Flappy Bird

Project Report

Version 1.0

Industrial Training & Presentation (ECS 791)

BACHELOR OF TECHNOLOGY (CSE)

PROJECT GUIDE:

Ms. Rohaila Naaz

SUBMITTED BY:

Aman Madhok (TCA1809023)
Kshma Sethi (TCA1809072)

December, 2021

COLLEGE OF COMPUTING SCIENCES AND INFORMATION TECHNOLOGY

TEERTHANKER MAHAVEER UNIVERSITY, MORADABAD

ACKNOWLEDGEMENT

On the submission of our thesis report on “Flappy Bird”, we would like to extend our gratitude and sincere thanks to our supervisor Ms. Rohaila Naaz, Department Computer Science Engineering for her constant motivation and support during the course of our work in the last semester. We truly appreciate and value her esteemed guidance and encouragement from the beginning to the end of this thesis. We are indebted to her for having helped us shape the problem and providing insights towards the solution.

We want to thank all our teachers Mr. Aditya Jain, Mr. Tushar Mehrotra, Mr. Rupal Gupta for providing a solid background for out studies and research thereafter. They have been great sources of inspiration to us and we thank them from the bottom of our heart.

Also we would like to convey our gratitude to Udemy, YouTube and Google who gave us necessary information and guidance for these projects. Which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Above all, we would like to thank all our friends whose direct and indirect support helped us complete our project in time. The thesis would have been impossible without their perpetual moral support.

Aman Madhok (TCA1809023)
Kshma Sethi (TCA1809072)

Place: Moradabad

Date: 11/12/2021

DECLARATION

We hereby declare that this Project Report titled “Flappy Bird” submitted by us and approved by our project guide, to the College of Computing Sciences and Information Technology (CCSIT), Teerthanker Mahaveer University, Moradabad, is a Bonafede work undertaken by us and it is not submitted to any other University or Institution for the award of any degree diploma / certificate or published any time before.

Project Group:

Student Name:	Aman Madhok	TCA1809023
Student Name:	Kshma Sethi	TCA1809072
Project Guide:	Ms. Rohaila Naaz	

BRIEF ABOUT THE COMPANY

Internshala is an internship and online training platform, based in Gurgaon, India. Founded by Sarvesh Agrawal, an IIT Madras alumnus, in 2011, the website helps students find internships with organizations in India.

Internshala is a reputed Indian company. So, if you do any training or internship from the internship, the chances of getting a job increase a lot. They provide different kinds of courses, internships, training. Anyone can choose training or internship as their need. In addition to programming, personal development, digital marketing, there are many other courses through which anyone can easily improve their skills. You can do any course from here at a very low cost.

Internshala is a Gurgaon-based web venture providing internship resources and career services to students and employers.

The site offers internship searching and posting, and other career services such as counselling, cover-letter writing, resume building and training programs to students.

Internshala helped equipping us with relevant skills & gave us practical exposure to help us get the best possible start to our careers. It helped us discover our passion and turning it into your career.

Smart India Multi Management Institute is a pan India NGO registered in Haryana, India; carrying out welfare projects on education, healthcare, livelihood and women empowerment. We believe that unless members of the civil society are involved proactively in the process of development, sustainable change will not happen. Based on this, Simmi Foundation sensitizes and engages the civil society, making it an active partner in all its welfare initiatives.

We envision to develop a society based on legitimate rights, equity, justice, honesty, social sensitivity and a culture of service in which all are self-reliant.

Table of Contents

Table of Figures

1 Project Title

Flappy Bird using Python.

2 Problem Statement

Our objective for this project is to implement the game on computer screen. We try to create user-friendly interfaces to allow user better understand the game instructions and workflows. The game allows users to choose playing modes and background settings, and to control bird using keyboard buttons. It also contains a score board function to track previous history.

3 Project Description

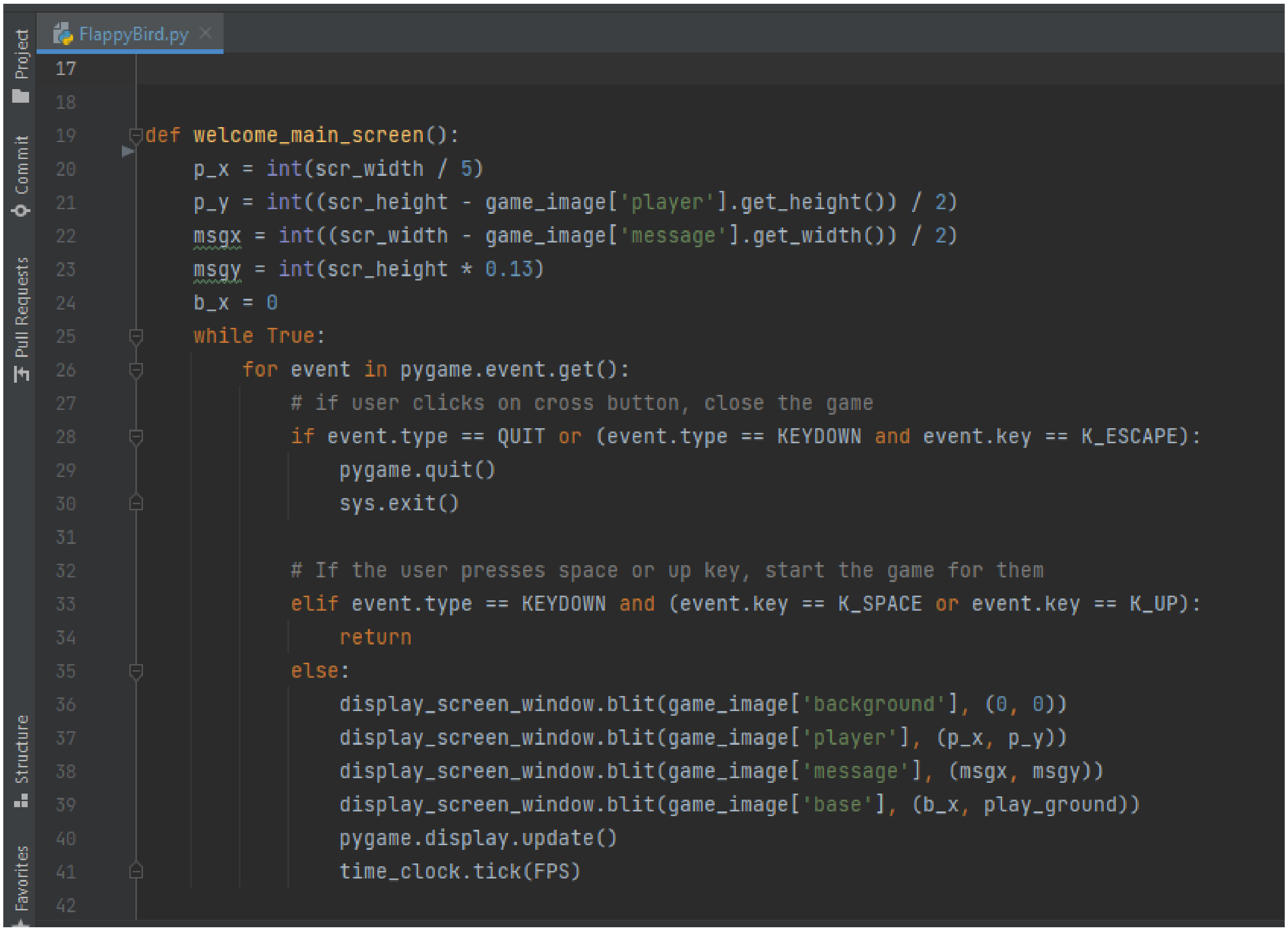
The In this project, we design and implement a Flappy Bird like video game. Flappy Bird is a very popular mobile game on Android platform, driving a lot of people crazy. In this game, the player can control the vertical movement of bird (every pressing on the keyboard makes the bird leap upward for a little bit, and the bird will fall freely without control). As soon as the game begins, tubes will keep appearing from the right side of the screen and moving leftwards. (So that it seems like the bird flying forward). The goal of this game is to control the bird, dodging and passing the incoming tubes, as many as possible. The game is endless until the bird eventually hit one of the tubes, ground, or ceiling.

