



TOOLS AND TECHNIQUES LAB PROJECT REPORT

Topic – Flappy Bird Game

GROUP - 03

Members:-

ARYAN DEV – 2005716

SHREST RAJ (2005759)

UTKARSH GARG (2005348)

SHOBHIT SHUKLA (2005611)

PRATYUSH SRIVASTAVA (2005740)

Submitted to

KIIT Deemed to be University

**BACHELOR'S DEGREE IN
COMPUTER SCIENCE**

UNDER THE GUIDANCE OF

AJAY ANAND

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

April 2023



TABLE OF CONTENTS

Sl. No.	Section	Topics	Pages
1.	List of figures		3
2.	Abstract		4
3.	Description of Reference project		6
4.	Proposed Enhancements		8
5.	Libraries used		13
6.	Work done		15
7.	Results		19
8.	Conclusion		22
9.	Reference		23

LIST OF FIGURES

1. Section 1: Abstract
2. Section 2: Description of reference project
 - 2.1. Fig 1.1: Initial Start game screenshot
 - 2.2. Fig 1.2: Screenshot while playing
3. Section 3: Proposed Enhancements
4. Section 4: Libraries used
5. Section 5: Libraries used
6. Section 6: Work done
7. Section 7: Results
 - 7.1. Fig 2.1: Initial Start of modified game screenshot
 - 7.2. Fig 2.2: Screenshot while playing the modified game
8. Section 8: Conclusion

ABSTRACT

With a primary focus on the Pygame library, the interesting and nostalgic Flappy Bird was created using Python. The game has been altered to add a number of features that increase user attractiveness and interaction.

Gamers are guaranteed an immersive, captivating gaming experience. High-quality sound effects have also been incorporated into the game's design to further improve the player's experience.

The flappy bird in the game have been updated, making it more interactive than the originals. With this modification, gamers will enjoy a fresh and engaging gameplay experience that calls for skill use and methods for getting over each level. The incorporation of the three levels within the game also gives players more difficulty to change the simple playing to something interesting.

The game's ability to keep high scores while playing is another intriguing feature. As they play the game, players can monitor their progress and try to beat their previous high score. The game also shows the top three high scores at the conclusion of each session so that players can assess how they performed in comparison to other players.

The spacebar can be used to fly the helicopter, and pressing ESC ends the game. With these controls, gamers can easily move about the game and have a more enjoyable time playing. A welcome screen is also added so that the game directly does not starts.

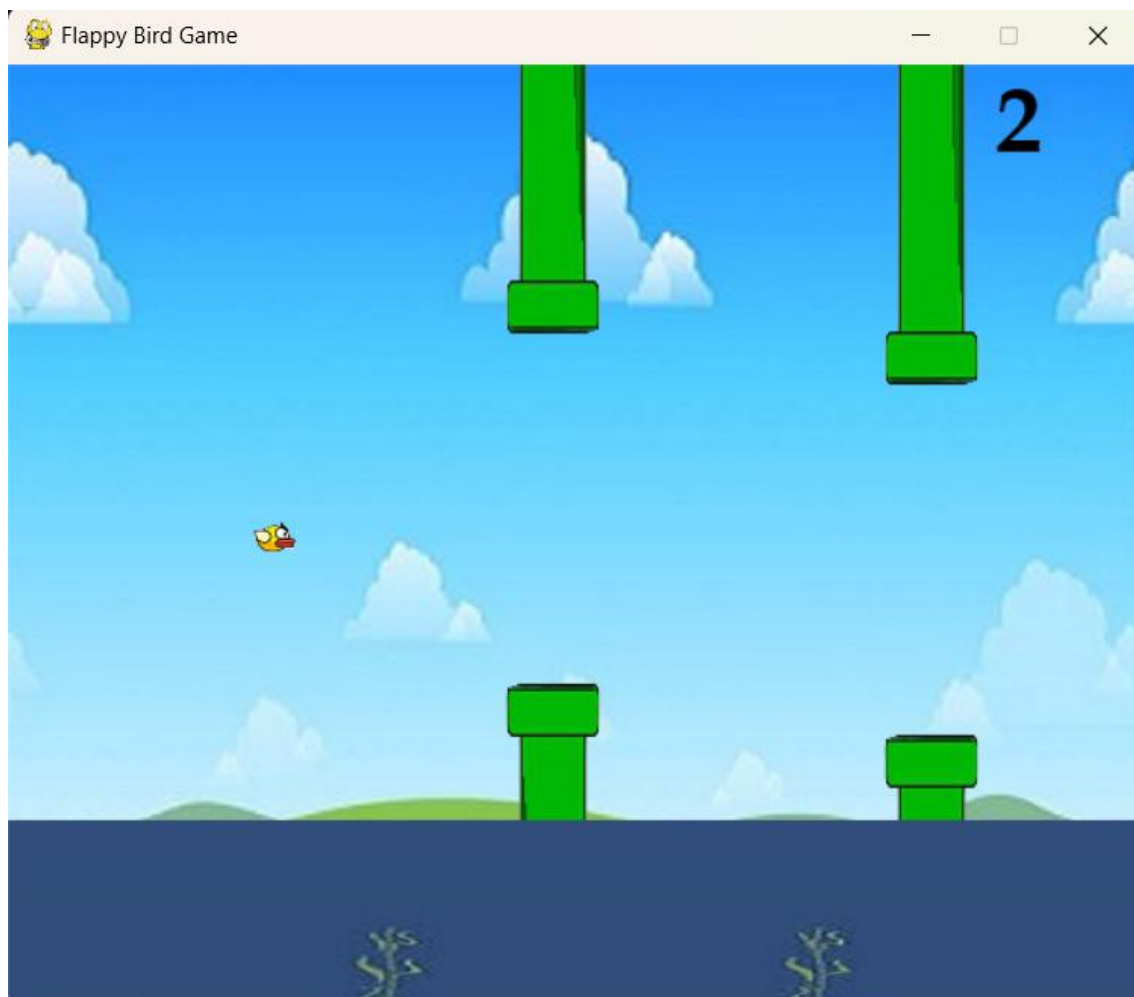
All things considered, the Flappy bird game is a superb illustration of a well-designed and entertaining game that includes a number of cutting-edge elements to offer an immersive gameplay experience. The game distinguishes itself from competing titles on the market thanks to its sound effects, helicopter, virtual background with altering day and night themes and high-score monitoring. It is a fantastic game for players of all ages and abilities, offering a thrilling and captivating experience that entices them to play again and again.

