

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»

Отчет по домашнему заданию
«Разработка игры Flappy Bird на языке программирования Python»

Выполнил:
студент группы ИУ5-35Б
Шакиров Тимур
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Юрий Евгеньевич
Подпись и дата:

Постановка задачи

1. Написать логику игры Flappy Bird, используя язык программирования Python
2. Описать объекты, необходимые для создания игры
3. Реализовать графическую составляющую игры

Текст программы

objects.py

```
import pygame
import random

SCREEN = WIDTH, HEIGHT = 288, 512
display_height = 0.80 * HEIGHT

pygame.mixer.init()
wing_fx = pygame.mixer.Sound('Game\\Sounds\\wing.wav')

class Grumpy:
    def __init__(self, win):
        self.win = win

        self.im_list = []
        bird_color = random.choice(['red', 'blue', 'yellow'])
        for i in range(1,4):
            img = pygame.image.load(f'Game\\Assets\\Grumpy\\{bird_color}{i}.png')
            self.im_list.append(img)

        self.reset()

    def update(self):
        # gravity
        self.vel += 0.3
        if self.vel >= 8:
            self.vel = 8
        if self.rect.bottom <= display_height:
            self.rect.y += int(self.vel)

        if self.alive:

            # jump
            if pygame.mouse.get_pressed()[0] == 1 and not self.jumped:
                wing_fx.play()
                self.jumped = True
                self.vel = -6
            if pygame.mouse.get_pressed()[0] == 0:
                self.jumped = False

            self.flap_counter()

            self.image = pygame.transform.rotate(self.im_list[self.index], self.vel * -2)
        else:
```

```

        if self.rect.bottom <= display_height:
            self.theta -= 2
            self.image = pygame.transform.rotate(self.im_list[self.index], self.theta)

        self.win.blit(self.image, self.rect)

def flap_counter(self):
    #animation
    self.counter += 1
    if self.counter > 5:
        self.counter = 0
        self.index += 1
    if self.index >= 3:
        self.index = 0

def draw_flap(self):
    self.flap_counter()
    if self.flap_pos <= -10 or self.flap_pos > 10:
        self.flap_inc *= -1
    self.flap_pos += self.flap_inc
    self.rect.y += self.flap_inc
    self.rect.x = WIDTH // 2 - 20
    self.image = self.im_list[self.index]
    self.win.blit(self.image, self.rect)

def reset(self):
    self.index = 0
    self.image = self.im_list[self.index]
    self.rect = self.image.get_rect()
    self.rect.x = 60
    self.rect.y = int(display_height) // 2
    self.counter = 0
    self.vel = 0
    self.jumped = False
    self.alive = True
    self.theta = 0
    self.mid_pos = display_height // 2
    self.flap_pos = 0
    self.flap_inc = 1

class Base:
    def __init__(self, win):
        self.win = win

        self.image1 = pygame.image.load('Game\\Assets\\base.png')
        self.image2 = self.image1
        self.rect1 = self.image1.get_rect()
        self.rect1.x = 0
        self.rect1.y = int(display_height)
        self.rect2 = self.image2.get_rect()
        self.rect2.x = WIDTH
        self.rect2.y = int(display_height)

    def update(self, speed):

```

```

        self.rect1.x -= speed
        self.rect2.x -= speed

        if self.rect1.right <= 0:
            self.rect1.x = WIDTH - 5
        if self.rect2.right <= 0:
            self.rect2.x = WIDTH - 5

        self.win.blit(self.image1, self.rect1)
        self.win.blit(self.image2, self.rect2)

class Pipe(pygame.sprite.Sprite):
    def __init__(self, win, image, y, position):
        super(Pipe, self).__init__()

        self.win = win
        self.image = image
        self.rect = self.image.get_rect()
        pipe_gap = 100 // 2
        x = WIDTH

        if position == 1:
            self.image = pygame.transform.flip(self.image, False, True)
            self.rect.bottomleft = (x, y - pipe_gap)
        elif position == -1:
            self.rect.topleft = (x, y + pipe_gap)

    def update(self, speed):
        self.rect.x -= speed
        if self.rect.right < 0:
            self.kill()
        self.win.blit(self.image, self.rect)

class Score:
    def __init__(self, x, y, win):
        self.score_list = []
        for score in range(10):
            img = pygame.image.load(f'Game\\Assets\\Score\\{score}.png')
            self.score_list.append(img)
            self.x = x
            self.y = y

        self.win = win

    def update(self, score):
        score = str(score)
        for index, num in enumerate(score):
            self.image = self.score_list[int(num)]
            self.rect = self.image.get_rect()
            self.rect.topleft = self.x - 15 * len(score) + 30 * index, self.y
            self.win.blit(self.image, self.rect)