3.1 Scope of the Work

Our design process is to first create the static interfaces. Then we started to create single player mode, including pipe and background shifting, bird falling or jumping, score and life tracking, sound effect.

3.2 Project Modules

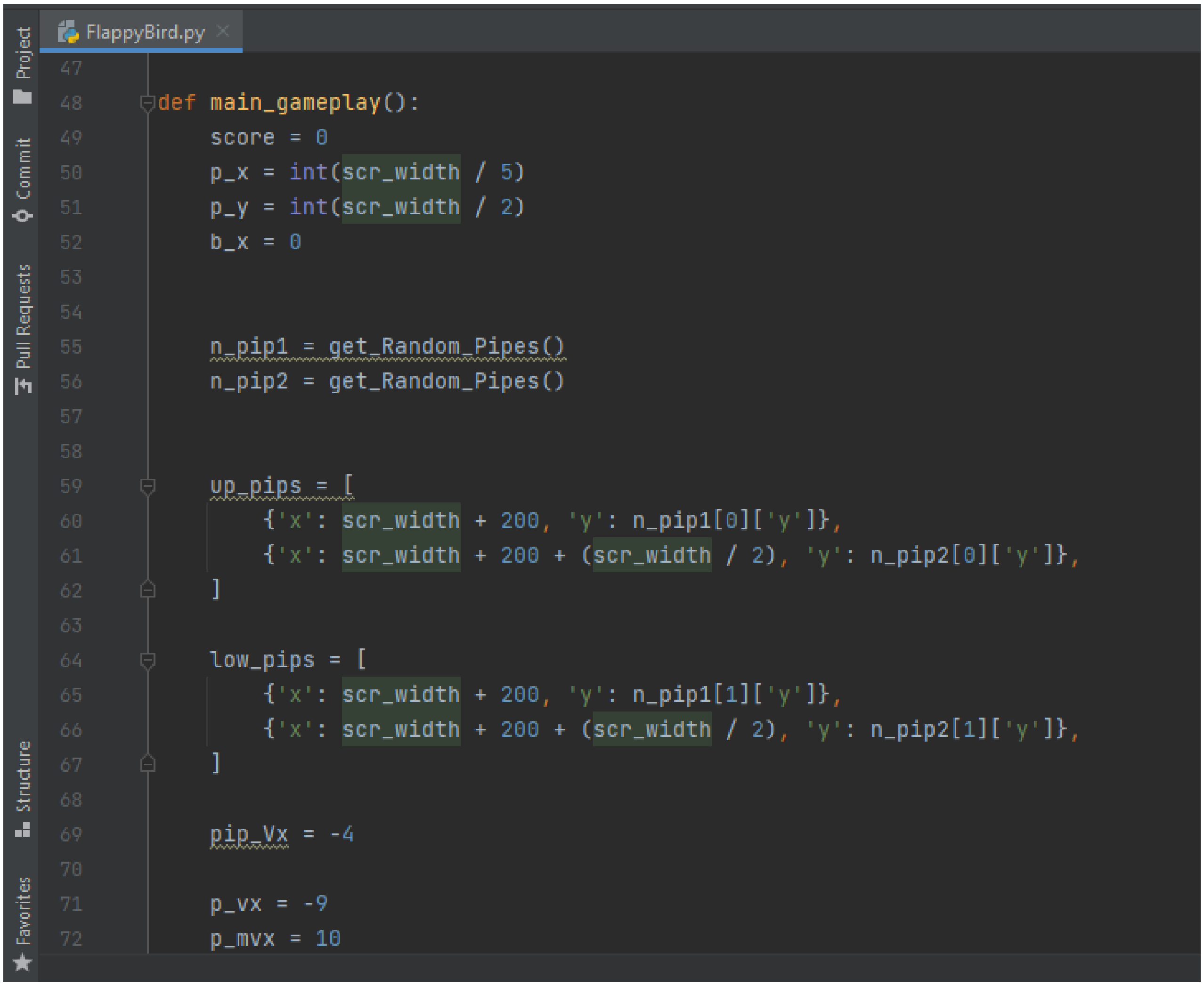
- **Main Screen**- It will display the welcome screen on starting the game. We start by assigning the values of the x-coordinate and y-coordinate for the player, message, and title images. Here we make use of a for loop for analyzing all the events taking place throughout the game using pygame.event.get() function. Then we check that whenever a quit type of event is encountered by pressing the escape key, the game window will close.



```python


def welcome_main_screen():
    p_x = int(scr_width / 5)
    p_y = int((scr_height - game_image['player'].get_height()) / 2)
    msgx = int((scr_width - game_image['message'].get_width()) / 2)
    msgy = int(scr_height * 0.13)
    b_x = 0
    while True:
        for event in pygame.event.get():
            # if user clicks on cross button, close the game
            if event.type == QUIT or (event.type == KEYDOWN and event.key == K_ESCAPE):
                pygame.quit()
                sys.exit()

            # If the user presses space or up key, start the game for them
            elif event.type == KEYDOWN and (event.key == K_SPACE or event.key == K_UP):
                return
            else:
                display_screen_window.blit(game_image['background'], (0, 0))
                display_screen_window.blit(game_image['player'], (p_x, p_y))
                display_screen_window.blit(game_image['message'], (msgx, msgy))
                display_screen_window.blit(game_image['base'], (b_x, play_ground))
                pygame.display.update()
                time_clock.tick(FPS)

