

**BHARATI VIDYAPEETH’S INSTITUTE OF COMPUTER APPLICATIONS AND MANAGEMENT (BVICAM)**

**(GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY)**

**NEW DELHI**

**“FLAPPY BIRD”**

**MASTER OF COMPUTER APPLICATIONS**

**(M.C.A.)**

**BY**

**SAURAV GUPTA (02911604421)**

**RITIKA KUMARI (00635304421)**

**MCA – 3rd SEMESTER**

**CANDIDATE’S DECLARATION**

I hereby declare that the work which is being presented in this project work entitled “**Flappy Bird**” in partial fulfilment of the requirements for the award of the degree of **Master in Computer Applications at Bharati Vidyapeeth’s Institute of Computer Applications and Management (BVICAM), New Delhi** is an authentic record of my own work carried out during the period October 2022 to January 2023 under the supervision and guidance of **Faculty Names (Dr .Sunil Pratap Singh & Mr. Uttam Singh , BVICAM).**

I have not submitted the matter embodied in this project work anywhere for the award of any degree or diploma.

**Students’ Name :-**

**Saurav Gupta (02911604421)**

**Ritika Kumari (00635304421)**

**ACKNOWLEDGEMENT**

It is my proud privilege to express my profound gratitude to the entire management of Bharati Vidyapeeth’s Institute of Computer Applications and Management and teachers of the institute for providing me with the opportunity to avail the excellent facilities and infrastructure. The knowledge and values inculcated have proved to be of immense help at the very start of my career. Special thanks to Hon’ble Founder, Bharati Vidyapeeth, Pune for having provided us an excellent infrastructure at BVICAM.

I am grateful to **Dr .Sunil Pratap Singh & Mr. Uttam Singh** (Project Guide, BVICAM, New Delhi) for his astute guidance, constant encouragement and sincere support for this project work.

I feel pride and privileged in expressing my deep sense of gratitude to all those who have helped me in presenting this assignment. I would be failing in my endeavour if I do not place my acknowledgement**.**

**Student’s Name:-**

**Ritika Kumari (00635304421)**

**Saurav Gupta (02911604421)**

Introduction

Flappy Bird is a [mobile game](https://en.wikipedia.org/wiki/Mobile_game) developed by [Vietnamese](https://en.wikipedia.org/wiki/Vietnam) [video game](https://en.wikipedia.org/wiki/Video_game_artist) [artist](https://en.wikipedia.org/wiki/Video_game_artist) and [programmer](https://en.wikipedia.org/wiki/Video_game_programmer) Dong Nguyen under his game development company .The game is a [side-scroller](https://en.wikipedia.org/wiki/Side-scrolling_video_game) where the player controls a bird, attempting to fly between columns of green pipes without hitting them. The game was released in May 2013 but received a sudden rise in popularity in early 2014 and became a [sleeper hit](https://en.wikipedia.org/wiki/Sleeper_hit). At the end of January 2014, it was the most downloaded free game in the Play

store for Android .

Flappy Bird is an [arcade](https://en.wikipedia.org/wiki/Arcade_game#Arcade_genre)-style game in which the player controls the bird Flappy, which moves persistently to the right. The player is tasked with navigating Flappy through pairs of pipes that have equally sized gaps placed at random heights. Flappy automatically descends and only ascends when the player taps the [touchscreen](https://en.wikipedia.org/wiki/Touchscreen). Each successful pass through a pair of pipes awards the player one point.

Colliding with a pipe or the ground ends the gameplay.

**Prerequisite:-**

The prerequisite of this project is the basic knowledge of python.

For this project, we are going to use the PyCharm IDE and the pygame module. Pygame is a library that is used in creating games in Python. It has four important things.

* Game Loop
* Events
* Sprites
* Sound

**Must have things for this Mini Project :**

1. Install Your Favourite Python IDE (We have used Pycharm IDE)
2. Pygame should also be installed on your system .

**What is 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.

# Prerequisites for Pygame:

To learn pygame, it is required to have basic knowledge of Python.