```

main.py

```
import pygame
import random

from objects import Grumpy, Pipe, Base, Score

pygame.init()
SCREEN = WIDTH, HEIGHT = 288, 512
display_height = 0.80 * HEIGHT
info = pygame.display.Info()

width = info.current_w
height = info.current_h

if width >= height:
    win = pygame.display.set_mode(SCREEN, pygame.NOFRAME)
else:
    win = pygame.display.set_mode(SCREEN, pygame.NOFRAME | pygame.SCALED |
pygame.FULLSCREEN)

clock = pygame.time.Clock()
FPS = 60

# COLORS

RED = (255, 0, 0)
WHITE = (255, 255, 255)
BLACK = (0, 0, 0)

# Backgrounds

bg1 = pygame.image.load('Game\\Assets\\background-day.png')
bg2 = pygame.image.load('Game\\Assets\\background-night.png')

bg = random.choice([bg1, bg2])

im_list = [pygame.image.load('Game\\Assets\\pipe-green.png'),
pygame.image.load('Game\\Assets\\pipe-red.png')]
pipe_img = random.choice(im_list)

gameover_img = pygame.image.load('Game\\Assets\\gameover.png')
flappybird_img = pygame.image.load('Game\\Assets\\flappybird.png')
flappybird_img = pygame.transform.scale(flappybird_img, (200,80))

# Sounds & fx

die_fx = pygame.mixer.Sound('Game\\Sounds\\die.wav')
hit_fx = pygame.mixer.Sound('Game\\Sounds\\hit.wav')
point_fx = pygame.mixer.Sound('Game\\Sounds\\point.wav')
swoosh_fx = pygame.mixer.Sound('Game\\Sounds\\swoosh.wav')
wing_fx = pygame.mixer.Sound('Game\\Sounds\\wing.wav')

# Objects
```

```

pipe_group = pygame.sprite.Group()
base = Base(win)
score_img = Score(WIDTH // 2, 50, win)
grumpy = Grumpy(win)

# Variables

base_height = 0.80 * HEIGHT
speed = 0
game_started = False
game_over = False
restart = False
score = 0
start_screen = True
pipe_pass = False
pipe_frequency = 1600

running = True
while running:
    win.blit(bg, (0,0))

    if start_screen:
        speed = 0
        grumpy.draw_flap()
        base.update(speed)

        win.blit(flappybird_img, (40, 50))
    else:

        if game_started and not game_over:

            next_pipe = pygame.time.get_ticks()
            if next_pipe - last_pipe >= pipe_frequency:
                y = display_height // 2
                pipe_pos = random.choice(range(-100,100,4))
                height = y + pipe_pos

                top = Pipe(win, pipe_img, height, 1)
                bottom = Pipe(win, pipe_img, height, -1)
                pipe_group.add(top)
                pipe_group.add(bottom)
                last_pipe = next_pipe

            pipe_group.update(speed)
            base.update(speed)
            grumpy.update()
            score_img.update(score)

            if pygame.sprite.spritecollide(grumpy, pipe_group, False) or grumpy.rect.top <= 0:
                game_started = False
                if grumpy.alive:
                    hit_fx.play()
                    die_fx.play()
                grumpy.alive = False
                grumpy.theta = grumpy.vel * -2

