

Top Down Driving Game

Workbook

Programming Tutorials by PyAngelo



Name: _____

Task 1 - Draw the Road

To draw the road you will need to understand the **coordinate system** and the following functions:

- `setCanvasSize()`
- `background()`
- `fill()`
- `noFill()`
- `stroke()`
- `noStroke()`
- `rect()`
- `circle()`

Sketch Your Version of the Road



Drawing Commands

Main Road:

Left Edge of Road:

Right Edge of Road:

Left Plant:

Right Plant:

Lane Marking 1:

Lane Marking 2:

Lane Marking 3:

Write Down Your Code

+ `main.py` Top Down Driving Game 



Check Your Knowledge

The x-axis moves _____ the screen.

The y-axis moves _____ the screen.

What command would set the canvas size to be 640 pixels wide and 360 pixels high?

What command would draw a rectangle with the lower left corner at (20, 50) with a width of 100 and a height of 200?

What command would draw a circle with its centre at (550, 20) with a radius of 10 pixels?

Task 2 - Animate the Road

To animate the road you will need to understand:

- How animation works for computers
- Infinite loops
- If statements
- The PyAngelo Sprite Library

Animation

Animation in computers works by looping through the following actions in rapid succession:

- Clearing the screen
- Moving the position of some images in memory
- Drawing your images on the screen
- Sleeping for a small amount of time

Infinite Loops

In Python, you can create an infinite loop using a “while True:” loop. The block of code that is indented below the “while True:” will be repeated forever.

If Statements

If statements allow you to execute a block of code only if a certain condition is met. We will use an if statement to check if our object has passed the bottom of the canvas and if so we'll respawn it at the top. Example if statement:

if dash1.y < -dash.height:

dash.y = height # Only executed if dash has passed the bottom of the canvas

PyAngelo Sprite Library

To include the PyAngelo sprite library in your sketch use the following code at the top of your program:

```
from sprite import *
```

RectangleSprite

A rectangle sprite is used to draw and move a **rectangle** around the canvas. You create a RectangleSprite with the following command:

```
dash = RectangleSprite(295, 0, 10, 50)
```

This creates an object called dash that has an initial location of (295, 0) with a width of 10 and a height of 50.

CircleSprite

A circle sprite is used to draw and move a **circle** around the canvas. You create a CircleSprite with the following command:

```
plant = CircleSprite(500, 20, 10)
```

This creates an object called plant that has an initial location of (500, 20) with a radius of 10 pixels.

Sprite Methods

draw(), moveBy(), setColour()

Sprite Properties

x, y

width, height (RectangleSprite)

radius (CircleSprite)

Sample Animation Code

Top Down Driving Game 

+ main.py



Check Your Knowledge

Does the following program draw anything to the screen?

```
from sprite import *  
  
dash = RectangleSprite(100, 20, 20, 40)  
  
while True:  
    background(220, 220, 220)  
    dash.moveBy(1, 1)
```

What would the following program print to the screen?

```
from sprite import *  
  
dash = RectangleSprite(100, 20, 30, 40)  
  
print(dash.x)  
print(dash.y)
```

What would the following program print to the screen?

```
from sprite import *  
  
dash = RectangleSprite(100, 20, 30, 40)  
  
print(dash.width)  
print(dash.height)
```

What command would I need to type to draw the circle on the canvas and on what line? What would happen if I drew the plant between lines 4 and 5?

```
1 from sprite import *
2 plant = CircleSprite(100, 50, 30)
3 setCanvasSize(640, 360)
4 while True:
5     background(100, 100, 100)
6
```

Task 3 - Create Your Car

Drawing Your Car

To create your car you need to create an image. You can use any graphics program and one such online program is <https://www.photopea.com>

When you create your image ensure:

- It is 100px wide by 150px high
- It has a transparent background
- Your drawing takes up most of the space as the full size of the image is used for collision detection even if you haven't drawn near the edge

Controlling Your Car - Key Presses

In PyAngelo we can tell if the user is pressing the key using the `isKeyPressed()` function. The function takes one parameter which is the key we are checking. PyAngelo has constants you can use for each key:

- `KEY_W`
- `KEY_A`
- `KEY_S`
- `KEY_D`
- `KEY_SPACE`
- `KEY_ENTER`
- `KEY_LEFT`
- `KEY_RIGHT`
- `KEY_UP`
- `KEY_DOWN`

We can check if a certain key has been pressed and then move our car using the `moveBy()` function.

Sample Code To Control a Sprite

Top Down Driving Game 

+ main.py



Check Your Knowledge

What is wrong with the following code?

```
from sprite import *  
  
car = Sprite("Car.png", 250, 450)  
  
setCanvasSize(600, 600)  
  
while True:  
    car.draw()  
    background(100, 100, 100)
```

What is wrong with the following code?

```
from sprite import *  
  
car = Sprite("Car.png", 250, 450)  
  
setCanvasSize(600, 600)  
  
while True:  
    background(100, 100, 100)  
    car.draw()  
    if keyPressed(KEY_LEFT):  
        car.moveBy(1, 0)  
    sleep(0.005)
```

Task 4 - Enemy Cars

Drawing Your Car

Draw your enemy car using a graphics program such as photopea.com.