**Pygame Installation –**

1. **Installing through pip:** The good way to install Pygame is with the pip tool (which is what python uses to install packages). The command is the following:
2. py -m pip install -U pygame --user

**2. Installing through an IDE:** The second way is to install it through an IDE and here we are using Pycharm IDE. Installation of pygame in the pycharm is straightforward. We can install it by running the above command in the terminal .

|  |  |  |
| --- | --- | --- |
|  | | |
| **About Project**  Simple Flappy Bird Game project is written in Python. The project file contains | |  |
| asset files, python scripts (flappybird.py). The gameplay Graphics is good enough and the controls are simple for the users. Talking about the gameplay, it’s one of the most addictive and played games for all. All the playing methods are too simple just like the real one. All you have to do is just try to stay in the middle of the screen until long green pipes appear in front of you. Here, the user has to control the bird flapping up, down using Spacebar, without touching pipes in order to score game points. This means the more you pass through green pipes, more will be the game points. A simple GUI is provided for the easy gameplay. The gameplay design is so simple that user won’t find it difficult to use and navigate. This simple flappy bird game provides the simplest gameplay of flappy bird. Though the gameplay is a bit laggy but the gameplay works well. In order to run the project, you must have installed [Python](https://www.python.org/downloads/release/python-365/), Pygame on your PC. This is a simple GUI Based game written for the beginners. | | |
| **Team Members with Roles** |  | |
|  |  | |

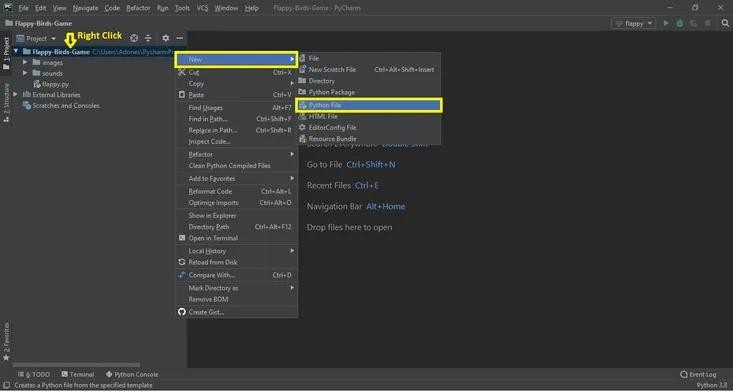
1) **Ritika Kumari** – I have done the designing part ie; all the images that are attached in this project are designed and collected by me and I have written the two function (welcomeGame() and endGame()) and implement their workings in game .

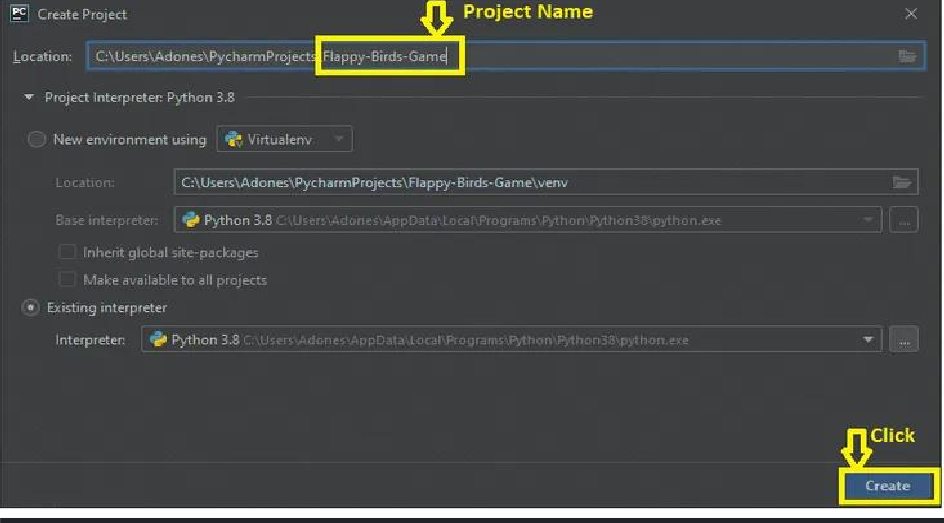
2) **Saurav Gupta -** I have collected the sounds required for this project and written remaining of the source code and implement all of the workings with synchronization of all function , sound effects and graphics.

# Steps on How to Create Flappy Bird Game in Python.

**Step 1: Create a project name.**

First, when you finished installing the **Pycharm IDE** in your computer, open it and then create a “**project name**” after creating a project name click the “**create**” button.



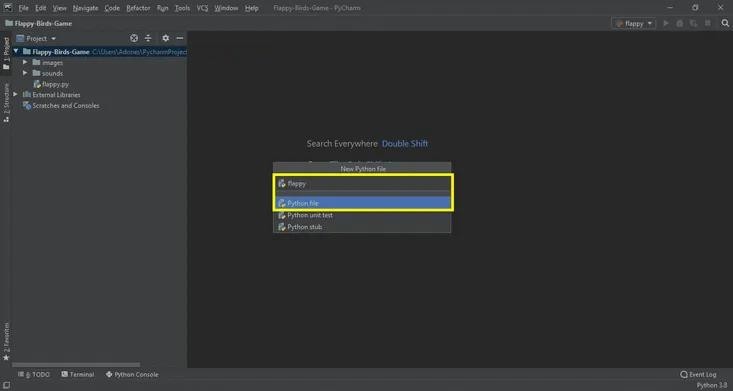


Second after creating a project name, “**right click**” your project name and then click “**new**” after that click the “**python file**“.