DESCRIPTION OF REFERENCE PROJECT

INTRODUCTION:-

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.

We will be using Pygame(a Python library) to create this Flappy Bird game. **Pygame** is an open-source library that is designed for making video games. it helps us to create fully functional games and multimedia programs in python.



CODE:-

The code for the Flappy game Game is written in Python using the Pygame library. It is divided into several modules, each of which contains different functions and classes.

The main module is called '**flappy.py**' and contains the game loop and the main game logic. The game loop is responsible for updating the game state and rendering the graphics on the screen. It contains total code with all the functions to run the game.

Audio folder – It contain all the sounds which are executed during the game as wav file which are small sound effects.

Images folder – It contains all the images which are seen during the playing of the game and some files are png while some are jpg.

Leaderboard.txt – It contains the scores of all the people who have played the game which their name as well as their score separated by a comma.

FUNCTIONS:-

The Flappy Bird Game includes several functions that are used throughout the game. These functions include:

- The **max_num_in_file()** function counts the top 3 scores from the leaderboard.txt file and prints the leaderboard.
- The **welcomeScreen()** function displays an interactive pop up which indicates how the game will look like and it indicates to press spacebar to start the game.
- The **flappygame()** function is the function where all the conditions are embedded to run the game.
- The **isGameOver()** function checks if the user's game is over then it restart the game again if pressed spacebar.
- The **createPipe()** function creates the pipes up and down the screen in a random height maintaining the decoram.
- The **sound()** function is the calling of background sound if any activity occurs in the game.

CLASSES:-

The Flappy Bird Game includes no classes that are used throughout the game.

FEATURES:-

The Flappy Bird Game includes several features that make it an engaging and enjoyable game to play. These features include:

- Player movement: The player should be able to fly up using the spacebar keys.
- Pipes arrangement: The pipes arranged in parallel manner randomly so that there would always be a certain space for the player to pass between it.
- Score tracking: The player's score should increase each time a pipe is passes.
- Welcome screen: When the player starts the game a "Welcome Screen" screen is displayed which indicates the player to press spacebar if he/she wants to start the game.
- Sound effects: Sound effects can be added to enhance the player's experience, such as passing the pipe, scoring, losing etc.
- Difficulty levels: Increasing the speed of the flappy bird into three different levels can create more challenging gameplay.
- Virtual background: The background changes at each level and even it changes from day to night
- Bird and Pipe change: bird is changed into Helicopter and the pipes are changed to buildings.
- Leader board: All the scores ever played are scored externally to a text file and top three scores are displayed.