```

```

    if grumpy.rect.bottom >= display_height:
        speed = 0
        game_over = True

    if len(pipe_group) > 0:
        p = pipe_group.sprites()[0]
        if grumpy.rect.left > p.rect.left and grumpy.rect.right < p.rect.right and not
pipe_pass and grumpy.alive:
            pipe_pass = True

        if pipe_pass:
            if grumpy.rect.left > p.rect.right:
                pipe_pass = False
                score += 1
                point_fx.play()

    if not grumpy.alive:
        win.blit(gameover_img, (50,200))

for event in pygame.event.get():
    if event.type == pygame.QUIT:
        running = False
    if event.type == pygame.KEYDOWN:
        if event.key == pygame.K_ESCAPE or \
            event.key == pygame.K_q:
            running = False
    if event.type == pygame.MOUSEBUTTONDOWN:
        if start_screen:
            game_started = True
            speed = 2
            start_screen = False

        game_over = False
        # grumpy.reset()
        last_pipe = pygame.time.get_ticks() - pipe_frequency
        next_pipe = 0
        pipe_group.empty()

        speed = 2
        score = 0

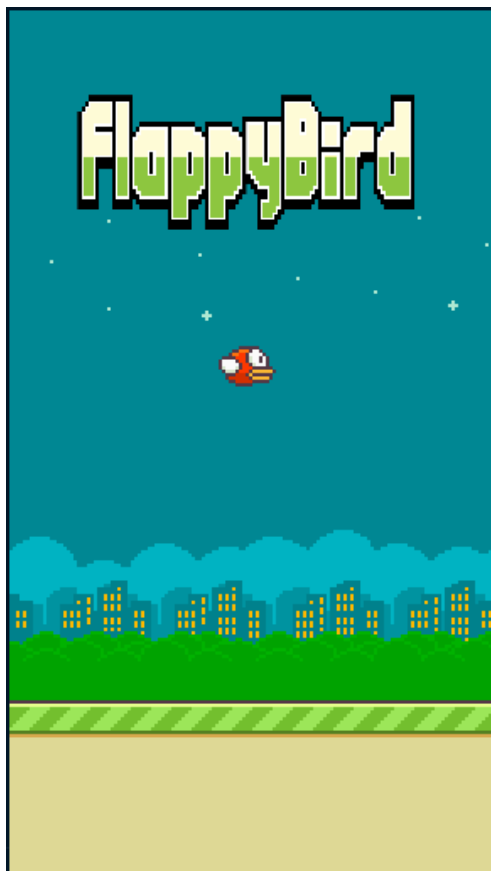
    if game_over:
        start_screen = True
        grumpy = Grumpy(win)
        pipe_img = random.choice(im_list)
        bg = random.choice([bg1, bg2])

clock.tick(FPS)
pygame.display.update()
pygame.quit()

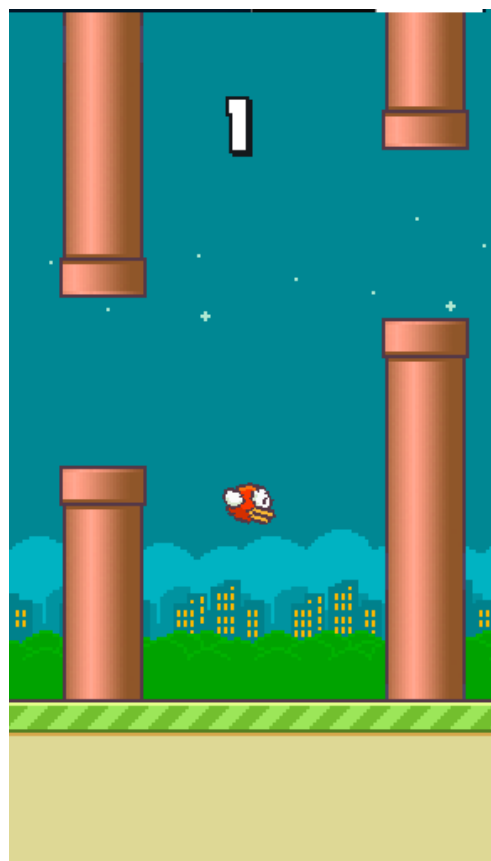
```

Анализ результатов

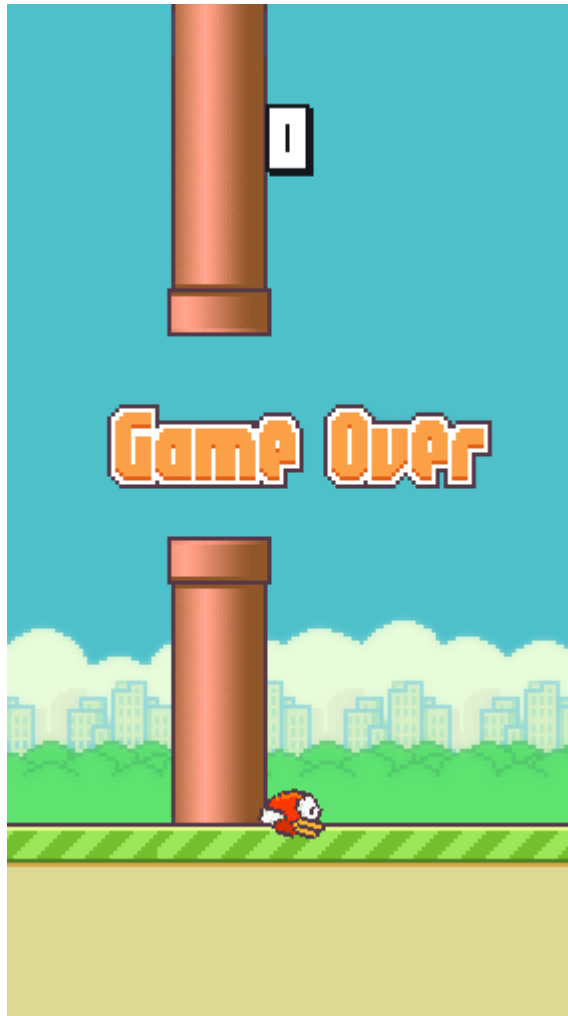
Начало игры



Игра идёт



Конец игры



Вывод

Я изучил библиотеку Pygame языка программирования Python и создал с помощью неё игру, в которую можно поиграть, когда скучно. Кроме того, я научился работать с аудио и видео файлами в Python.