Algorithm and Programming

Final Project

Application of Pygame in Python as Game Building Usage

A picture containing logo

Description automatically generated

Benedictus Filbert Federico

2502005263

L1AC

Computer Science

1. Definition

As a rookie in the programming world, I wish to re-create a game called flappy bird. Flappy bird is a 2D side-scroller game. The bird moves a constant speed along the X-axis, the player is required to tap the screen allowing the bird to gain lift as well as avoid obstacle with combining game mechanics of lift and gravity the player’s goal is to maneuver the bird as long as the player can avoid as many obstacles resulting as their score.

2.1 Coding

To create the game successfully, some modules are required such as sys module, pygame module, and random module. As for version requirements, it is advised that user installed the latest version. As for bear minimum the pygame version must at least to be version 1.9. This program is created with python therefore it is advised to use the latest python version.

Text

Description automatically generated

To start the program, pygame.innit() is used like any other pygame related. Next is the display screen, flappy bird is known to be a portrait resolution game, I use 288 as the width and 512 as the height.

Text

Description automatically generated

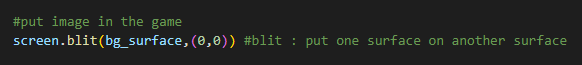
The codes in the picture purpose is for the program to keep running, accepts inputs, and prevention of crashing upon the closing of the program. Pygame.event.get() is for pygame to recognize any activity the user do. Pygame.QUIT is for the program to close if the user press the X button on the top right on the screen. Finally sys.exit() is to properly close the program.

Next is to control the display, to do this there’s the code : Text

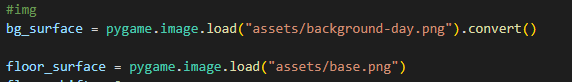
Description automatically generated pygame.display.update() is used to display any code typed above this line, as context that is the line above 215 as seen in the picture. There is also the clock which regulates how many frames per second the program will run. The clock is based on the code :



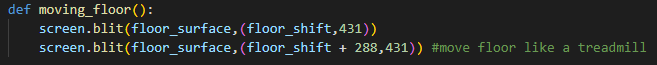
Next is to load the pictures in the game :



The code above is to load in the background in the program, “blit” is use to put a surface to the program. I will use the variable to input the background. As for 0,0 is the coordinates within the x and y axis, 0,0 is located on the top left of the screen, increasing x will to the right, increasing y will go the bottom, likewise decreasing x and y will do the opposite. Without any intervention, the background should have the same area as the screen.



Here is how I input the background image and the floor, however to make the floor move, a simple trickery is required. The moving floor is actually a copy of one and move the previous floor image to the left, the newly copied floor image will move and replace the previous floor were. This whole process repeats like a treadmill, as seen in the codes below:



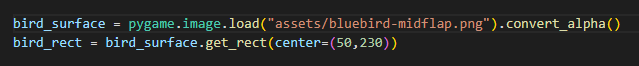
The above is specifically for the floor the move left, the number 431 is fofr the floor to be position on the bottom of the screen, the number 288 is exactly the width of the screen therefore it spawn a new floor outside the screen. The process of addition of more floors are like below:

Text

Description automatically generated

The If loop is required so the whole process goes infinitely.

Next is to load the bird:



convert\_alpha() and convert() isn’t strictly necessary but its purpose is to convert the image into a type of file that is easier for python to run, it will be mention again later so as for now lets focus on loading the bird. To load the bird I use; pygame.image. load("assets/background-day.png"). after that is for get.rect() is to put a rectangle around the bird, the purpose the rectangle is for entity collision and tilting of the bird when flying. The numbers that are 50 and 230 is for the bird so upon starting the program the bird will spawn on the center point that is those numbers, practically it spawned in the middle of the screen. Next is to use blit in the while True loop so the bird gets load in the game. Upon the loading up the bird it’ll stay stationary with no presence of gravity, therefore lets make the gravity:  I made a variable that gravity is 0.25. this variable has no connection whatsoever with the gravity of earth. Next variable is bird\_movement, the point of this is we are going to add gravity to every frame to the bird\_movement thus moving the bird\_rect(previous variable) downward and moving the bird\_surface to a different coordinate.  is used in the while True loop. This is a loop therefore the gravity will add up per loop making it faster per loop, just like in real life where gravity speeds up if there are more distance to cover. To give the bird the ability to fly or jump : Text