```

Fig 1. Code Snippet for Main Screen

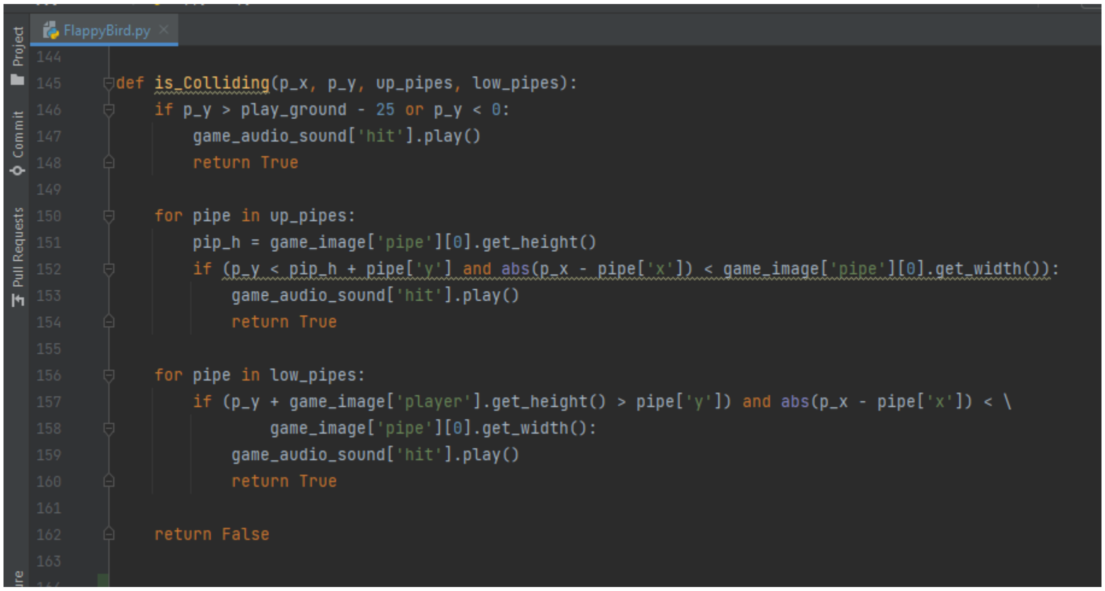
- **Gameplay**- This is the main module, Defining the variable score as 0, bird velocity, minimum bird velocity, maximum bird velocity, and pipes velocity. Handle the key events using pygame.event.get() and checking for the game is over or not if it is over return from the function. Updating the score and in game images such as background, pipe, and bird on the window.language.



```python

def main_gameplay():
    score = 0
    p_x = int(scr_width / 5)
    p_y = int(scr_width / 2)
    b_x = 0


    n_pip1 = get_Random_Pipes()
    n_pip2 = get_Random_Pipes()


    up_pips = [
        {'x': scr_width + 200, 'y': n_pip1[0]['y']},
        {'x': scr_width + 200 + (scr_width / 2), 'y': n_pip2[0]['y']},
    ]

    low_pips = [
        {'x': scr_width + 200, 'y': n_pip1[1]['y']},
        {'x': scr_width + 200 + (scr_width / 2), 'y': n_pip2[1]['y']},
    ]

    pip_Vx = -4

    p_vx = -9
    p_mvx = 10
```

Fig 2. Code Snippet of Gameplay

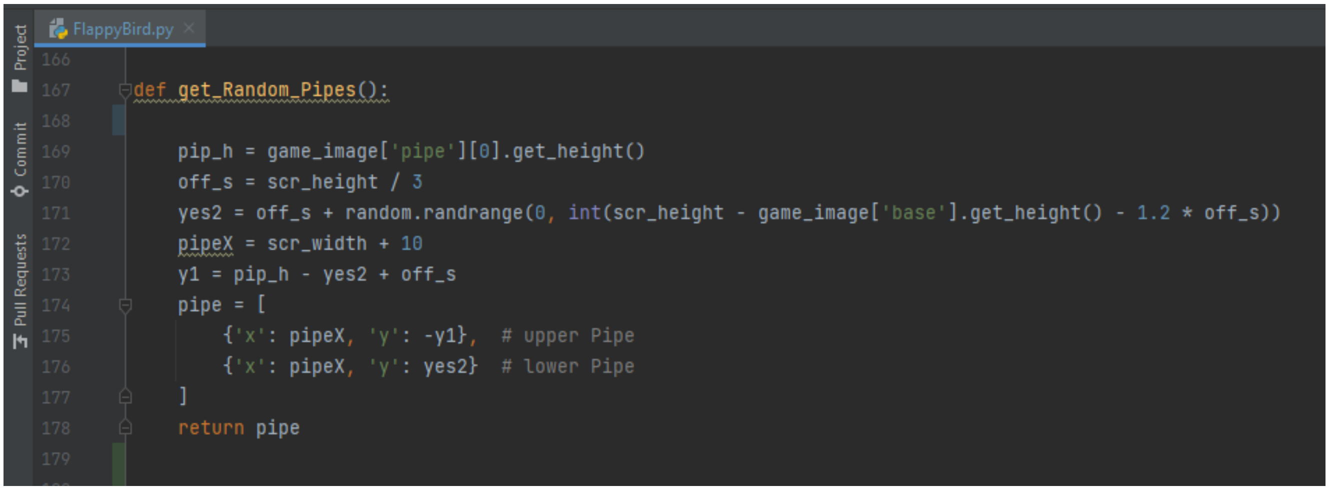
- **Colliding**- represents whether the bird has hit the pipes or fall into the sea. According to my thought, three conditions lead to a situation of the game over. if the difference between our elevation and a certain height is less than vertical it means the bird has crossed its boundaries resulting in a game over and if the bird hits any of the lower and upper pip then this will also lead to game over condition.



```
def is_Colliding(p_x, p_y, up_pipes, low_pipes):
    if p_y > play_ground - 25 or p_y < 0:
        game_audio_sound['hit'].play()
        return True

    for pipe in up_pipes:
        pip_h = game_image['pipe'][0].get_height()
        if (p_y < pip_h + pipe['y'] and abs(p_x - pipe['x']) < game_image['pipe'][0].get_width()):
            game_audio_sound['hit'].play()
            return True

    for pipe in low_pipes:
        if (p_y + game_image['player'].get_height() > pipe['y']) and abs(p_x - pipe['x']) < \
                game_image['pipe'][0].get_width():
            game_audio_sound['hit'].play()
            return True

    return False
```