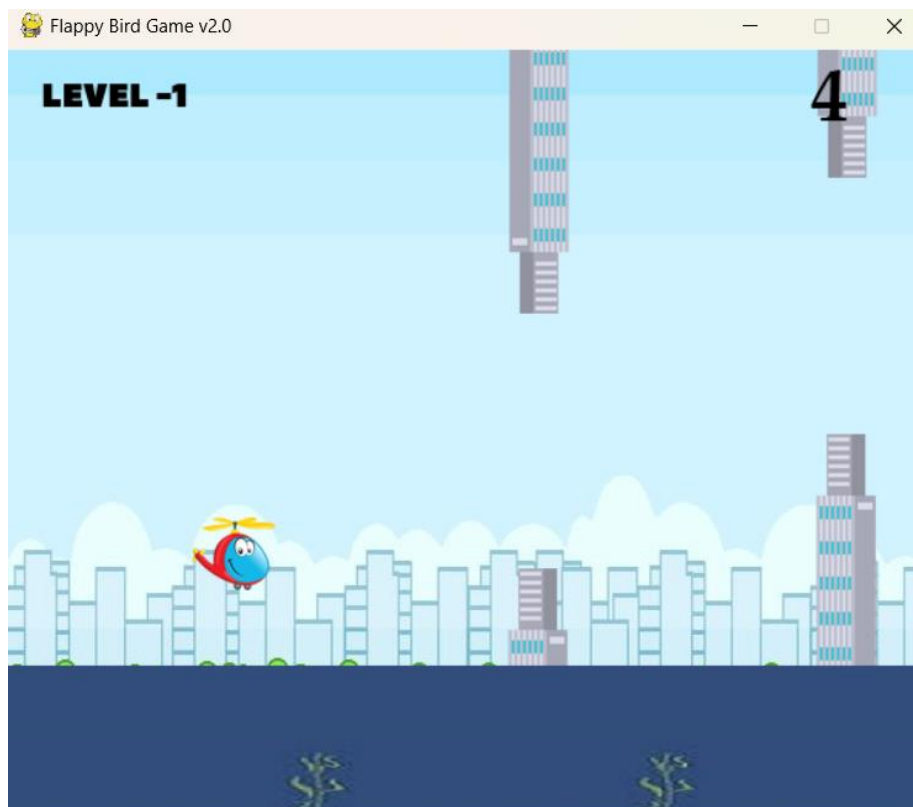
PROPOSED ENHANCEMENTS :-

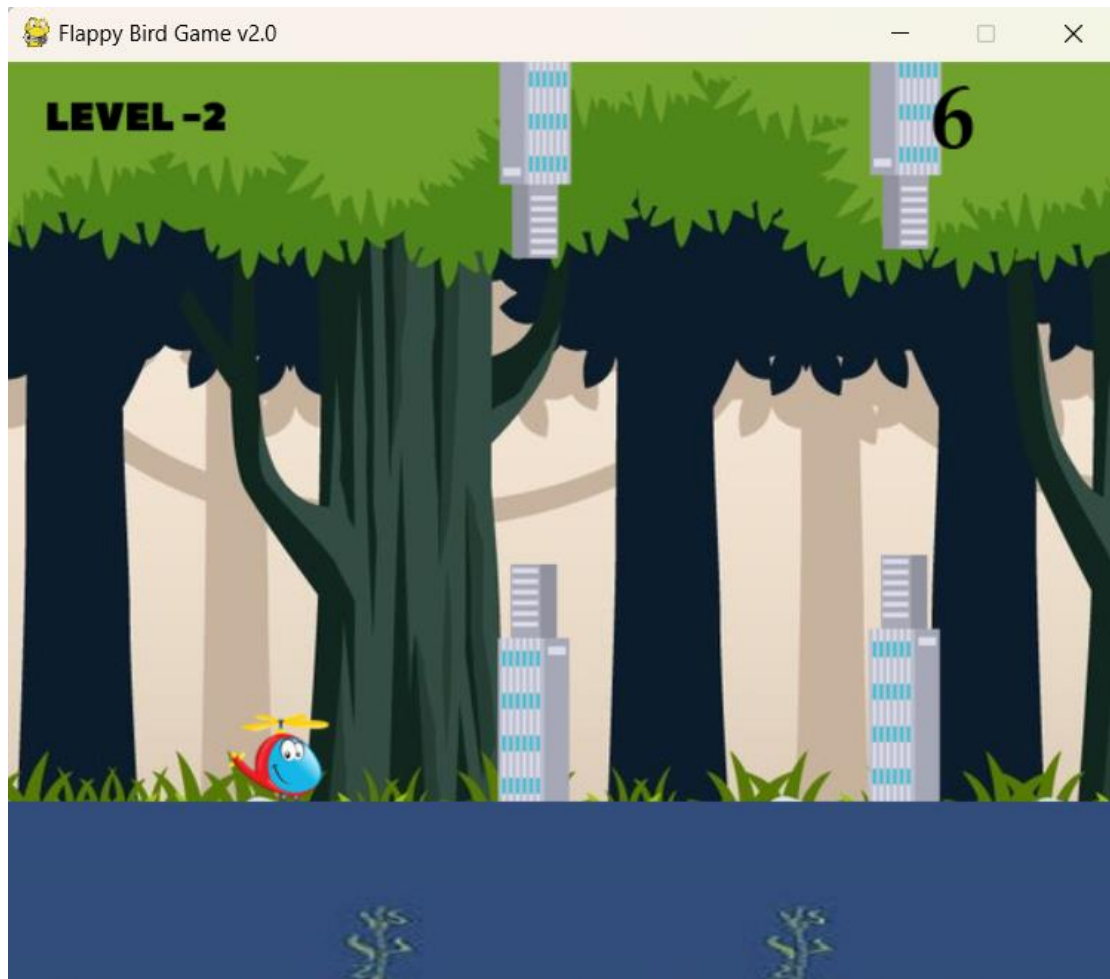
1. **Adding feature to compare all players and display the top 3 scorers** ever played this game using basic file handling concepts