**Step 2: Create a python file.**

**Step 3: Name your python file.**

Third after creating a python file, Name your python file after that click “**enter**“



**Step 4: The actual code.**

This is the actual coding on how to create **Flappy Bird Game.**

# Modules Used

**Importing Random Module**

In the code given below. which is for the **random**() function, which generates **random** numbers between 0 and 1.

import random

# Importing Sys Module

In the code given below, which is for the function of sys module. We will use sys.exit to exit the program.

import sys

# Importing Pygame Module

In the code given below, which is pygame library is an open-source module for the Python programming language specifically intended to help you make games and other multimedia applications. Pygame can run across many platforms and operating systems.

import pygame

from pygame.locals import \*

# This module is for the global variables

In the code given below, which is for the global variables for the game.

FPS = 32

SCREENWIDTH = 289

SCREENHEIGHT = 511

SCREEN = pygame.display.set\_mode((SCREENWIDTH, SCREENHEIGHT)) GROUNDY = SCREENHEIGHT \* 0.8

GAME\_SPRITES = {} GAME\_SOUNDS = {}

PLAYER = 'gallery/sprites/bird.png' BACKGROUND = 'gallery/sprites/background.png' PIPE = 'gallery/sprites/pipe.png'

# This module is for the welcome main screen

def welcomeScreen():

*"""*

*Shows welcome images on the screen """*

playerx = int(SCREENWIDTH / 5)

playery = int((SCREENHEIGHT - GAME\_SPRITES['player'].get\_height()) /

2)

messagex = int((SCREENWIDTH - GAME\_SPRITES['message'].get\_width()) / 2)

messagey = int(SCREENHEIGHT \* 0.13) basex = 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:

SCREEN.blit(GAME\_SPRITES['background'], (0, 0)) SCREEN.blit(GAME\_SPRITES['player'], (playerx, playery)) SCREEN.blit(GAME\_SPRITES['message'], (messagex, messagey)) SCREEN.blit(GAME\_SPRITES['base'], (basex, GROUNDY))

pygame.display.update() FPSCLOCK.tick(FPS)

**This module is for the main gameplay**

In the code given below, which is for the function of main gameplay of a game. Create 2 pipes for blitting on the screen. Includes the list of upper and lower pipes. Checking for score function, Add a new pipe when the first is about to cross the leftmost part of the screen.

def mainGame(): score = 0

playerx = int(SCREENWIDTH / 5) playery = int(SCREENWIDTH / 2) basex = 0

# Create 2 pipes for blitting on the screen newPipe1 = getRandomPipe()

newPipe2 = getRandomPipe()

# my List of upper pipes upperPipes = [

{'x': SCREENWIDTH + 200, 'y': newPipe1[0]['y']},

{'x': SCREENWIDTH + 200 + (SCREENWIDTH / 2), 'y':

newPipe2[0]['y']},

]

# my List of lower pipes lowerPipes = [

{'x': SCREENWIDTH + 200, 'y': newPipe1[1]['y']},

{'x': SCREENWIDTH + 200 + (SCREENWIDTH / 2), 'y':

newPipe2[1]['y']},

]

pipeVelX = -4

playerVelY = -9

playerMaxVelY = 10

playerMinVelY = -8

playerAccY = 1

playerFlapAccv = -8 # velocity while flapping

playerFlapped = False # It is true only when the bird is flapping

while True:

for event in pygame.event.get():

if event.type == QUIT or (event.type == KEYDOWN and event.key == K\_ESCAPE):

pygame.quit() sys.exit()

if event.type == KEYDOWN and (event.key == K\_SPACE or event.key

== K\_UP):

if playery > 0:

playerVelY = playerFlapAccv playerFlapped = True GAME\_SOUNDS['wing'].play()

crashTest = isCollide(playerx, playery, upperPipes,

lowerPipes) # This function will return true if the player is

crashed

if crashTest: return

# check for score

playerMidPos = playerx + GAME\_SPRITES['player'].get\_width() / 2 for pipe in upperPipes:

pipeMidPos = pipe['x'] + GAME\_SPRITES['pipe'][0].get\_width() / 2 if pipeMidPos <= playerMidPos < pipeMidPos + 4:

score += 1

print(f"Your score is {score}")

GAME\_SOUNDS['point'].play()

if playerVelY < playerMaxVelY and not playerFlapped: playerVelY += playerAccY

if playerFlapped: playerFlapped = False

playerHeight = GAME\_SPRITES['player'].get\_height()

playery = playery + min(playerVelY, GROUNDY - playery - playerHeight)