Fig 3. Code Snippet of Colliding

- **Random Pipes**- generates a new pipe of random height. First of all, we have to fetch the height of the pipe using getheight() function. After this generates a random number between 0 to a number (such that the height of the pipe should be adjustable to our window height). After that, we create a list of dictionaries that contains coordinates of upper and lower pipes and return it.



```
def get_Random_Pipes():

    pip_h = game_image['pipe'][0].get_height()
    off_s = scr_height / 3
    yes2 = off_s + random.randrange(0, int(scr_height - game_image['base'].get_height() - 1.2 * off_s))
    pipeX = scr_width + 10
    y1 = pip_h - yes2 + off_s
    pipe = [
        {'x': pipeX, 'y': -y1},  # upper Pipe
        {'x': pipeX, 'y': yes2}  # lower Pipe
    ]
    return pipe
```

Fig 4. Code Snippet of Random Pipes

4 Implementation Methodology

We all are familiar with this game. In this game, the main objective of the player is to gain the maximum points by defending the bird from hurdles. Here, we will build our own Flappy Bird game using Python.

This is the step-wise implementation of this project.

Step 1: In this first step, we have to import libraries.

- After that, we have to set the height and width of the screen to which the game will be played. Now we have to define some images which we shall use in our game like pipes as hurdles, bird's images, and also a background image of the flappy bird game.

Step 2: After declaring game variables and importing libraries we have to initialize the Pygame.

- Initialize the program using pygame.init() and set the caption of the window. Here pygame.time.Clock() will be used further in the main loop of the game to alter the speed the bird. Load the images from the system in pygame using pygame.image.load().

Step 3: Initialize the position of the bird and starting the game loop.

- Initialize the position of bird and sea level to the ground. Adding conditions in the loop defines the game conditions. The variable horizontal and vertical is used to set the position of the bird. We have to run the program until the user stops or exits (using sys.exit()) if the program so, we create an infinite while loop.

Step 4: Create a function that generates a new pipe of random height.

- First of all, we have to fetch the height of the pipe using getheight() function. After this generates a random number between 0 to a number (such that the height of the pipe should be adjustable to our window height). After that, we create a list of dictionaries that contains coordinates of upper and lower pipes and return it.

Step 5: Now we create a GameOver() function which represents whether the bird has hit the pipes or fall into the sea.

- According to my thought, three conditions lead to a situation of the game over. if the difference between our elevation and a certain height is less than vertical it means the bird has crossed its boundaries resulting in a game over and if the bird hits any of the lower and upper pip then this will also lead to game over condition.

Step 6: Now we will be creating our main function (flappygame()) that will do the following things:

- Initialize the variables and creating two pipes by createPipe() function. Create two lists first is of lower pipes and the other is of lower pipes. Defining the bird velocity, minimum bird velocity, maximum bird velocity, and pipes velocity. Handle the key events using pygame.event.get() and checking for the game is over or not if it is over return from the function. Updating the score and blit game images such as background, pipe, and bird on the window.

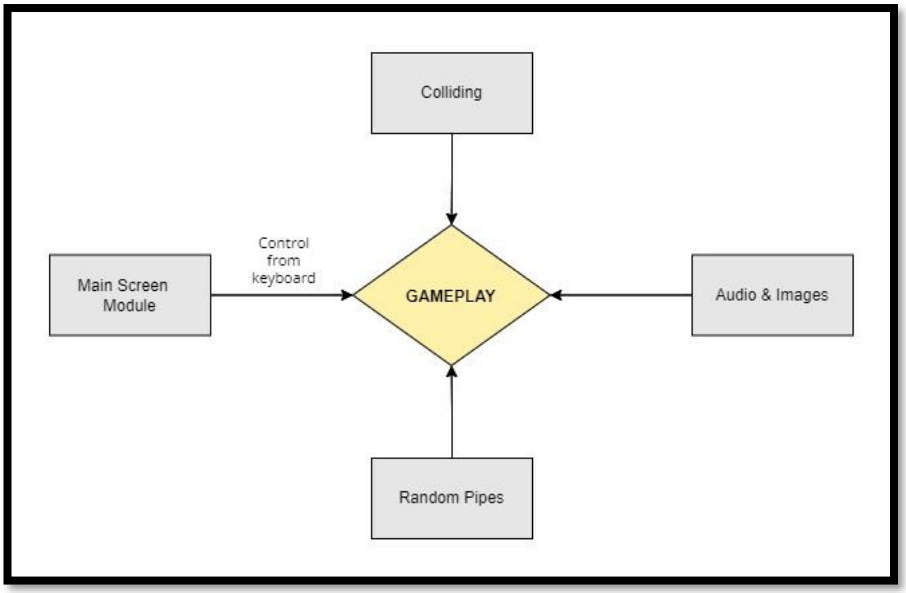


Fig 5. Modules and their Integration

5 Technologies to be used

5.1 Software Platform

- Operating system- Windows 7 or higher versions
- Programming Language- Python 3.8 (Install Conda for python)

5.2 Hardware Platform

RAM, Hard Disk, OS, Editor, Browser etc.

5.3 Libraries

- **Pygame:** This project was done with “Pygame”. Pygame is a cross-platform set of Python modules which is used to create video games. It consists of computer graphics and sound libraries designed to be used with the Python programming language. Pygame was officially written by *Pete Shinners* to replace PySDL. Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

- **Random:** Python Random library is an in-built module of Python which is used to generate random numbers. These are pseudo-random numbers means these are not truly random. This module can be used to perform random actions such as generating random numbers, print random a value for a list or string, etc. It contains function like, seed, getstate, setstate, randint, etc.

- **Sys:** The sys library in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter. Features of sys modules are, sys.argv, sys.exit, etc.

6 Advantages of this Project

- **Make it really hard**

  Of course, the first thing you notice when playing *Flappy Bird* is how hard it is. Some of the team at Tech in Asia haven’t even flown past the first pole. It’s so hard that it’s ridiculously frustrating, annoying, and somehow existentially hilarious. It’s so hard it’s funny. In this way, *Flappy Bird* is punishing. If you take a gander at the reviews on the App Store, you’ll immediately see how much people love to hate it. And this is probably the single biggest reason for its success. People love to torture themselves, and they love to share it. The difficulty of the game is engineered for virality.