in python and sorting methods with appropriate key values.

```
Your score is 7

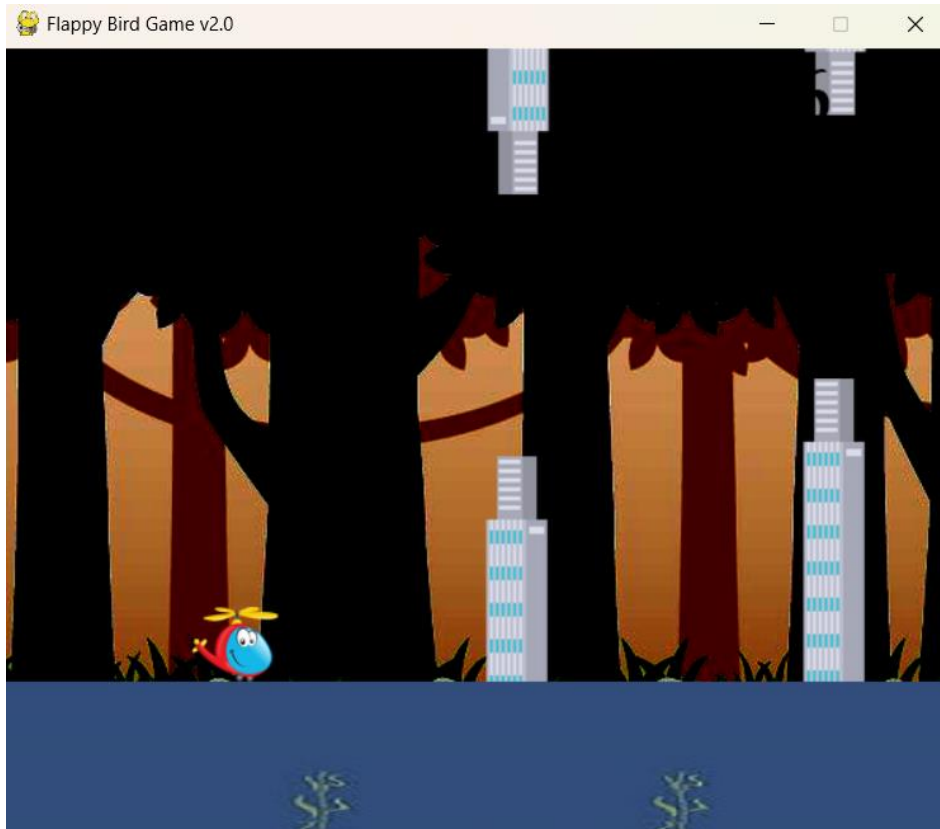
--LEADERBOARD--
-Top 3 scores are-
NAME      SCORE
anakin    9
ashoka    8
Aryan     7
PS C:\Users\aryan\OneDr
```