# move pipes to the left

for upperPipe, lowerPipe in zip(upperPipes, lowerPipes): upperPipe['x'] += pipeVelX

lowerPipe['x'] += pipeVelX

# Add a new pipe when the first is about to cross the leftmost part of the screen if 0 < upperPipes[0]['x'] < 5:

newpipe = getRandomPipe() upperPipes.append(newpipe[0]) lowerPipes.append(newpipe[1])

# if the pipe is out of the screen, remove it

if upperPipes[0]['x'] < -GAME\_SPRITES['pipe'][0].get\_width(): upperPipes.pop(0)

lowerPipes.pop(0)

# Lets blit our sprites now SCREEN.blit(GAME\_SPRITES['background'], (0, 0)) for upperPipe, lowerPipe in zip(upperPipes, lowerPipes):

SCREEN.blit(GAME\_SPRITES['pipe'][0], (upperPipe['x'], upperPipe['y']))

SCREEN.blit(GAME\_SPRITES['pipe'][1], (lowerPipe['x'], lowerPipe['y']))

SCREEN.blit(GAME\_SPRITES['base'], (basex, GROUNDY))

SCREEN.blit(GAME\_SPRITES['player'], (playerx, playery)) myDigits = [int(x) for x in list(str(score))]

width = 0

for digit in myDigits:

width += GAME\_SPRITES['numbers'][digit].get\_width() Xoffset = (SCREENWIDTH - width) / 2

for digit in myDigits: SCREEN.blit(GAME\_SPRITES['numbers'][digit], (Xoffset,

SCREENHEIGHT \* 0.12))

Xoffset += GAME\_SPRITES['numbers'][digit].get\_width() pygame.display.update()

FPSCLOCK.tick(FPS)

# This module is for the get random pipes

In the code given below, which is for the function of get random pipes. Generate positions of two pipes(one bottom straight and one top rotated ) for blitting on the screen.

def getRandomPipe():

*"""*

*Generate positions of two pipes(one bottom straight and one top rotated ) for blitting on the screen*

*"""*

pipeHeight = GAME\_SPRITES['pipe'][0].get\_height() offset = SCREENHEIGHT / 3

y2 = offset + random.randrange(0, int(SCREENHEIGHT - GAME\_SPRITES['base'].get\_height() - 1.2 \* offset))

pipeX = SCREENWIDTH + 10

y1 = pipeHeight - y2 + offset pipe = [

{'x': pipeX, 'y': -y1}, # upper Pipe

{'x': pipeX, 'y': y2} # lower Pipe

]

return pipe

# This module is for images

if name == " main ":

# This will be the main point from where our game will start pygame.init() # Initialize all pygame's modules FPSCLOCK = pygame.time.Clock() pygame.display.set\_caption('Flappy Bird by Ujjwal Shukla') GAME\_SPRITES['numbers'] = (

pygame.image.load('gallery/sprites/0.png').convert\_alpha(), pygame.image.load('gallery/sprites/1.png').convert\_alpha(), pygame.image.load('gallery/sprites/2.png').convert\_alpha(), pygame.image.load('gallery/sprites/3.png').convert\_alpha(), pygame.image.load('gallery/sprites/4.png').convert\_alpha(),

pygame.image.load('gallery/sprites/5.png').convert\_alpha(), pygame.image.load('gallery/sprites/6.png').convert\_alpha(), pygame.image.load('gallery/sprites/7.png').convert\_alpha(), pygame.image.load('gallery/sprites/8.png').convert\_alpha(), pygame.image.load('gallery/sprites/9.png').convert\_alpha(),

)

GAME\_SPRITES['message'] = pygame.image.load('gallery/sprites/message.png').convert\_alpha()

GAME\_SPRITES['base'] =

pygame.image.load('gallery/sprites/base.png').convert\_alpha() GAME\_SPRITES['pipe'] =

(pygame.transform.rotate(pygame.image.load(PIPE).convert\_alpha(), 180), pygame.image.load(PIPE).convert\_alpha()

)