- **Make it extremely simple**

  Flappy Bird has almost no barrier to entry. *Angry Birds*, *Clash of Clans*, and other big-name mobile games take minutes on the dollar for gamers to learn how to play. And for every second you have to train a new user how to play, you lose another user. I can hand anybody my iPhone on the street and they will, within one second, know how to play *Flappy Bird*. That means anybody can play and anybody will play. And this is somehow perfectly balanced with the difficulty of the game. It's easy to learn but it's near impossible to master. *Flappy Bird* is in the sweet spot of learning curves.

- **Make it feel almost winnable**

  Probably the worst, and therefore best, part about *Flappy Bird* is that it always gives you a feeling that you could have done better than last time. With the case of Angry Birds, I'm usually satisfied even if I just get two stars on a level. The feeling is "at least I passed it". With *Flappy Bird*, there is no such relief. It is essentially one level. And each point just tells you how much you suck at it.

- **No in-app purchases**

  This is probably the most deceptively compelling aspect of *Flappy Bird*. Most of the top apps these days are all about in-app purchases. You open up a new app wincing at the inevitable fee, paywall, ad, or random in-app purchase that will beautify your character or make the level easier for you. *Flappy Bird* has none of this. Yes, it's littered with ads, but the ads are placed in the game as if to hamper your gameplay. It adds another difficulty level. In other words, it's totally free. It breaks the chains of greedy game designers. And it's exactly what the doctor ordered. Users don't want to have to pay anymore. They're sick of it.

- **No more than three buttons**

  If you notice, when playing *Flappy Bird*, there's only ever three buttons on screen. In total, for the whole game, there's Start, Score, Rate, Pause, Tap, Okay, and Share. That's seven buttons. It's part of Nguyen's dedication to simplicity and also what makes the game so viral. In other games, users will be easily overwhelmed by the options games throw at you. Would you like to take a screenshot? Would you like to share points with your close friends? Would you like to look at the leaderboards? Would you like to get a hint for this level? The mechanics of the game make for a game that is instantly replayable and immediate.

- **Rely on gravity**

  If you haven’t seen the movie Gravity, you may not know what I mean here. But the physics of *Flappy Bird* are visceral. You can feel yourself nosediving along with Flappy at a rate of 9.8 meters per second squared. You can feel for the little guy. With *Angry Birds*, you don’t feel much for the birds, nor the pigs. It’s the same for the little characters in *Clash of Clans*. They’re just part of your strategy, your tactics, or your game. When Flappy falls, you can hear your mind going “Oh damnit. I died again.”

- **It’s all about skill**

  Some people just suck at *Flappy Bird*. After a few tries, they throw their phone against the wall and they give up. They never pass the first pole and they’re done. App uninstalled. But there are shining knights on the App Store that are pushing past 50 or even 60 points in Flappy Bird. This is not a game where you can skip levels or get hints on how to solve puzzles. This is the big league. This is Flappy Bird. Only the truly skilled people will get to the top. If you pass beyond ten points, it’s because you really know how to tap your fingers. You know how to anticipate the gravity of Flappy’s world.

- **Make gameplay as brief as possible, so you get as many plays as possible**

  One of the other really compelling mechanics of *Flappy Bird* is that since it’s so hard to play and you die so easily, that means you will essentially play the same level over and over again, many times. It’s like a bad habit that you can’t shirk. The more cupcakes you eat, the more cupcakes you want. *Flappy Bird* is a dopamine fiend.

7 Future Scope and further enhancement of the Project

- We want to create the bird rotating effect when jumping or falling.
- We are working on adding a multiplayer mode. This gives users the opportunity to compete with their friends.
- Also, we can increase the game difficulty by decreasing the gap between pipes as time goes on.
- Finally, we can increase the game difficulty by making pipes move up and down as time goes on.