Description automatically generated

Diagram, box and whisker chart

Description automatically generated with medium confidenceas we know there is a thing called momentum, gravity is set as 0.25 and even that point it’ll double or triple the number because the bird is already falling. Simply observe the picture, the loop is quite fast resulting the buildup of speed in the gravity variable. Practically speaking it’ll only dissipate the momentum of the bird falling. bird\_movement = 0 is for cancelling all gravity elements and enabling the bird to jump perfectly. As for the death sound, it’s for when the bird goes outside the screen or crash to the floor as it counts as deaths.

Gravity = 10

Gravity = 0.25

Elevate =

-4.5

Gravity = 5

Next is loading up the pipes simply by :

Text

Description automatically generated

To infinitely spawn these pipes we need the empty list. To create an interval between each pipe spawning the timer is needed. Next is to put it in the while True loop :

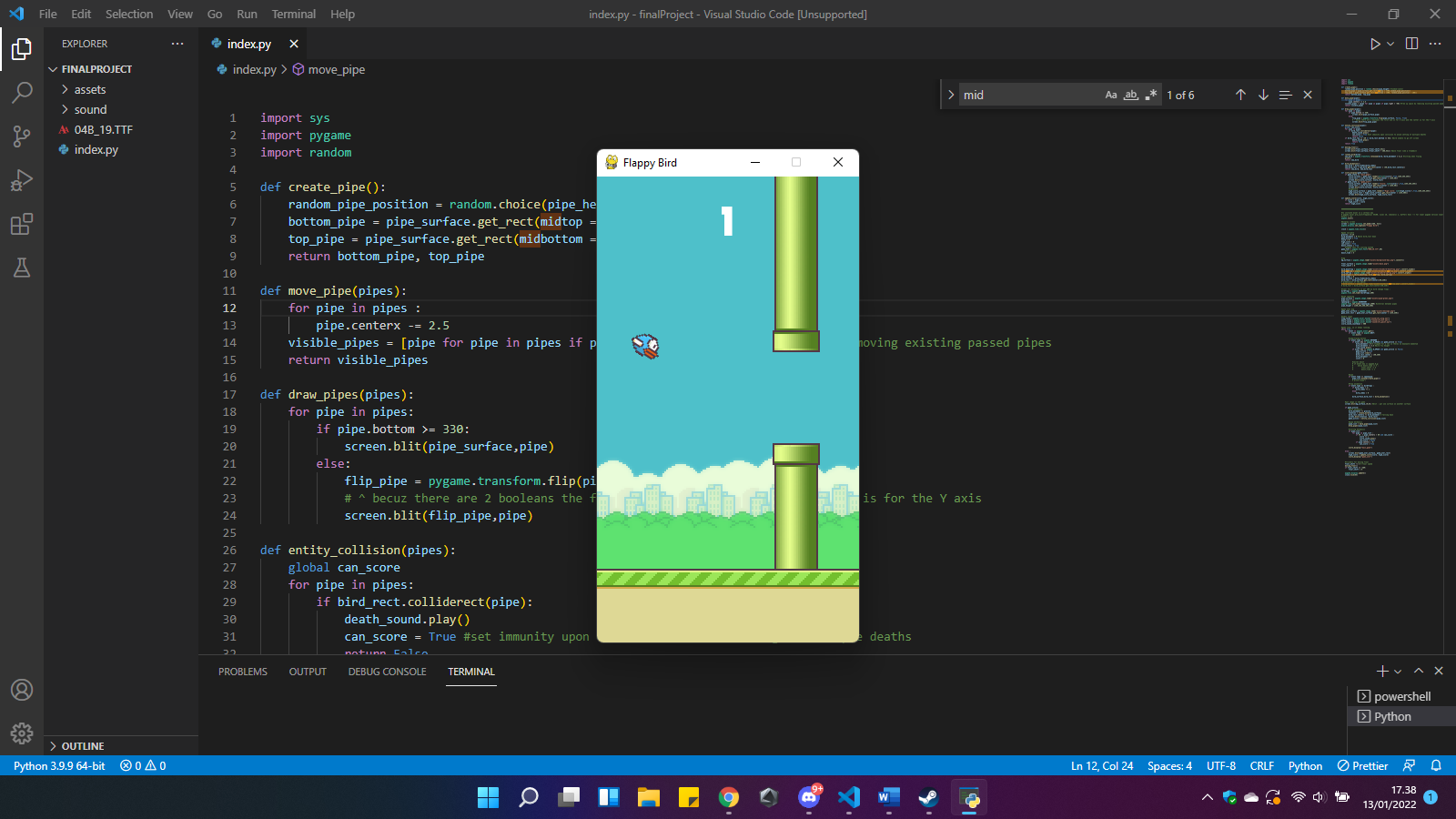


As the mechanics for spawning the pipes are near completion, next is to spawn the pipes and to resize their heights as these pipes are equal in height.



Text

Description automatically generated

For this the random module is required. As you can see the pipe height are randomly chosen from the pipe\_height list, the random number will be use to spawn the differing the pipe heights. The number 300 is so the pipes spawn outside of the screen. the top pipe gets subtracted by 100 so it’ll get higher so there are some spaces between the two. The pipes are almost complete however running the game how it is right now will resulted the top pipe in an awkward position, therefore the code here will help with this problem

Graphical user interface, text

Description automatically generated

This code will check if the pipes are above y axis 330 because around 330 is the top pipe, with the use of Boolean expression therefore it’ll flip the pipe with the picture on the right as an example. With that all done next is implementing entity collision, in order for the program to recognize the obstacle we will use get.rect() of the bird and the pipe. Upon collision the program will end the game meaning the bird crashed into the pipe. The code below is used for entity collision :

A screenshot of a computer

Description automatically generated with medium confidence