**This module is for the sound**

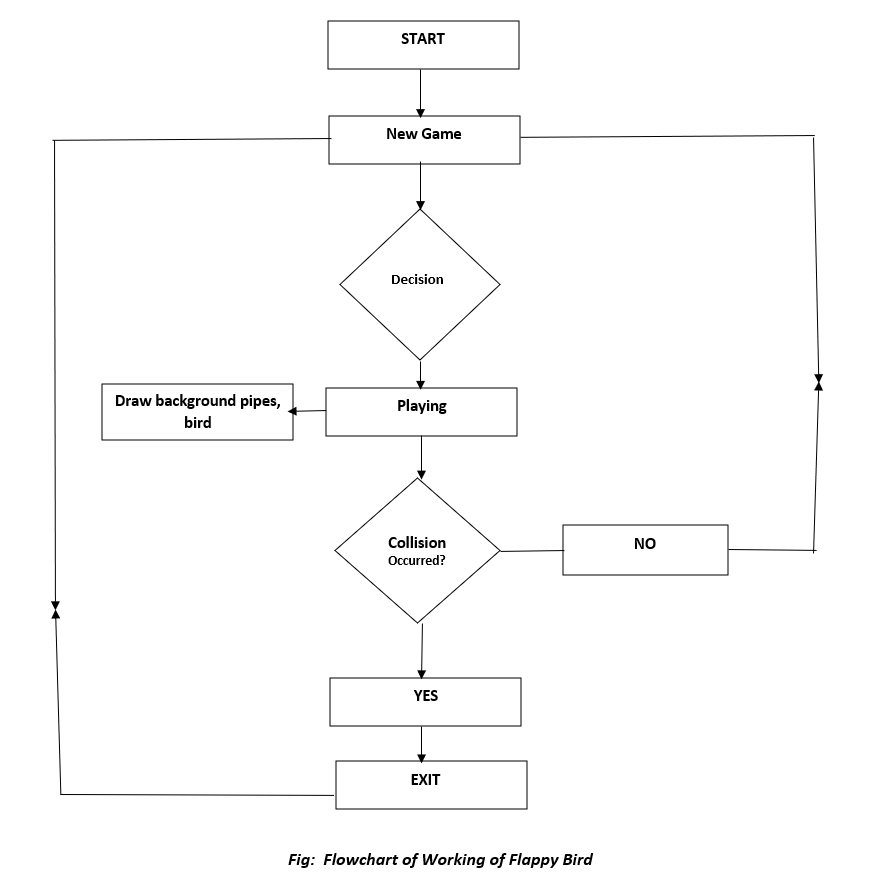
# Game sounds

GAME\_SOUNDS['die'] = pygame.mixer.Sound('gallery/audio/die.wav') GAME\_SOUNDS['hit'] = pygame.mixer.Sound('gallery/audio/hit.wav') GAME\_SOUNDS['point'] = pygame.mixer.Sound('gallery/audio/point.wav') GAME\_SOUNDS['swoosh'] = pygame.mixer.Sound('gallery/audio/swoosh.wav') GAME\_SOUNDS['wing'] = pygame.mixer.Sound('gallery/audio/wing.wav')

GAME\_SPRITES['background'] = pygame.image.load(BACKGROUND).convert()

GAME\_SPRITES['player'] = pygame.image.load(PLAYER).convert\_alpha()

