|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
| МИНОБРНАУКИ РОССИИ | | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | |
| **Институт кибербезопасности и цифровых технологий** | | | |
|  | |

Отчёт для практического задания по дисциплине

**Интерпретируемый язык программирования высокого уровня**

Выполнил:

Студент 3 курса

Группа ББСО-03-19.

Шифр: 19У0415

Бахтуев Даниил Игоревич

**Москва 2021**

*Формализированное описание проекта*

Проект представляет собой копию некогда популярной игры “Flappy Bird”. Принцип игры остался неизменным; необходимо, управляя птичкой пролетать через трубы, и набирать очки. Создан проект с помощью PyGame, в качестве редактора кода использован PyCharm.

*Особенности игры*

* Цвет труб меняется раз в 10 сек
* Фон будет меняться раз в 5 очков
* В игре присутствует несколько уровней сложности
* С повышением уровня сложности увеличивается скорость птицы и труб

Описание модуля Pygame

**Pygame** — набор модулей (библиотек) языка программирования Python, предназначенный для написания компьютерных игр и мультимедиаприложений. Pygame базируется на мультимедийной библиотеке SDL.

Одна из библиотек, предоставляющих доступ к API SDL (существуют и другие). В тоже время дает возможность написания более высокоуровневого кода.

Pygame-приложения могут работать под Android на телефонах и планшетах с использованием подмножества Pygame для Android (***pgs4a***). На этой платформе поддерживаются звук, вибрация, клавиатура, акселерометр.

*Листинг программного кода:*

import pygame  
import random  
  
  
def shift(lst, steps):  
 if steps < 0:  
 steps = abs(steps)  
 for \_ in range(steps):  
 lst.append(lst.pop(0))  
 else:  
 for \_ in range(steps):  
 lst.insert(0, lst.pop())  
  
  
def choose\_bg\_color(points\_amount):  
 if points\_amount % 10 >= 5:  
 screen.fill(BLACK)  
 else:  
 screen.fill(SKY)  
  
  
# Настройки окна  
WIDTH = 500  
HEIGHT = 500  
FPS = 60  
  
# Цвета  
SKY = (133, 193, 233)  
YELLOW = (255, 215, 0)  
GREEN = (46, 204, 113)  
WHITE = (255, 255, 255)  
VIOLET = (199, 21, 133)  
MAROON = (128, 0, 0)  
BLUE = (25, 25, 112)  
BLACK = (16, 16, 16)  
RED = (255, 99, 71)  
GRAY = (211, 211, 211)  
colors = [GREEN, YELLOW, VIOLET, MAROON, RED, BLUE]  
color = random.choice(colors)  
  
  
pygame.init()  
screen = pygame.display.set\_mode((WIDTH, HEIGHT))  
pygame.display.set\_caption('Floppy Bird')   
  
bird = pygame.Rect(40, 250, 30, 23)  
bird\_img1 = pygame.image.load('assets/bird/fpbs1.png')  
bird\_img2 = pygame.image.load('assets/bird/fpbs2.png')  
bird\_img3 = pygame.image.load('assets/bird/fpbs3.png')  
bird\_images = [bird\_img1, bird\_img2, bird\_img3]  
points = 0  
  
  
DEFAULT\_FONT = 'Arial Rounded'  
menu\_font = pygame.font.SysFont(DEFAULT\_FONT, 20)  
difficulty\_font = pygame.font.SysFont(DEFAULT\_FONT, 38)  
points\_font = pygame.font.SysFont(DEFAULT\_FONT, 30)  
game\_over\_font = pygame.font.SysFont(DEFAULT\_FONT, 50)

game\_over\_text = game\_over\_font.render('GAME OVER', True, WHITE)  
title\_text = game\_over\_font.render('Floppy Bird', True, WHITE)  
play\_text = menu\_font.render('Play', True, BLACK)  
difficulty\_plus\_text = difficulty\_font.render("+", True, BLACK)  
difficulty\_minus\_text = difficulty\_font.render("-", True, BLACK)  
exit\_text = menu\_font.render('Exit', True, BLACK)  
back\_to\_menu\_text = menu\_font.render('Menu', True, BLACK)  
  
menu\_bg = pygame.image.load("assets/menu\_bg.jpg")  
game\_over\_bg = pygame.image.load("assets/GO\_bg.png")  
  