2. **Adding 3 levels of different difficulty** because the game has been always too easy for players, so that there will be real competition this is achieved by changing the fps value level by level.

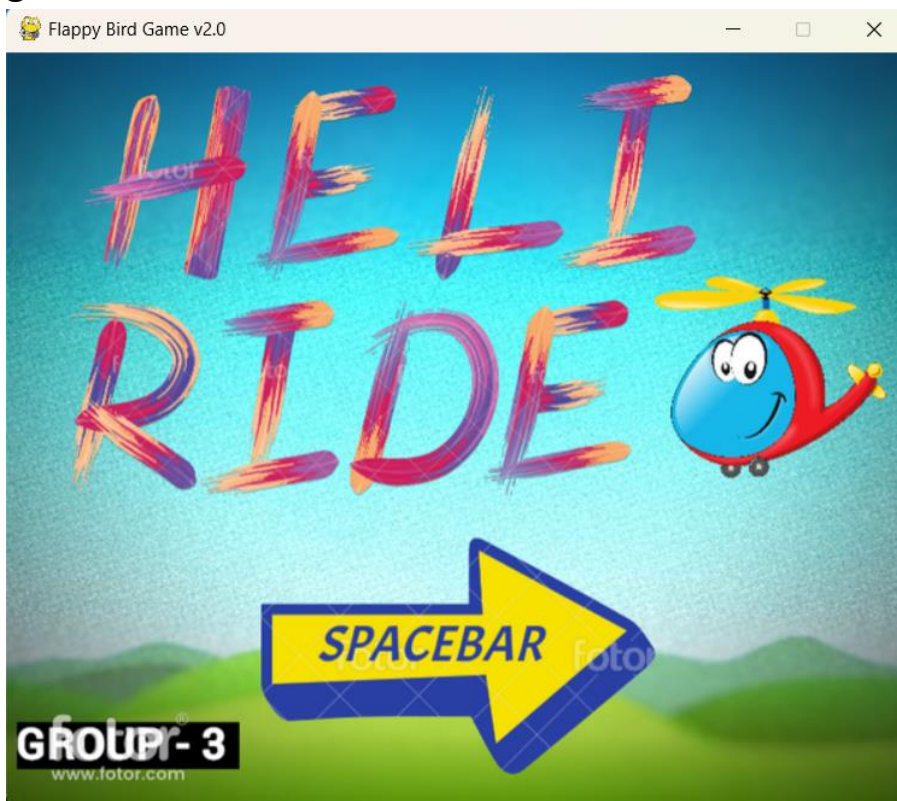




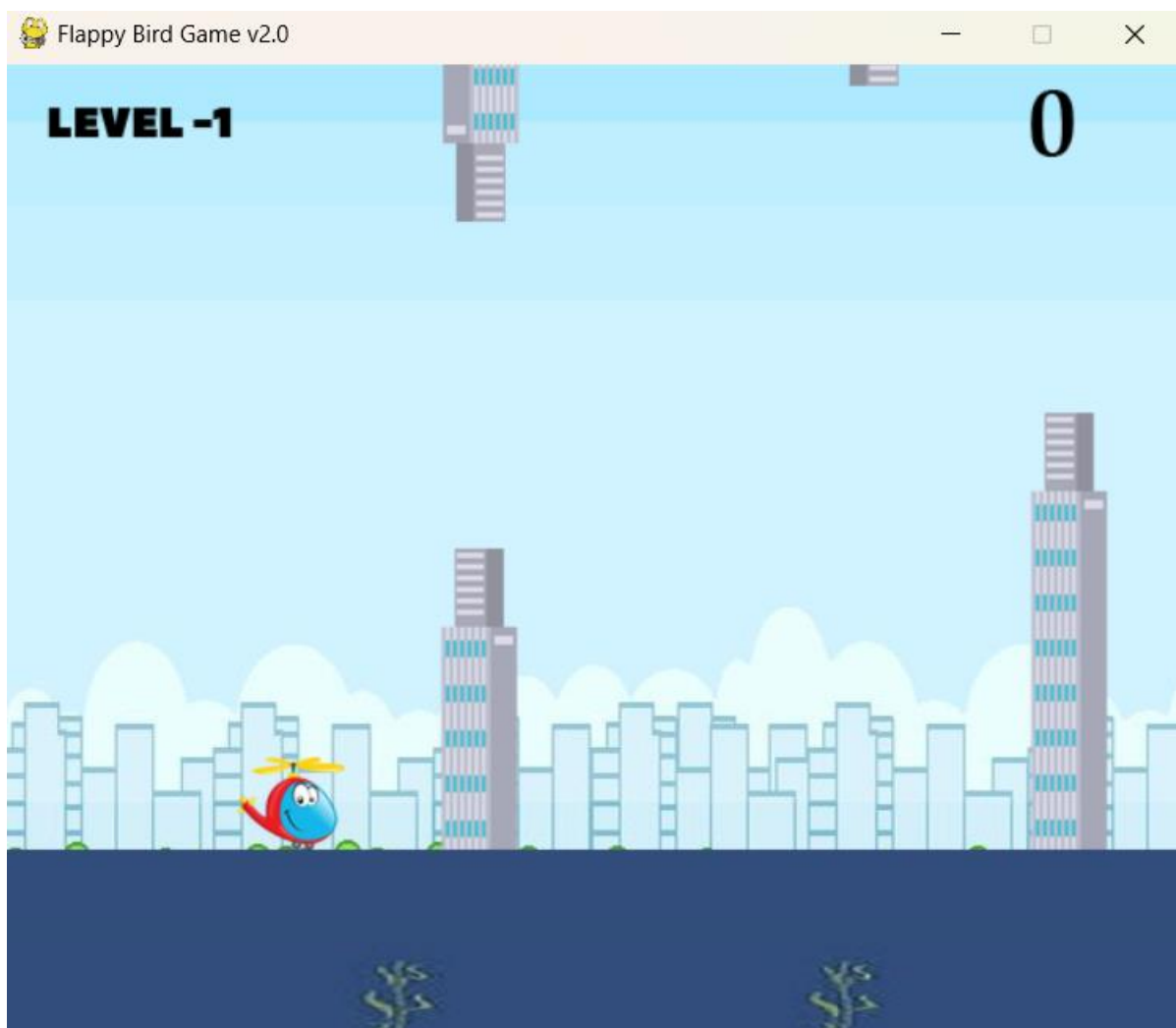
3. Building a very **cool virtual environment**(like changing themes **from day to night** or changing the whole background itself) this can be achieved by using the tint function to change the alpha values of the photo that we do in image processing.



4. **Adding a welcome screen** to the game where user can use spacebar to start the game instead of directly starting the main game.



5. **Adding sound to all the activities** in the game as the game is too boring without any sound and make it very fun and interesting.
6. **Also changed the bird's architecture** to helicopter as well as pipes to buildings for enhancing the design interface of the game as a whole.



.LIST OF LIBRARIES

Pygame:

Pygame is a popular Python library used in the development of the Flappy Bird game. It provides a range of functions for creating and managing game objects such as images, sounds, and text.

In Flappy Bird, Pygame is used to load and display images for the game objects such as the bird, the pipes, and the background. It also handles user input from the keyboard or mouse to control the bird's movement and manages the collision detection between the bird and the pipes.

Pygame also allows developers to add sound effects and background music to their games. In Flappy Bird, sound effects are used for actions such as the bird flapping its wings, hitting a pipe, or passing through a gap in the pipes.

Overall, Pygame provides a powerful and flexible framework for developing games like Flappy Bird in Python.

Random:

We have used this random library of python to generate the random integers while creating the pipes in which we give some random integers from the range of the offset to the calculated screen height-base-1.2*offset.

So for either the up pipe or the down pipe the calculation of the y coordinate of the pipes requires the generation random integer values between the range as calculated mathematically like in the above example of screen height-base-1.2*offset.

In Flappy Bird, the randint() function is used to generate a random

position for the gap between the pipes, which is then used to position the pipes on the screen.

This helps to create a sense of unpredictability and challenge for the player.

WAND LIBRARY:-

The Wand library is an open-source software package used for image manipulation in Python. It is built on top of the ImageMagick command-line tool and provides an easy-to-use interface for performing a variety of image processing tasks, such as cropping, resizing, rotating, and applying filters to images.

The `tint()` function in the Wand library is used to apply a color tint to an image. This function takes two arguments: a color and a level of intensity. The color argument specifies the color of the tint to be applied, while the intensity argument determines how strongly the tint should be applied to the image.

So here using the `tint()` function we have changed the theme of the flappy bird game from day to night through addition of the filters on the images passed in the input.