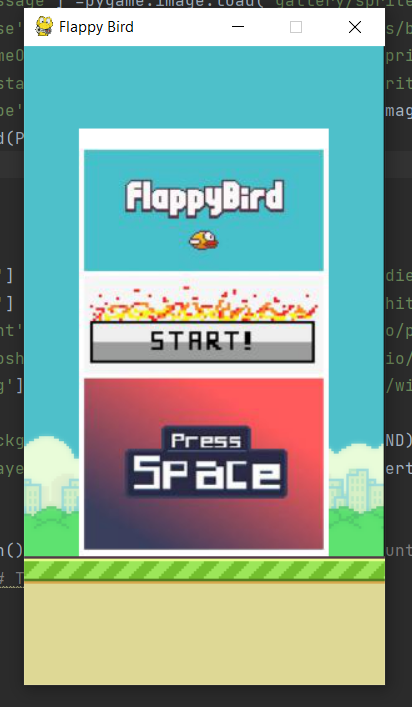
**Flowchart Of Flappy bird**

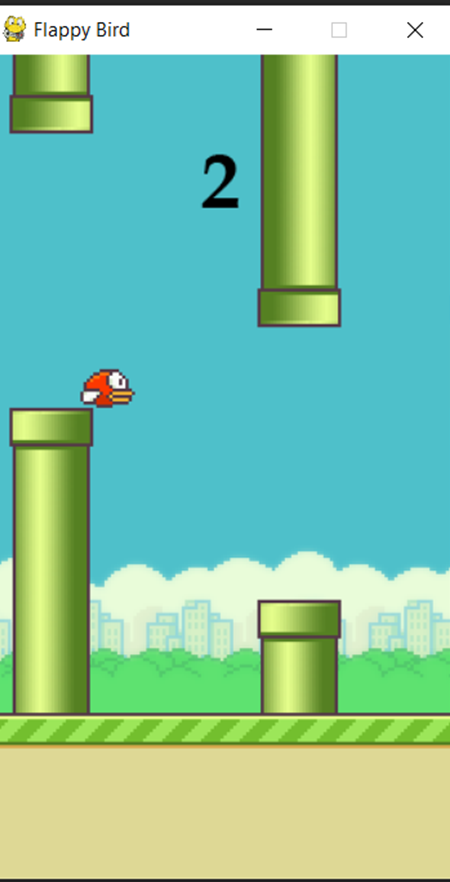
****

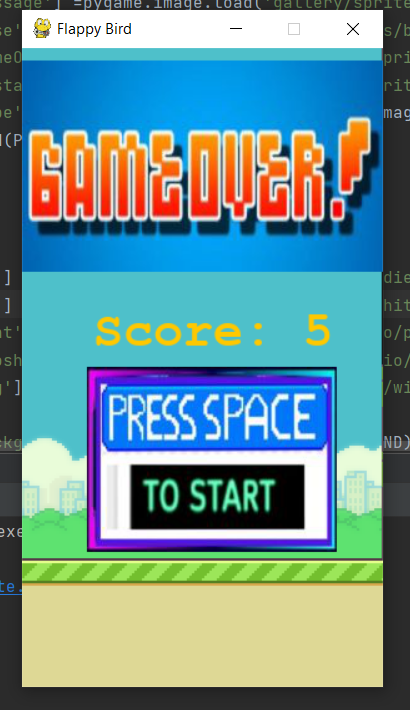
**Complete Source Code of Flappy Bird Game in Python**

import random # For generating random numbers  
import sys # We will use sys.exit to exit the program  
import pygame  
from pygame.locals import \* # Basic pygame imports  
  
# Global Variables for the game  
FPS = 32  
SCREENWIDTH = 289  
SCREENHEIGHT = 511  
SCREEN = pygame.display.set\_mode((SCREENWIDTH, SCREENHEIGHT))  
GROUNDY = SCREENHEIGHT \* 0.8  
GAME\_SPRITES = {}  
GAME\_SOUNDS = {}  
PLAYER = 'gallery/sprites/bird.png'  
BACKGROUND = 'gallery/sprites/background.png'  
PIPE = 'gallery/sprites/pipe.png'  
GAMEOVER='gallery/sprites/gm6.jpeg'  
def welcomeScreen():  
 *"""  
 Shows welcome images on the screen  
 """* playerx = int(SCREENWIDTH/5)  
 playery = int((SCREENHEIGHT - GAME\_SPRITES['player'].get\_height())/2)  
 messagex = int((SCREENWIDTH - GAME\_SPRITES['message'].get\_width())/2)  
 messagey = int(SCREENHEIGHT\*0.13)  
 basex = 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):  
 mainGame()  
 else:  
 SCREEN.blit(GAME\_SPRITES['background'], (0, 0))   
 SCREEN.blit(GAME\_SPRITES['player'], (playerx, playery))   
 SCREEN.blit(GAME\_SPRITES['message'], (messagex,messagey ))   
 SCREEN.blit(GAME\_SPRITES['base'], (basex, GROUNDY))   
 pygame.display.update()  
 FPSCLOCK.tick(FPS)  
  
def mainGame():  
 score = 0  
 playerx = int(SCREENWIDTH/5)  
 playery = int(SCREENWIDTH/2)  
 basex = 0  
  
 # Create 2 pipes for blitting on the screen  
 newPipe1 = getRandomPipe()  
 newPipe2 = getRandomPipe()  
  
 # my List of upper pipes  
 upperPipes = [  
 {'x': SCREENWIDTH+200, 'y':newPipe1[0]['y']},  
 {'x': SCREENWIDTH+200+(SCREENWIDTH/2), 'y':newPipe2[0]['y']},  
 ]  
 # my List of lower pipes  
 lowerPipes = [  
 {'x': SCREENWIDTH+200, 'y':newPipe1[1]['y']},  
 {'x': SCREENWIDTH+200+(SCREENWIDTH/2), 'y':newPipe2[1]['y']},  
 ]  
  
 pipeVelX = -4  
  
 playerVelY = -9  
 playerMaxVelY = 10  
 playerMinVelY = -8  
 playerAccY = 1  
  
 playerFlapAccv = -8 # velocity while flapping  
 playerFlapped = False # It is true only when the bird is flapping  
  
  
 while True:  
 for event in pygame.event.get():  
 if event.type == QUIT or (event.type == KEYDOWN and event.key == K\_ESCAPE):  
 pygame.quit()  
 sys.exit()  
 if event.type == KEYDOWN and (event.key == K\_SPACE or event.key == K\_UP):  
 if playery > 0:  
 playerVelY = playerFlapAccv  
 playerFlapped = True  
 GAME\_SOUNDS['wing'].play()  
  
  
 crashTest = isCollide(playerx, playery, upperPipes, lowerPipes) # This function will return true if the player is crashed  
 if crashTest:  
 endGame(score)  
 #return  
  