8 Data Flow Diagram (DFD)

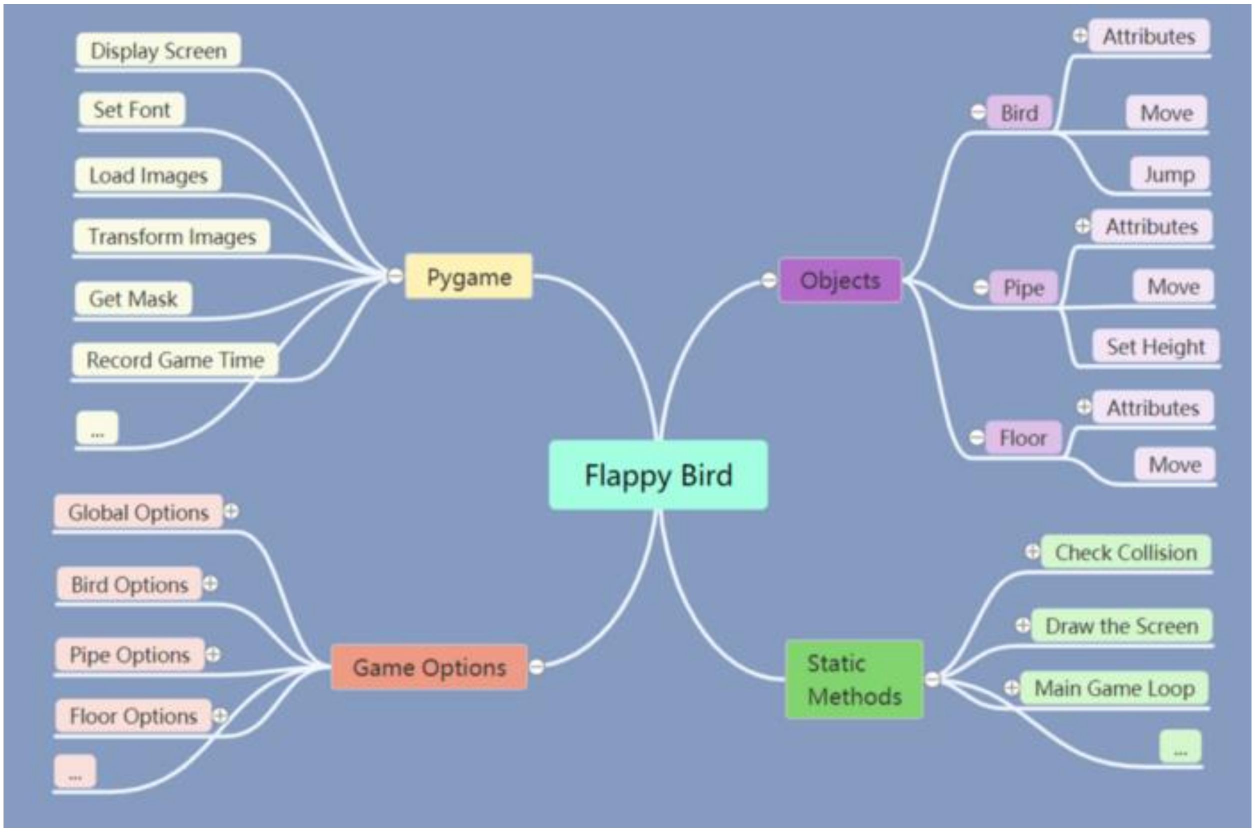


Fig 6. Data Flow Diagram

9 Entity-Relationship Diagram (ERD)

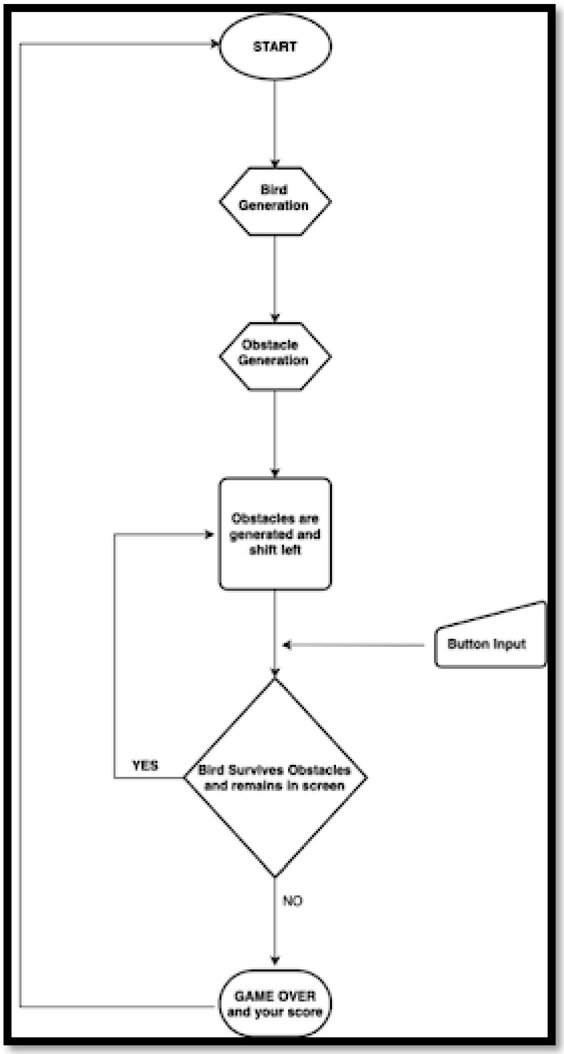

START
Bird Generation
Obstacle Generation
Obstacles are generated and shift left
Button Input
YES
Bird Survives Obstacles and remains in screen
NO
GAME OVER and your score