WORK DONE:-

CHANGE THE ALREADY INPUTED IMAGE TO NIGHT VERSION AND SAVE IT

```
with Image(filename='images/back2.png') as img:
    # tinted image using tint() function
    img.tint(color="yellow", alpha="rgb(40 %, 60 %, 80 %)")
    img.save(filename="images/back2night.png")
background_image3 = 'images/back2night.png'
```

DISPLAY THE TOP THREE SCORES WITH THEIR NAMES

```
def max_num_in_file(filename):
    """Returns the largest integer found in the file"""
    with open(filename, "r") as f:
        contents = f.readlines()
    data = []
    for line in contents:
        name,score=line.strip().split(',')
        record = {'Name': name, 'Score': score}
        data.append(record)
    #data = [int(x) for x in f.readlines()]
    def compare_score(record):
        return int(str(record['Score']).zfill(3))
    sorted_data = sorted(data, key=compare_score, reverse=True)
    print("\n\n--LEADERBOARD--")
    print("-Top 3 scores are-")
    print("NAME\t\tSCORE")
    c=0
    for each_record in sorted_data:
        print(each_record['Name'] + "\t\t" + each_record['Score'])
        c+=1
        if(c==3):
            break
```

WELCOME SCREEN

```
def welcomeScreen():
    """
    Shows welcome images on the screen
    """

    playerx = int(window_width/5)
    playery = int((window_height - game_images['flappybird'].get_height())/2)
    ground = 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:
                window.blit(game_images['background1'], (0, 0))
                window.blit(game_images['flappybird'], (playerx, playery))
                window.blit(game_images['sea_level'], (ground, elevation))
                window.blit(game_images['message'], (0, 0))
                pygame.display.update()
                framepersecond_clock.tick(framepersecond)

```

ADDING THREE LEVELS AND CHANGING THE BACKGROUND AT EACH LEVEL

```

if your_score >= scorearr[2]:
    window.blit(game_images['background3'], (0, 0))
    for upperPipe, lowerPipe in zip(up_pipes, down_pipes):
        window.blit(game_images['pipeimage'][0],(upperPipe['x'],
upperPipe['y']))
        window.blit(game_images['pipeimage'][1],(lowerPipe['x'],
lowerPipe['y']))

    window.blit(game_images['sea_level'], (ground, elevation))
    window.blit(game_images['flappybird'], (horizontal, vertical))

    framepersecond_clock.tick(framepersecond + 32)
    # Fetching the digits of score.
    #numbers = [int(x) for x in list(str(your_score))]
    #width = 0

    # finding the width of score images from numbers.
    for num in numbers:
        width += game_images['scoreimages'][num].get_width()
        Xoffset = (window_width - width)/1.1

    # Blitting the images on the window.

```



```

        for num in numbers:
            window.blit(game_images['scoreimages'][num],
                        (Xoffset, window_width*0.02))
            Xoffset += game_images['scoreimages'][num].get_width()
        pygame.display.update()
    elif your_score >= scorearr[1]:
        window.blit(game_images['background2'], (0, 0))
        for upperPipe, lowerPipe in zip(up_pipes, down_pipes):
            window.blit(game_images['pipeimage'][0], (upperPipe['x'],
upperPipe['y']))
            window.blit(game_images['pipeimage'][1], (lowerPipe['x'],
lowerPipe['y']))
        window.blit(game_images['sea_level'], (ground, elevation))
        window.blit(game_images['flappybird'], (horizontal, vertical))

    framepersecond_clock.tick(framepersecond + 16)
    # Fetching the digits of score.
    # numbers = [int(x) for x in list(str(your_score))]
    # width = 0

    # finding the width of score images from numbers.
    for num in numbers:
        width += game_images['scoreimages'][num].get_width()
        Xoffset = (window_width - width)/1.1

    # Blitting the images on the window.
    for num in numbers:
        window.blit(game_images['scoreimages'][num],
                    (Xoffset, window_width*0.02))
    Xoffset += game_images['scoreimages'][num].get_width()
    pygame.display.update()
else:
    pygame.display.update()
    framepersecond_clock.tick(framepersecond)

```

ADDING BACKGROUND SOUND

```

def sound():
    game_sounds['hit'] = pygame.mixer.Sound('audio/hit.wav')
    game_sounds['point'] = pygame.mixer.Sound('audio/point.wav')
    game_sounds['swoosh'] = pygame.mixer.Sound('audio/swoosh.wav')
    game_sounds['wing'] = pygame.mixer.Sound('audio/wing.wav')
    game_sounds['die'] = pygame.mixer.Sound('audio/die.wav')

```

ADDING MORE IMAGES

```

game_images['background1'] = pygame.image.load(

```

```
background_image1).convert_alpha()  
game_images['background2'] = pygame.image.load(  
    background_image2).convert_alpha()  
game_images['background3'] = pygame.image.load(  
    background_image3).convert_alpha()
```

SAVING NAME AND SCORE IN TXT FILE

```
if game_over:  
    with open('leaderboard.txt', 'a') as f:  
        f.write(str(nameofplayer) + ',' + str(high_score)+'\n')  
        f.close()  
    with open('leaderboard.txt', 'a') as f:  
        max_num_in_file('leaderboard.txt')  
    return
```

RESULTS:-

Depending on the game's goal and target audience, the Flappy Bird Game's potential outcomes can change. Here are some possible results:

User Engagement: The result may be greater user engagement if the game is made to be interesting and enjoyable. It's possible that players will spend more time playing the game because they enjoy it, which could result in more downloads, shares, and general popularity.