GRAVITY = 0.3  
y\_change = 0  
  
is\_jump = False  
hop\_count = 10  
  
  
pipes = []  
pipe\_cd = 1500

pipe\_cd\_step = 500   
pipe\_speed = 2  
space\_between\_pipe = 100   
  
check = []  
  
clock = pygame.time.Clock()  
current\_time = 0  
last\_pipe\_time = 0  
  
  
color\_counter = 0  
  
game\_mode = 'MENU'  
running = True  
while running:  
  
   
 if game\_mode == 'MENU':

screen.fill(SKY)  
 screen.blit(menu\_bg, (0, 0))  
  
 screen.blit(title\_text, (120, 80))  
  
   
 for event in pygame.event.get():  
   
 if event.type == pygame.QUIT:  
 running = False  
  
   
 if event.type == pygame.MOUSEBUTTONDOWN:

if 170 < event.pos[0] < 160 + 170 and 180 < event.pos[1] < 180 + 50:  
 game\_mode = 'GAME'  
  
   
 if 100 < event.pos[0] < 100 + 50 and 250 < event.pos[1] < 250 + 50:  
   
 if event.button == 1:  
   
 if pipe\_speed > 1:

pipe\_speed -= 1  
 pipe\_cd += pipe\_cd\_step  
  
   
 if 350 < event.pos[0] < 350 + 50 and 250 < event.pos[1] < 250 + 50:

if event.button == 1:  
   
 if pipe\_speed < 5:

pipe\_speed += 1  
 pipe\_cd -= pipe\_cd\_step

if pipe\_speed == 5:  
 pipe\_cd = 500

if 170 < event.pos[0] < 160 + 170 and 320 < event.pos[1] < 320 + 50:  
 running = False

pygame.draw.rect(screen, GREEN, (170, 180, 160, 50))  
 pygame.draw.rect(screen, YELLOW, (170, 250, 160, 50))  
 pygame.draw.rect(screen, GRAY, (350, 250, 50, 50))

pygame.draw.rect(screen, GRAY, (100, 250, 50, 50))

pygame.draw.rect(screen, RED, (170, 320, 160, 50))  
  
   
 screen.blit(play\_text, (230, 192))  
 difficulty\_text = menu\_font.render("Difficulty: " + str(pipe\_speed), True, BLACK)  
 screen.blit(difficulty\_text, (197, 263))  
 screen.blit(difficulty\_plus\_text, (365, 253))  
 screen.blit(difficulty\_minus\_text, (118, 250))  
 screen.blit(exit\_text, (230, 332))  
  
 if game\_mode == 'GAME':  
  
 choose\_bg\_color(points)  
  
   
 for i in pygame.event.get():

if i.type == pygame.QUIT:  
 running = False

if i.type == pygame.KEYDOWN:

if i.key == pygame.K\_SPACE:  
 is\_jump = True  
 hop\_count = 0  
  
 if is\_jump:  
 hop\_count += 1  
 bird.top -= 6  
  
 if hop\_count == 5:  
 y\_change = 0  
 is\_jump = False  
   
 else:  
 y\_change += GRAVITY  
 bird.top += y\_change  
  
   
 current\_time = pygame.time.get\_ticks()  
  
   
 if current\_time - last\_pipe\_time > pipe\_cd:  
 width\_pipe = 40  
   
 height\_up\_pipe = random.randint(50, 400)  
 up\_pipe = pygame.Rect(WIDTH, 0, width\_pipe, height\_up\_pipe)  
  
   
 y\_down\_pipe = height\_up\_pipe + space\_between\_pipe  
 height\_down\_pipe = HEIGHT - y\_down\_pipe  
 down\_pipe = pygame.Rect(WIDTH, y\_down\_pipe, width\_pipe, height\_down\_pipe)

y\_middle\_space = height\_up\_pipe  
 height\_middle\_space = HEIGHT - height\_up\_pipe - height\_down\_pipe  
 middle\_space = pygame.Rect(WIDTH, y\_middle\_space, width\_pipe, height\_middle\_space)  
 check.append(middle\_space)