Fig 7. Entity Relationship Diagram

10 Use-Case Diagram (UCD)

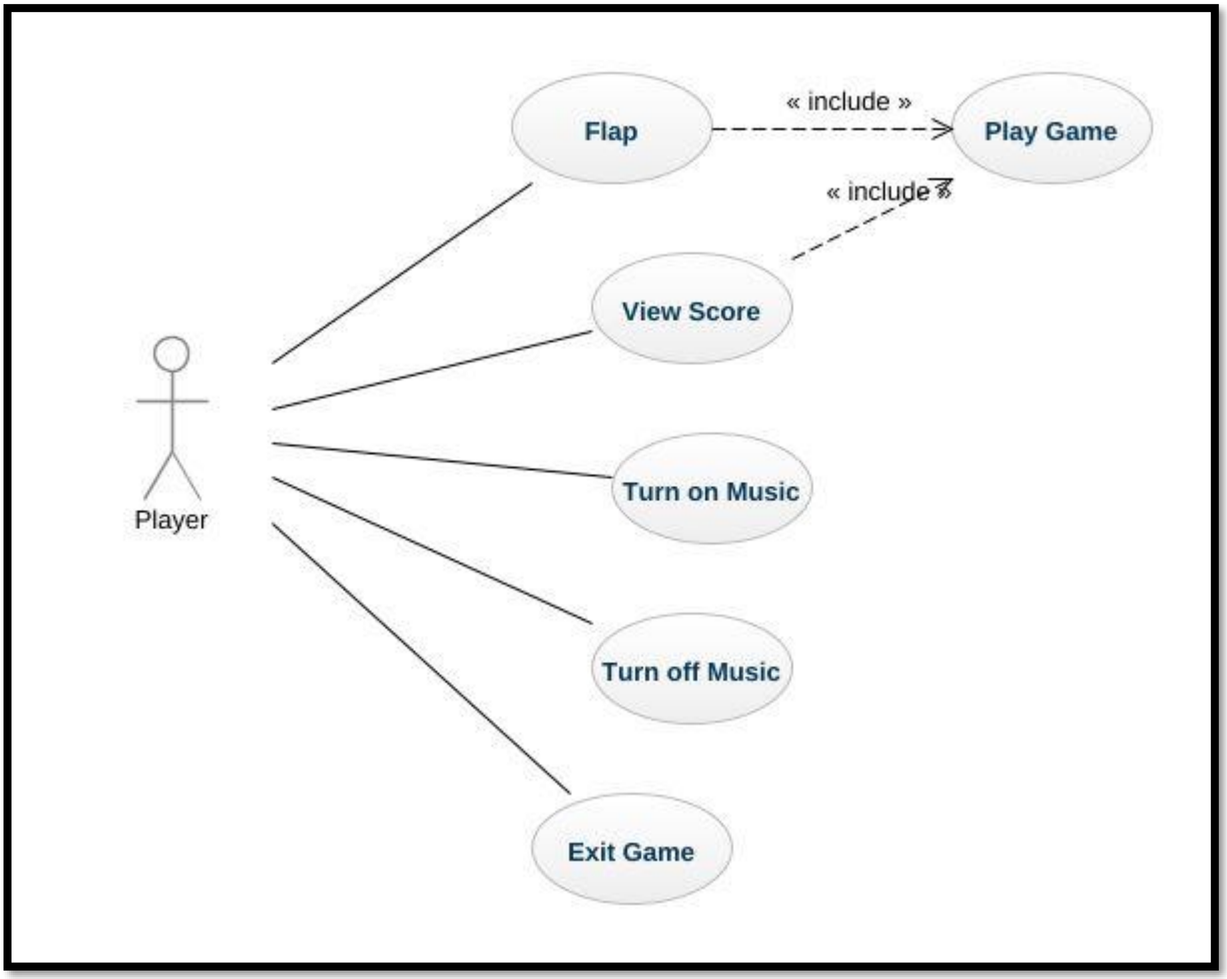


Fig 8. Use-Case Diagram

11 Screen Shots

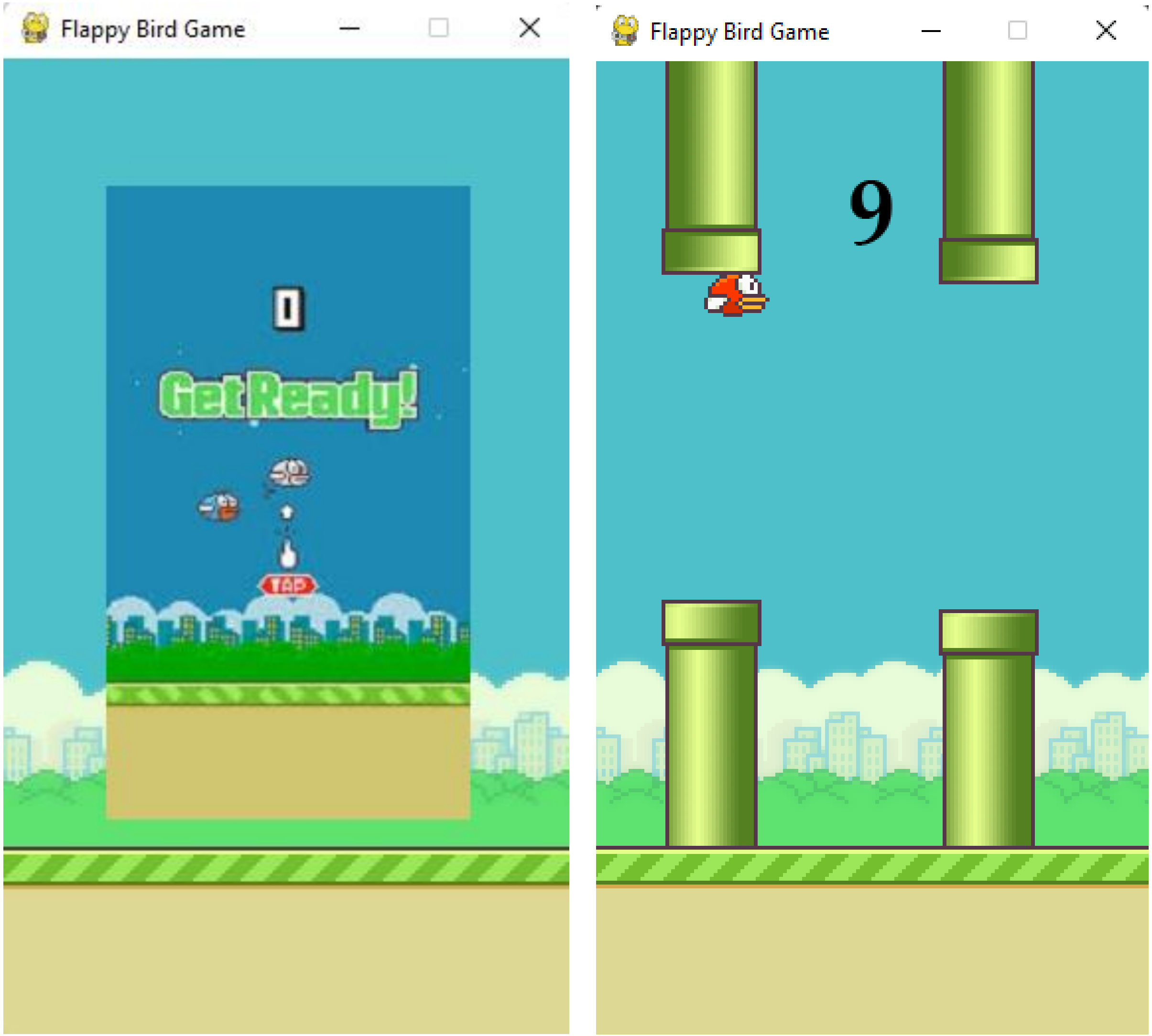


Fig 9. Screenshots of Gameplay

12 Conclusion

Our final project was planned, developed and demonstrated as expected. We designed a new version of Flappy Bird Game written in Python, which could be played on personal computers. Firstly, a user-friendly interface was implemented. and realized. It relies on gravity so User has to press Spacebar to keep the bird flying and pass tunnels, Final scores will be displayed and if the bird touches the ground or the tunnel, game restarts. It is simple yet difficult game, which will bring a lot of fun.

13 References

[1] *"Flappy Bird Game". August 18, 2020 by adones evangelista*

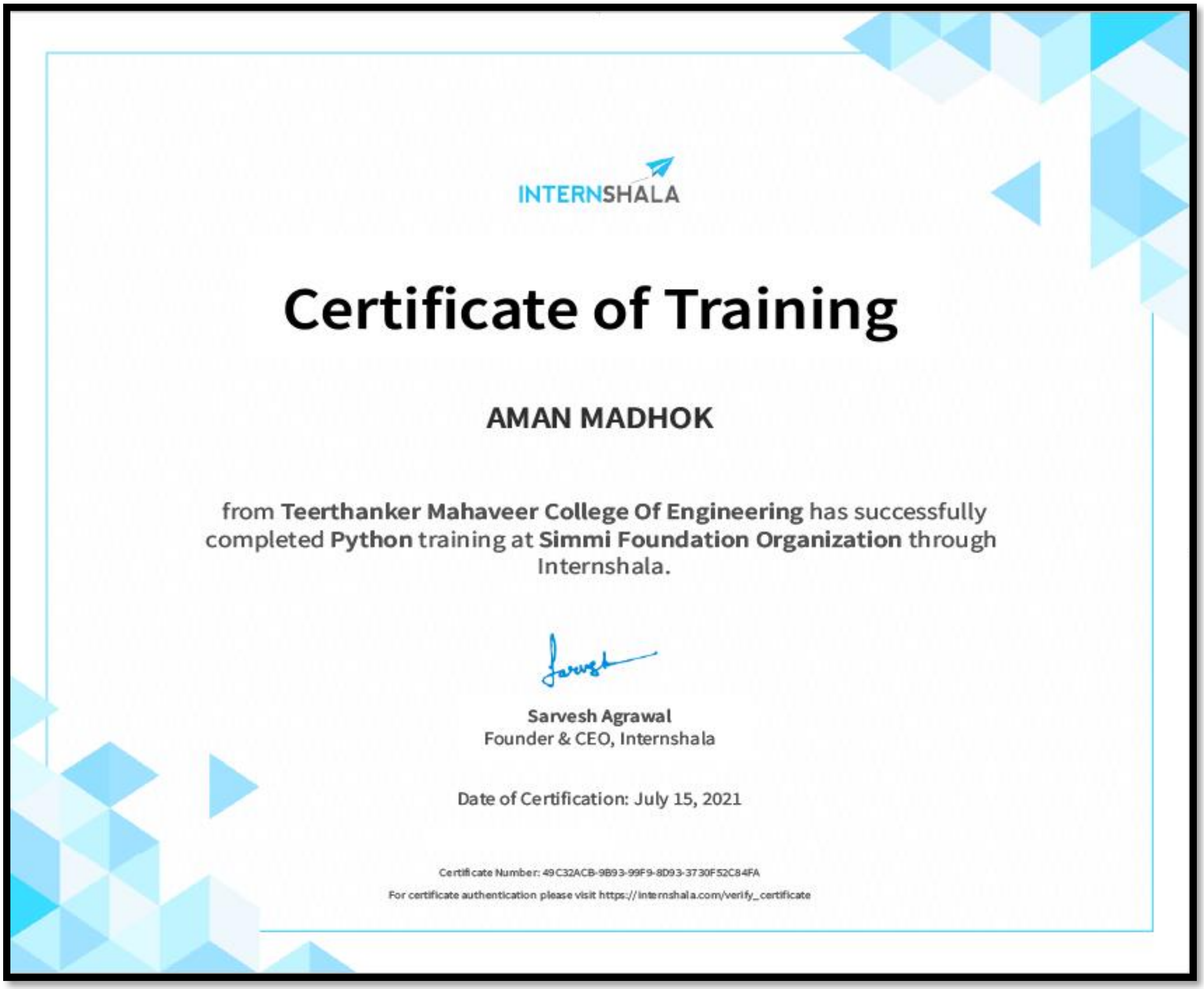
[2] *Flappy Bird CSEE 4840 Embedded system design*