First create a variable called can\_score this usage will be useful later, next is the variable game\_active, because on impact the program instantly declare the bird death without this the death sequence will be twitchy and sending a death declaration a couple times. Simply meaning if the death sound is active it’ll activate the sound a couple times because the program act on milliseconds therefore within this millisecond the dying not once. Game\_active is to deactivate the game mechanisms. Entity\_collision(pipes\_list) purpose is upon collision within bird an pipe the program declare the bird crashed. The if function purpose is when the bird fly over the screen, the program declares the bird dead. Text

Description automatically generatedAs for game\_active : It’s a key for the game to activate.Text

Description automatically generated any game mechanisms such as gravity, bird mechanisms, and pipe spawning. If this function deactivate it will trigger the game over screen.When facing the game over screen simply pressing the space bar will restart the game. Pipe\_list.clear() will empty out the pipes and put the bird in the center of the screen with static momentum because of bird\_movement = 0. There’s also a risk of lagging the game or slowing down because of all the pipe spawning, therefore excess pipes need to remove that is those pipes the bird passed, simply using one line of code:

visible\_pipes = [pipe for pipe in pipes if pipe.right > -50]

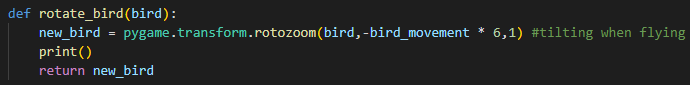
Text

Description automatically generated

Next is for the bird animations, that are tilt of the bird when flying & falling and flapping animation. As for tilting:

A screenshot of a computer

Description automatically generated with medium confidence



In pygame, rotozoom is use to rotate something. The first argument is the object we want to rotate, second is the angle of rotation. There is an alternative way to code this but this is visually more easily to understand. After completing this function there’ll be a black box surrounding the bird to remove this we use convert\_alpha() :

Next lets animate the bird flapping giving the illusion that it is flying.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

How this work is the timer sets the pace for the image of the bird to swap, it goes from downflap to midpflap then to upflap and repeating the whole sequence orderly.