 #check for score  
 playerMidPos = playerx + GAME\_SPRITES['player'].get\_width()/2  
 for pipe in upperPipes:  
 pipeMidPos = pipe['x'] + GAME\_SPRITES['pipe'][0].get\_width()/2  
 if pipeMidPos<= playerMidPos < pipeMidPos +4:  
 score +=1  
 print(f"Your score is {score}")   
 GAME\_SOUNDS['point'].play()  
  
  
 if playerVelY <playerMaxVelY and not playerFlapped:  
 playerVelY += playerAccY  
  
 if playerFlapped:  
 playerFlapped = False   
 playerHeight = GAME\_SPRITES['player'].get\_height()  
 playery = playery + min(playerVelY, GROUNDY - playery - playerHeight)  
  
 # move pipes to the left  
 for upperPipe , lowerPipe in zip(upperPipes, lowerPipes):  
 upperPipe['x'] += pipeVelX  
 lowerPipe['x'] += pipeVelX  
  
 # Add a new pipe when the first is about to cross the leftmost part of the screen  
 if 0<upperPipes[0]['x']<5:  
 newpipe = getRandomPipe()  
 upperPipes.append(newpipe[0])  
 lowerPipes.append(newpipe[1])  
  
 # if the pipe is out of the screen, remove it  
 if upperPipes[0]['x'] < -GAME\_SPRITES['pipe'][0].get\_width():  
 upperPipes.pop(0)  
 lowerPipes.pop(0)  
   
 # Lets blit our sprites now  
 SCREEN.blit(GAME\_SPRITES['background'], (0, 0))  
 for upperPipe, lowerPipe in zip(upperPipes, lowerPipes):  
 SCREEN.blit(GAME\_SPRITES['pipe'][0], (upperPipe['x'], upperPipe['y']))  
 SCREEN.blit(GAME\_SPRITES['pipe'][1], (lowerPipe['x'], lowerPipe['y']))  
  
 SCREEN.blit(GAME\_SPRITES['base'], (basex, GROUNDY))  
 SCREEN.blit(GAME\_SPRITES['player'], (playerx, playery))  
 myDigits = [int(x) for x in list(str(score))]  
 width = 0  
 for digit in myDigits:  
 width += GAME\_SPRITES['numbers'][digit].get\_width()  
 Xoffset = (SCREENWIDTH - width)/2  
  
 for digit in myDigits:  
 SCREEN.blit(GAME\_SPRITES['numbers'][digit], (Xoffset, SCREENHEIGHT\*0.12))  
 Xoffset += GAME\_SPRITES['numbers'][digit].get\_width()  
 pygame.display.update()  
 FPSCLOCK.tick(FPS)  
  
def isCollide(playerx, playery, upperPipes, lowerPipes):  
 if playery> GROUNDY - 25 or playery<0:  
 GAME\_SOUNDS['hit'].play()  
 return True  
  
 for pipe in upperPipes:  
 pipeHeight = GAME\_SPRITES['pipe'][0].get\_height()  
 if(playery < pipeHeight + pipe['y'] and abs(playerx - pipe['x']) < GAME\_SPRITES['pipe'][0].get\_width()):  
 GAME\_SOUNDS['hit'].play()  
 return True  
  
 for pipe in lowerPipes:  
 if (playery + GAME\_SPRITES['player'].get\_height() > pipe['y']) and abs(playerx - pipe['x']) < GAME\_SPRITES['pipe'][0].get\_width():  
 GAME\_SOUNDS['hit'].play()  
 return True  
  
 return False  
  
def getRandomPipe():  
 *"""  
 Generate positions of two pipes(one bottom straight and one top rotated ) for blitting on the screen  
 """* pipeHeight = GAME\_SPRITES['pipe'][0].get\_height()  
 offset = SCREENHEIGHT/3  
 y2 = offset + random.randrange(0, int(SCREENHEIGHT - GAME\_SPRITES['base'].get\_height() - 1.2 \*offset))  
 pipeX = SCREENWIDTH + 10  
 y1 = pipeHeight - y2 + offset  
 pipe = [  
 {'x': pipeX, 'y': -y1}, #upper Pipe  
 {'x': pipeX, 'y': y2} #lower Pipe  
 ]  
 return pipe  
  
def endGame(score):  
 gameOverX = int((SCREENWIDTH - GAME\_SPRITES['gameOver'].get\_width()) \* 0.5)  
 gameOverY = int(SCREENHEIGHT \* 0.1)  
 SCREEN.blit(GAME\_SPRITES['background'], (0, 0))  
 SCREEN.blit(GAME\_SPRITES['gameOver'], (gameOverX\*0.1, gameOverY\*0.2))  
 SCREEN.blit(GAME\_SPRITES['restart'], (SCREENWIDTH\*0.18, SCREENHEIGHT\*0.5))  
 SCREEN.blit(GAME\_SPRITES['base'], (0, GROUNDY))  
  