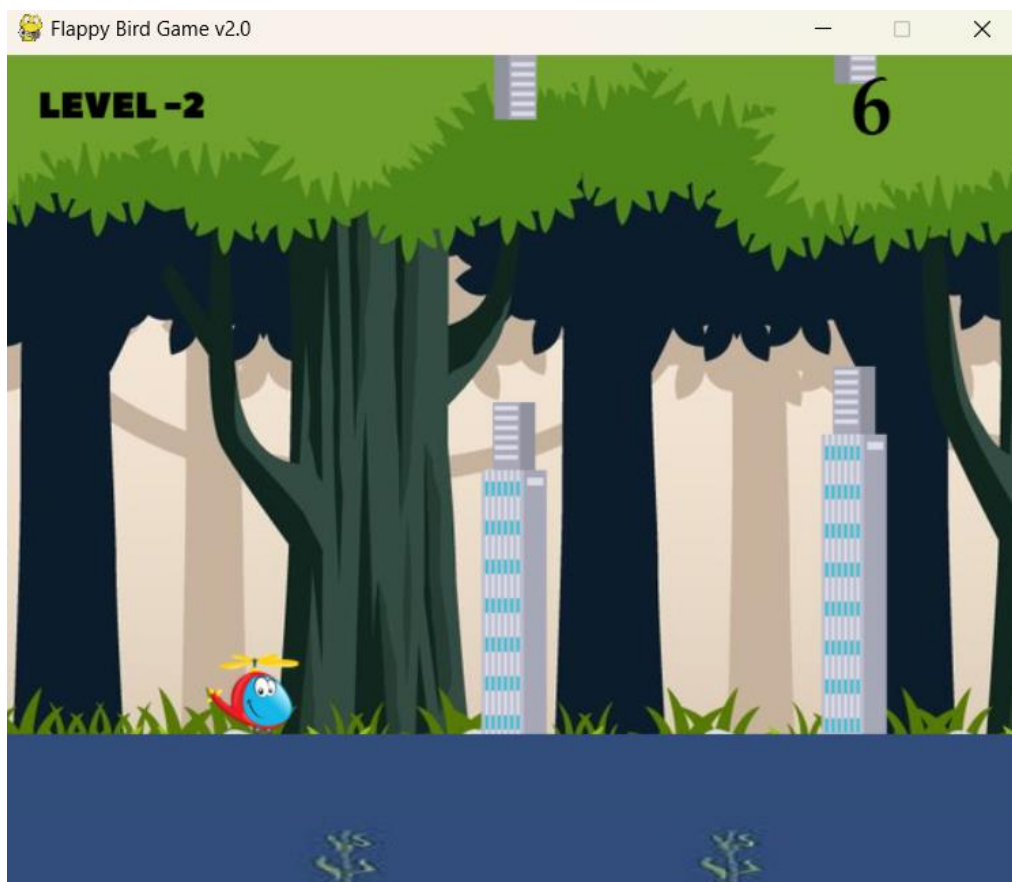
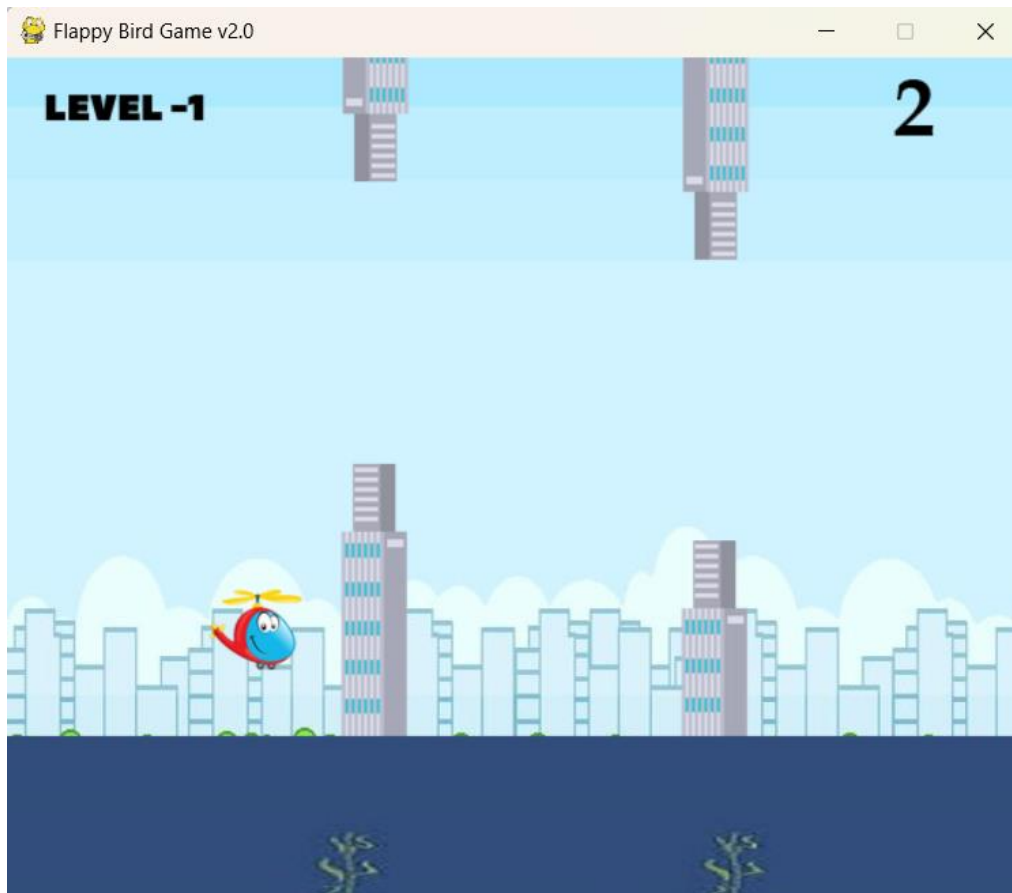
The players' skills may increase if the game is made to help them with things like hand-eye coordination, problem-solving, and strategic thinking. It's possible for players to get better at the game and use the abilities they pick up in actual life scenarios.