pipes.append((up\_pipe, down\_pipe))  
  
   
 last\_pipe\_time = current\_time  
 pipe\_cd = random.randint(pipe\_cd, pipe\_cd + pipe\_cd\_step)  
  
   
 if color\_counter == 300:  
 color = random.choice(colors)  
 color\_counter = 0  
  
   
 for pipe in pipes:  
   
 pygame.draw.rect(screen, color, pipe[0])  
 pygame.draw.rect(screen, color, pipe[1])  
   
 pipe[0].left -= pipe\_speed  
 pipe[1].left -= pipe\_speed

if bird.top < 0 or bird.bottom > HEIGHT:  
 game\_mode = 'GAME OVER'

for pipe in pipes:  
 up\_pipe = pipe[0]  
 down\_pipe = pipe[1]  
  
 if bird.colliderect(up\_pipe):  
 game\_mode = 'GAME OVER'  
 elif bird.colliderect(down\_pipe):  
 game\_mode = 'GAME OVER'  
  
   
 for flag in check:  
 flag.left -= pipe\_speed

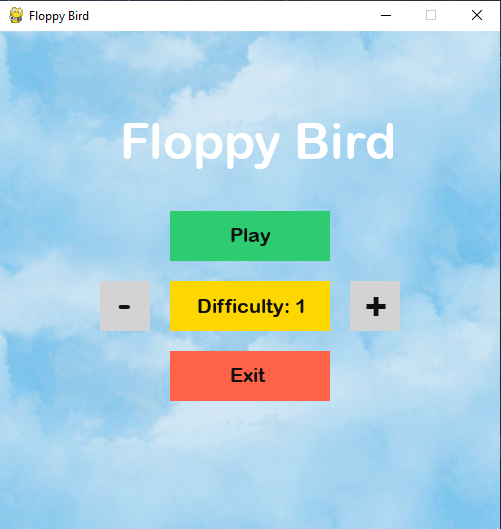
if bird.colliderect(flag):  
 points += 1  
 check.remove(flag)  
  
   
 shift(bird\_images, 1)  
 screen.blit(bird\_images[0], (bird.left, bird.top))  
  
 points\_text = points\_font.render("Points: " + str(points), True, WHITE)  
 screen.blit(points\_text, (10, 10))  
  
 color\_counter += 1  
  
   
 if game\_mode == 'GAME OVER':  
  
 choose\_bg\_color(points)  
 screen.blit(game\_over\_bg, (0, 0))  
  
 screen.blit(game\_over\_text, (100, 100))  
  
 final\_points\_text = game\_over\_font.render("Points: " + str(points), True, WHITE)  
 screen.blit(final\_points\_text, (145, 180))  
  
   
 pygame.draw.rect(screen, GREEN, (170, 300, 160, 50))  
 screen.blit(back\_to\_menu\_text, (222, 312))  
  
   
 for event in pygame.event.get():  
   
 if event.type == pygame.QUIT:  
 running = False

if event.type == pygame.MOUSEBUTTONDOWN:  
   
 if 170 < event.pos[0] < 160 + 170 and 300 < event.pos[1] < 300 + 50:  
 bird = pygame.Rect(40, 250, 30, 23)  
 points = 0  
 pipe\_cd = 1500  
 pipe\_speed = 2  
 pipes.clear()  
 check.clear()  
 y\_change = 0  
  
 game\_mode = 'MENU'  
  
 clock.tick(FPS)  
 pygame.display.update()  
pygame.quit()