[3] Flappy Bird (31 August 2021). GeeksForGeeks

[4] https://www.pygame.org/news

[5] *https://www.askpython.com/python*

14 Certificates



INTERNSHALA

Certificate of Training

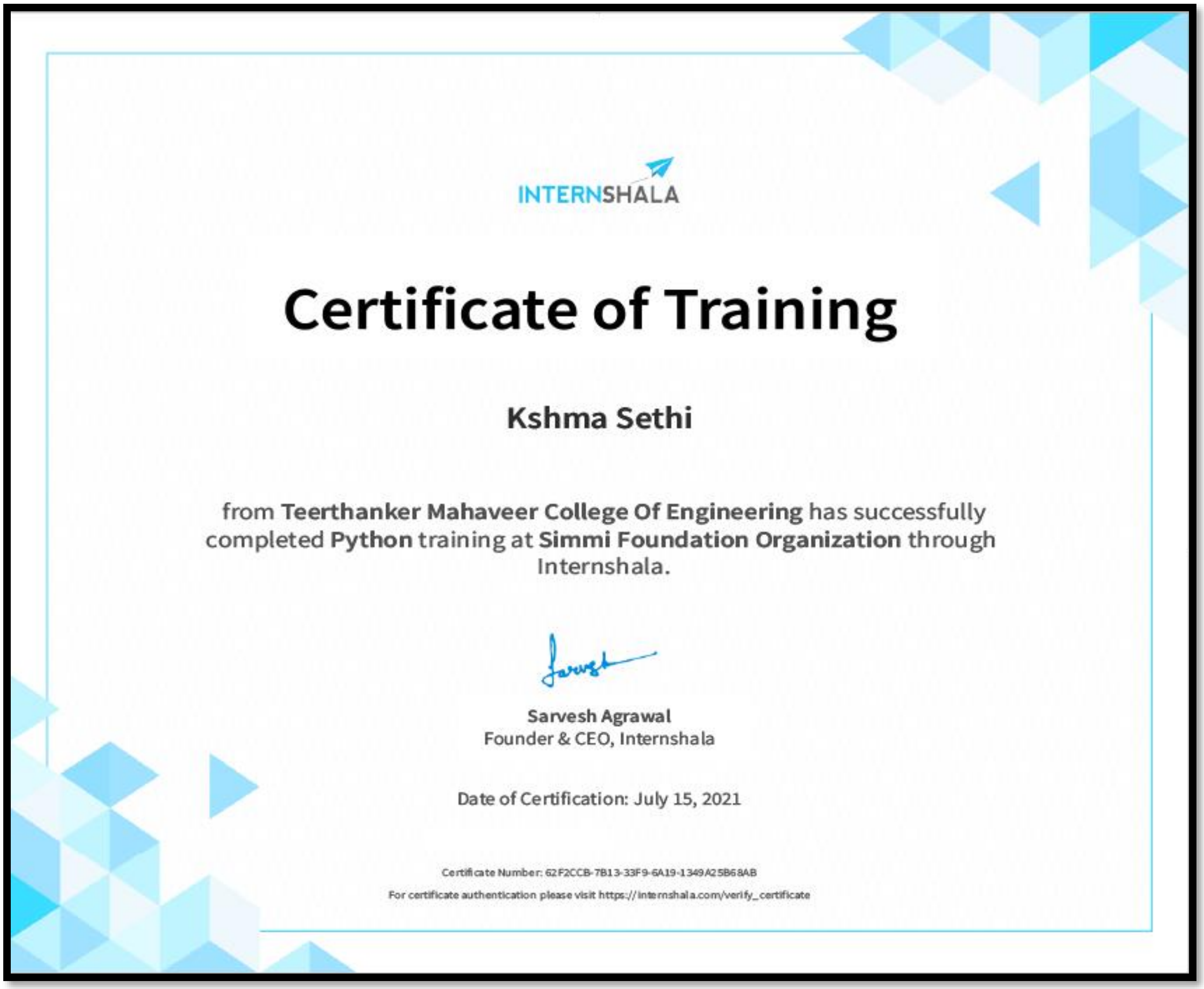
AMAN MADHOK

from **Teerthanker Mahaveer College Of Engineering** has successfully completed **Python** training at **Simmi Foundation Organization** through Internshala.

Sarvesh Agrawal
Founder & CEO, Internshala

Date of Certification: July 15, 2021

Certificate Number: 49C32ACB-9B93-99F9-8D93-3730F52C84FA
For certificate authentication please visit https://internshala.com/verify_certificate



INTERNSHALA

Certificate of Training

Kshma Sethi

from **Teerthanker Mahaveer College Of Engineering** has successfully completed **Python** training at **Simmi Foundation Organization** through Internshala.

Sarvesh Agrawal
Founder & CEO, Internshala

Date of Certification: July 15, 2021

Certificate Number: 62F2CCB-7B13-33F9-6A19-1349A25B68AB
For certificate authentication please visit https://internshala.com/verify_certificate