# show final score  
 scoreFont = pygame.font.SysFont('couriernew', 40, bold=True)  
 display = scoreFont.render(f"Score: {score}", True, (255,200,0))  
 SCREEN.blit(display, (SCREENWIDTH\*0.2, SCREENHEIGHT\*0.4))  
  
 while True:  
 for event in pygame.event.get():  
 if event.type == pygame.QUIT or (event.type == pygame.KEYDOWN and event.key == pygame.K\_ESCAPE):  
 pygame.quit()  
 sys.exit()  
 elif event.type == pygame.KEYDOWN and (event.key == pygame.K\_SPACE or event.key == pygame.K\_UP):  
 welcomeScreen()  
  
 pygame.display.update()  
 FPSCLOCK.tick(FPS)  
  
  
  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # This will be the main point from where our game will start  
 pygame.init() # Initialize all pygame's modules  
 FPSCLOCK = pygame.time.Clock()  
 pygame.display.set\_caption('Flappy Bird')  
 GAME\_SPRITES['numbers'] = (   
 pygame.image.load('gallery/sprites/0.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/1.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/2.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/3.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/4.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/5.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/6.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/7.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/8.png').convert\_alpha(),  
 pygame.image.load('gallery/sprites/9.png').convert\_alpha(),  
 )  
  
 GAME\_SPRITES['message'] =pygame.image.load('gallery/sprites/front5.jpg').convert\_alpha()  
 GAME\_SPRITES['base'] =pygame.image.load('gallery/sprites/base.png').convert\_alpha()  
 GAME\_SPRITES['gameOver'] = pygame.image.load('gallery/sprites/gm8.jpg').convert\_alpha()  
 GAME\_SPRITES['restart'] = pygame.image.load('gallery/sprites/restart9.jpg').convert\_alpha()  
 GAME\_SPRITES['pipe'] =(pygame.transform.rotate(pygame.image.load(PIPE).convert\_alpha(), 180),  
 pygame.image.load(PIPE).convert\_alpha()  
  
 )  
  
 # Game sounds  
 GAME\_SOUNDS['die'] = pygame.mixer.Sound('gallery/audio/die.wav')  
 GAME\_SOUNDS['hit'] = pygame.mixer.Sound('gallery/audio/hit.wav')  
 GAME\_SOUNDS['point'] = pygame.mixer.Sound('gallery/audio/point.wav')  
 GAME\_SOUNDS['swoosh'] = pygame.mixer.Sound('gallery/audio/swoosh.wav')  
 GAME\_SOUNDS['wing'] = pygame.mixer.Sound('gallery/audio/wing.wav')  
  
 GAME\_SPRITES['background'] = pygame.image.load(BACKGROUND).convert()  
 GAME\_SPRITES['player'] = pygame.image.load(PLAYER).convert\_alpha()  
  
 while True:  
 welcomeScreen() # Shows welcome screen to the user until he presses a button  
 #mainGame() # This is the main game function

# Output of the Program







**Future Plans:-**

* The current version of our project is offline, we are planning to make it online.
* We are planning to add the global leader boards functionality so that players can compete online.

**References:-**

* [**https://pygame.org/**](https://pygame.org/)
* **Learning pygame by making flappybird – Clear Code**
* [**https://www.youtube.com/watch?v=UZg49z76cLw&t=1215**](https://www.youtube.com/watch?v=UZg49z76cLw&t=1215s) [**s**](https://www.youtube.com/watch?v=UZg49z76cLw&t=1215s)
* **Tech with Tim – Pygame programming tutorials**
* [**https://www.youtube.com/watch?v=i6xMBig-pP4&list=PLz**](https://www.youtube.com/watch?v=i6xMBig-pP4&list=PLzMcBGfZo4-lp3jAExUCewBfMx3UZFkh5) [**McBGfZo4-lp3jAExUCewBfMx3UZFkh5**](https://www.youtube.com/watch?v=i6xMBig-pP4&list=PLzMcBGfZo4-lp3jAExUCewBfMx3UZFkh5)
* **Code with Harry - Pygame programming tutorials**
* **https://youtu.be/itB6VsP5UnA**