe_image, (pipe3_x, -pipe3_y))
    else:
        pipe3 = screen.blit(pipe_image, (pipe3_x, pipe3_y))

    display.update()

    if bird.colliderect(pipe):
        print("Game Over!")
        quit()
        break

    if bird.colliderect(pipe2):
        print("Game Over!")
        quit()
        break

    if bird.colliderect(pipe3):
        print("Game Over!")
        quit()
        break

    if pipe_x < -87:
        pipe_x = 800
        pipe_y = randint(100, 500)
        pipe_flipped = choice([True, False])

    if pipe2_x < -87:
        pipe2_x = 800
        pipe2_y = randint(100, 500)
        pipe2_flipped = choice([True, False])

    if pipe3_x < -87:
        pipe3_x = 800
        pipe3_y = randint(100, 500)
        pipe3_flipped = choice([True, False])
```