Animating the Enemy Car

The enemy car is driving in the same direction as us but we are going faster so we will overtake them. To make this look realistic, when our car is staying at the bottom of the screen, we need the enemy car to be moving backwards down the screen. It must move slower than the lane markings so it appears to be moving forward.

Variables for Speed

To enable us to test different speeds, we should create variables that control the speed of:

- The road
- The enemy car

The speed of the road should be faster than the speed of the enemy car.

Create Your Changes

Look back over the notes you have created and attempt to add the enemy cars and animate them in your program.

Check Your Knowledge

What is wrong with the following code?

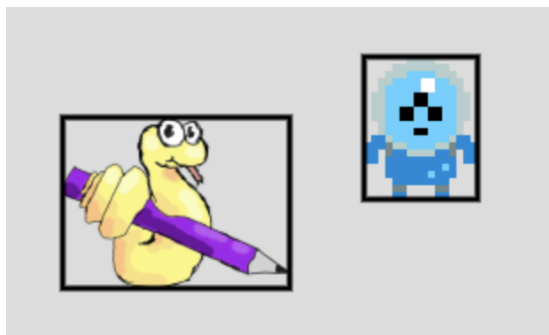
```
from sprite import *  
  
car = RectangleSprite("Car.png", 250, 450)  
  
setCanvasSize(600, 600)  
  
while True:  
    background(220, 220, 220)  
    car.draw()  
    sleep(0.005)
```

What is wrong with the following code?

```
from sprite import *  
  
car = Sprite(Car.png, 250, 450)
```

Task 5 - Collision Detection

Bounding Boxes



A simple method for checking if two sprites are colliding is called bounding boxes. This uses a box around each image to determine if they are overlapping. This has some limitations as shown in the following image:



These sprites are not touching visually, but their bounding boxes do overlap and so would be considered to be colliding using the bounding boxes method. Be aware of this when creating images for your games.

Collision Detection Sample Code

Top Down Driving Game 

+ main.py

Check Your Knowledge

What will the following code print to the console?

```
from sprite import *  
  
box = RectangleSprite(100, 100, 100, 100)  
  
ball = CircleSprite(100, 100, 10)  
  
if box.overlaps(ball):  
    print("Overlapping")  
  
else:  
    print("Not overlapping")
```

The image Car.png has a width of 100 pixels and a height of 150 pixels. What will the following code print to the console?

```
from sprite import *  
  
car = Sprite("Car.png", 200, 100)  
  
catcher = RectangleSprite(250, 50, 10, 10)  
  
if car.overlaps(catcher):  
    print("Overlapping")  
  
else:  
    print("Not overlapping")
```

Task 6 - Game States

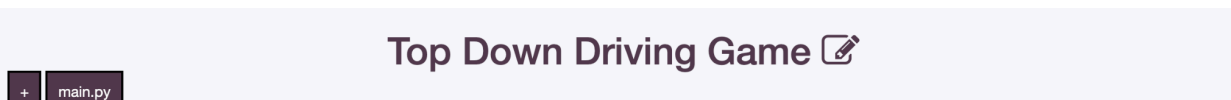
Most games have several states:

- Introduction - may show some instructions and animations
- Play - The main part of the game where you actually play the game
- Game over - a screen for when the game ends

Adding an Introduction and Game Over State

To add an introduction and a game over state to our game, we need to add some more sections to our game. We can do this by creating while loops that end when a condition is met such as pressing the spacebar or the enter key. This is very similar to the code we wrote for stopping the play when cars collide.

Sample Introduction Screen



Infinite Loop Around All 3 Game States

Top Down Driving Game 

+ main.py

Task 7 - Extension Ideas

Random Positioning of Enemy Car

At the moment the enemy car always respawns at the same x position. To make the game more interesting you could respawn the car at a random x position. To do this import the random module with the following code:

```
from random import *
```

Then you can use the randint function to choose a random x position:

```
enemyCar.x = randint(0, width - enemyCar.width)
```

Background Music and Sound Effects

Adding Background Music

```
music = loadSound("/samples/music/Turbo_Outrun_01.mp3") # Not in animation loop
```

```
playSound(music, loop = True) # Not in animation loop
```

Adding Sound Effects

```
collision= loadSound("/samples/sounds/collision.wav") # Not in animation loop
```

```
playSound(collision) # When a collision occurs
```

Scoring

There are a number of options for creating a scoring system. You could gain a point every time an enemy car moves off the bottom of the screen, or you could gain a point for every time the program goes through the animation loop.

You will need to create a **score** variable that is reset to zero every time the game loop starts.

Adding More Enemy Cars

Why only have one enemy car? You could create more sprites and have more cars!

Adding Pickups

At the moment the only challenge is to avoid the enemy cars. You could create a sprite that when the car overlaps it, the player gets some kind of reward. Perhaps they get extra points, or the speed at which they move could increase.

Increasing the Speed

Currently the speed of the enemy car remains constant. You could make the game harder by increasing the speed when the player reaches a certain score. You may also want to increase the speed of the lane markings at this time too to make it look like we are driving faster as well.