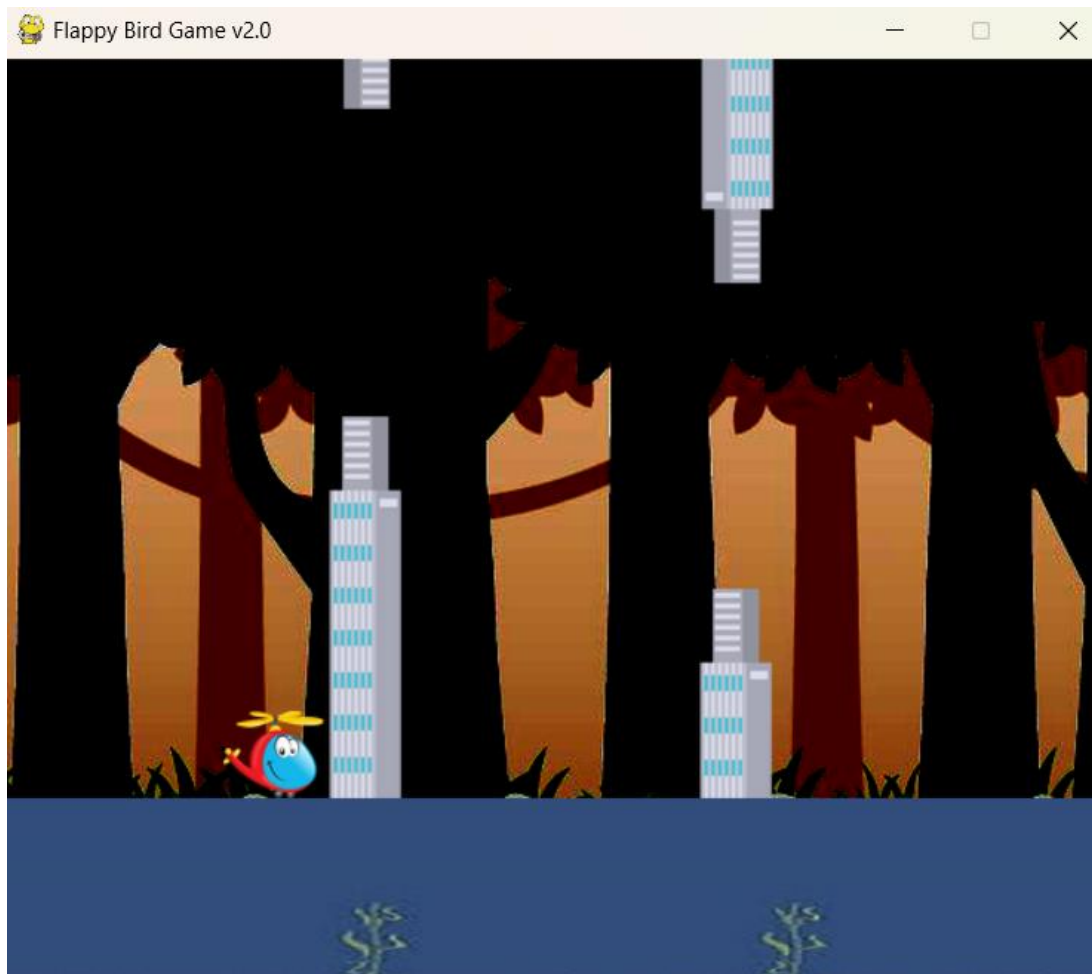
Learning: If a game is intended to instruct or enlighten participants, the results will be greater understanding and knowledge. The game can teach players things about history, physics, and other subjects.

Increased Revenue: This outcome is possible if the game is built to bring in money through in-app purchases or advertisements. A large number of players who make purchases or view advertisements could increase revenue for the game.

Increased brand awareness may be the result if the game is created as a marketing tool for a company or product. Players could get more familiar with the brand and associate the game with it.

The overall quality of the gameplay, graphics, sound, and user experience will all play a part in the game's success. a compelling game with good design





CONCLUSION:-

In conclusion, Python and Pygame were used to create the entertaining and entertaining Flappy Bird game. It features a number of improvements that have raised the difficulty and fun factor for the players. The game is visually appealing due to the customized bird and pipes, and the sound effects enhance the gameplay.

Players now have a better gameplay experience thanks to the 3 levels with virtual background in the game with changes with each approaching level and even changes from day to night. Players are encouraged to keep playing and attempt to outperform their prior scores by the game's maintenance of high scores throughout the game and the display of the top 3 high scores at the conclusion. This ups the level of competition in the game and encourages people to get better at it.

Due to the welcome screen, the game now includes a presentation screen before just starting the game. To fly through the buildings/pipes, the player must move swiftly and precisely by pressing the spacebar.

The game has been upgraded, making it more interesting and difficult for players while also improving the overall gameplay experience. Everyone who likes arcade-style games can play the game because it is appropriate for all age groups.

Learning programming fundamentals, such as object-oriented programming, game physics, and graphical programming, while creating this game with Python and Pygame is highly interactive for users. For newcomers who wish to learn how to create games using Python, it is a great project.

Overall, the Flappy Bird Game is a great illustration of a game that has been created with Python and Pygame that is entertaining and interesting. It exemplifies Pygame's adaptability as a game development toolkit and the strength of Python as a programming language. The collection of Python projects that can be used to improve programming abilities and discover new programming concepts has been greatly expanded by its inclusion.

REFERENCES:-

<https://www.geeksforgeeks.org/how-to-make-flappy-bird-game-in-pygame/>