These images of different stages of the bird flapping may have different dimensions therefore it is required that we put a new rectangle such as below

Text

Description automatically generated

Text

Description automatically generated

New\_bird will be the new surface and updates the bird. New\_bird\_rect is where we put the get\_rect() and as for the number 50 is for 50 meters from the left of the screen, as for the y is using the center y from the previous bird.

Next the text for the game and the scoring system. Flappy bird has specific font style called 04B\_19.ttf with size 40.

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidenceFirst is the score variable that is taken from the game, it’ll be explain later. 255,255,255 is RGB with each of these numbers it’ll give white color. Next is to get the score text in the top middle of the screen. The x is half of the width of the screen and y is 50 pixels from above of the screen, finally is to use blit in load in the text afterwards put score\_display() in game\_active.

To add the game over screen is simple that is to use load like the picture below after that is to add the whole thing like the picture above, in else part of game\_active so it’ll work when game\_active returns as False

A screenshot of a computer

Description automatically generated with medium confidence

Note that if the game over screen has some black box around simply put convert\_alpha() after pygame.image.load() like so :



Now lets discuss about the scoring system:

Text

Description automatically generated

Text

Description automatically generated

How flappy bird scoring works is that the game adds a score when the bird pass through the pipes, to be precise is when the pipe goes to the middle since the bird doesn’t actually move rather the floor and the pipes that are moving, since my screen width is 288 the game it add up the score when the center point of the pipe pass through the bird is counts as a score. However this scoring function will trigger to many times since the program uses milliseconds, that’s where can\_score goes in handy. Upon scoring the program will deactivate the scoring mechanism and after the pipe went by for a while using : will activate the scoring mechanism back

Last is the sound effects. As seen from below, to import sound pygame.mixer.sound() is used

Text

Description automatically generated with medium confidence

To put it in the game, put the code in the while True while and in the pygame.KEYDOWN section, as the sound will activates every time the player presses the space bar

Text

Description automatically generated

Text

Description automatically generatedThe death sounds are inserted in the entity collision function :

where as the bird would crash into an obstacle or the floor or maybe fly to high it’ll play the death sound.

Note if the user’s pygame version is lower than version 2 than this is required to avoid delays on the sounds: 

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated2.2 Program Execution

The program works just fine, it’s not mastercrafted but neatly done either way

2.3 Scrapped Material

Text

Description automatically generated

The original intention was to create a boost system, the bird would fly faster approximately twice as fast. Theoretically this can be done by increasing floor\_shift as it increases the floor speed next is to decrease intervals between pipes that is by decreasing the timer of the pipe spawing. These two elements will create a trick that the bird is using the boost. However this boost function cannot be implemented because for unknown reason the game simply crash upon pressing left shift(the trigger for the boost), as this is out of the original code there isnt a way to get this in the game legitimately therefore it is activated in-game using cheat

3. Evaluation

While it is not a daunting task to create flappy bird, I had my own share of problems too for example like using convert() after pygame.load() it some how disrupt get\_rect() and compromising the whole program since get\_rect() is widely used in collision and partially in animation. I did had fun creating this, flappy bird is current my second game I re-created with the first that is pong. Although its pale to comparison to flappy bird. I did try not to copy the whole thing and tried to add some of my own codes alas it didn’t work and with time limitation I can’t do much. I believe in due time I’ll be able to perfect the codes