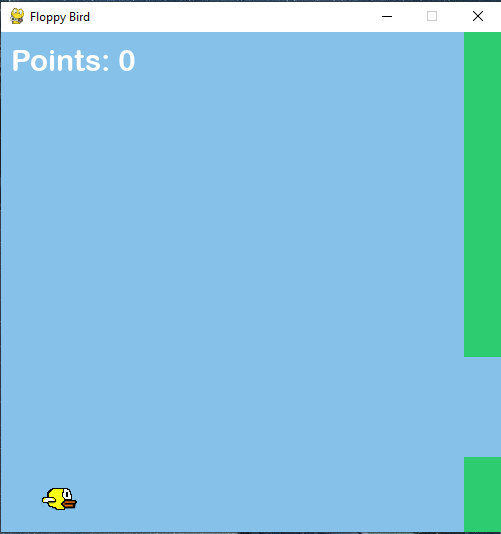
*Описание Функций, использованных в программе*

def shift(lst, steps):  
 *"""  
 Функция для циклического сдвига элементов списка на заданное количество шагов* ***:param*** *lst: список, в котором сдвигаем элементы* ***:param*** *steps: количество шагов для сдвига  
 """* if steps < 0:  
 steps = abs(steps)  
 for \_ in range(steps):  
 lst.append(lst.pop(0))  
 else:  
 for \_ in range(steps):  
 lst.insert(0, lst.pop())  
  
  
def choose\_bg\_color(points\_amount):  
 *"""  
 Выбор цвета фона игры в зависимости от количества очков  
 (каждые 5 очков цвет меняется с темного на светлый и наоборот)* ***:param*** *points\_amount: количества очков  
 """* if points\_amount % 10 >= 5:  
 screen.fill(BLACK)  
 else:  
 screen.fill(SKY)

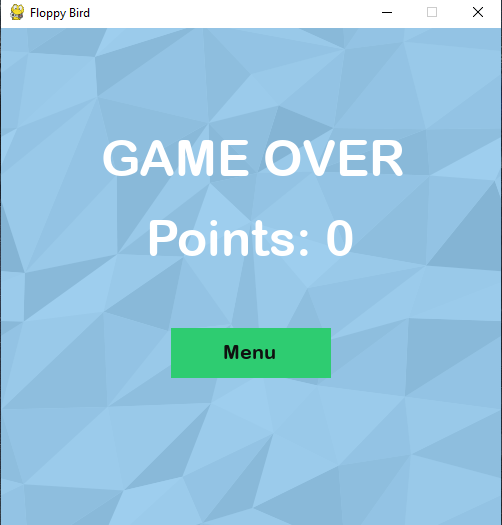
*Окно при выполнении проекта выглядит следующим образом:*

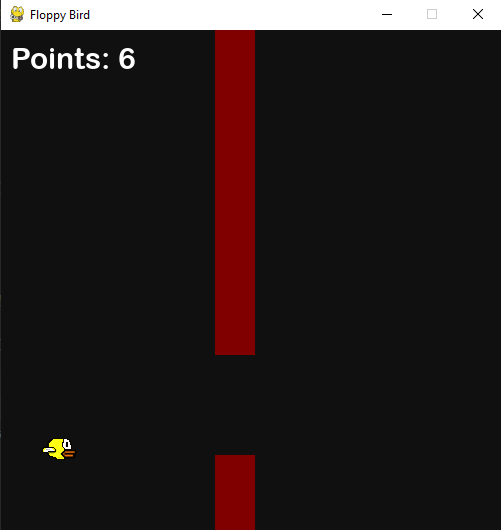


Здесь мы можем начать игру, выйти из неё, и повысить уровень сложности нашей игры.

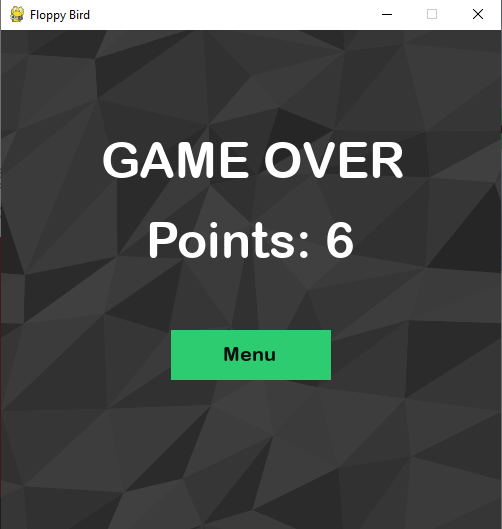


Запустив игру, мы переходим к самому игровому процессу/





На этом скриншоте показан процесс смена фона и смены цвета трубы.

Вот таким образом выглядит окно поражения, при наборе 5 и более очков.