**Flappy Game**

**Using Python (pygame Library)**

Flappy Bird is a mobile game developed by Vietnamese video game artist and programmer Dong Nguyen, under his game development company dotGears. The game is a side-scroller where the player controls a bird, attempting to fly between columns of green pipes without hitting them. Here is a python version of it.

**Code:**

**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'**

**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)**

**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)**

**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**

**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/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()**

**)**

**# 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**

****