МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В. Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа № 15**

по дисциплине: Объектно-ориентированное программирование

тема: «Знакомство с библиотеками языка Python. PyGame»

Выполнил: ст. группы ПВ-223

Игнатьев Артур Олегович

Проверил:

асс. Черников Сергей Викторович

Белгород 2024г.

**Лабораторная работа №15**

**«Знакомство с библиотеками языка Python. PyGame»**

**Цель работы:** приобретение практических навыков создания приложений на языке Python, быстрая разработка 2d игр.

**Вариант 3**



Код программы:

import pygame

BLACK = (0, 0, 0)

WHITE = (255, 255, 255)

GREEN = (0, 255, 0)

RED = (255, 0, 0)

class Button:

    def \_\_init\_\_(self, screen, color, x, y, width, height, text=''):

        self.screen = screen

        self.color = color

        self.x = x

        self.y = y

        self.width = width

        self.height = height

        self.text = text

    def draw(self, outline=None):

        if outline:

            pygame.draw.rect(self.screen, outline, (self.x-2, self.y-2, self.width+4, self.height+4), 0)

        pygame.draw.rect(self.screen, self.color, (self.x, self.y, self.width, self.height), 0)

        if self.text != '':

            font = pygame.font.Font(None, 24)

            text = font.render(self.text, 1, (0, 0, 0))

            self.screen.blit(text, (self.x + (self.width/2 - text.get\_width()/2), self.y + (self.height/2 - text.get\_height()/2)))

    def is\_over(self, pos):

        if pos[0] > self.x and pos[0] < self.x + self.width:

            if pos[1] > self.y and pos[1] < self.y + self.height:

                return True

        return False

class Game:

    def \_\_init\_\_(self, size):

        self.size = size

        self.screen = pygame.display.set\_mode(self.size)

        self.background\_image = pygame.image.load('back.jpg')  # Загрузка изображения фона игры

        self.background\_image = pygame.transform.scale(self.background\_image, self.size)  # Масштабирование изображения фона

        self.character\_image = pygame.image.load('Napcat.webp')  # Путь к файлу изображения персонажа

        self.character\_image = pygame.transform.scale(self.character\_image, (300, 300))

        # Устанавливаем начальные параметры персонажа

        self.character = {"x": 400, "y": 400, "Голод": 100, "Счастье": 100}

        self.happiness\_bar = 200  # Начальное значение шкалы счастья

        self.hunger\_bar = 200  # Начальное значение шкалы голода

        self.frame\_count = 0

        self.happiness\_button = Button(self.screen, GREEN, 50, 250, 150, 50, 'Поиграть')  # Увеличение счастья

        self.hunger\_button = Button(self.screen, RED, 600, 250, 150, 50, 'Покормить')  # Увеличение голода

    def draw\_character(self):

        # Отрисовка персонажа на экране

        self.screen.blit(self.character\_image, (self.character["x"] - 150, self.character["y"] - 100))  # Положение персонажа

    def update(self, frame\_count):

        self.frame\_count += 1

        # Обновление параметров персонажа, например, уменьшение уровня голода и счастья

        self.character["Голод"] -= 0.3  # Уменьшаем уровень голода на 0.05

        self.character["Счастье"] -= 0.3  # Уменьшаем уровень счастья на 0.1

        if self.character["Голод"] <= 0 or self.character["Счастье"] <= 0:

            self.game\_over()  # Вызываем метод для завершения игры

        # Чтобы уровень голода уменьшался медленнее, можно использовать меньшее значение

        if self.frame\_count % 2 == 0:  # Уменьшаем уровень голода только на четных кадрах

            self.character["Голод"] -= 0.01  # Уменьшаем уровень голода на 0.01

        # Похожий код для обновления шкал счастья и голода остается без изменений

        self.happiness\_bar = int(self.character["Счастье"] \* 3)  # Преобразование значения уровня счастья в размер шкалы

        self.hunger\_bar = int(self.character["Голод"] \* 3)  # Преобразование значения уровня голода в размер шкалы

        # Отрисовка персонажа и шкал

        self.screen.blit(self.background\_image, (0, 0))

        self.draw\_character()

        # Отрисовка шкал с подписями

        font = pygame.font.Font(None, 24)

        happiness\_text = font.render("Счастье", True, WHITE)

        self.screen.blit(happiness\_text, (20, 40))

        pygame.draw.rect(self.screen, GREEN, (20, 60, self.happiness\_bar, 30))  # Шкала счастья

        hunger\_text = font.render("Голод", True, WHITE)

        self.screen.blit(hunger\_text, (self.size[0] - 120, 40))

        pygame.draw.rect(self.screen, RED, (self.size[0] - self.hunger\_bar - 20, 60, self.hunger\_bar, 30))  # Шкала голода

        self.happiness\_button.draw(GREEN)  # Отрисовка кнопки для увеличения счастья

        self.hunger\_button.draw(RED)  # Отрисовка кнопки для увеличения уровня голода

        pygame.display.flip()

    def game\_over(self):

        font = pygame.font.Font(None, 100)

        text = font.render("Конец игры", True, RED)

        text\_rect = text.get\_rect(center=(self.size[0] // 2, self.size[1] // 2))

        self.screen.blit(text, text\_rect)

        pygame.display.flip()

        pygame.time.wait(2000)

        # Закрытие игры

        pygame.quit()

        quit()

    def run(self):

        clock = pygame.time.Clock()

        running = True

        while running:

            for event in pygame.event.get():

                if event.type == pygame.QUIT:

                    running = False

                if event.type == pygame.MOUSEBUTTONDOWN:

                    pos = pygame.mouse.get\_pos()

                    if self.happiness\_button.is\_over(pos):

                        self.character["Счастье"] += 10  # Увеличиваем уровень счастья при нажатии на кнопку

                    if self.hunger\_button.is\_over(pos):

                        self.character["Голод"] += 10  # Увеличиваем уровень голода при нажатии на кнопку

            # Обновление состояния персонажа

            self.update(self.frame\_count)

            clock.tick(60)  # Ограничение на 60 кадров в секунду

class GameMenu:

    def \_\_init\_\_(self, size):

        self.size = size

        self.screen = pygame.display.set\_mode(self.size)

        pygame.display.set\_caption("Poppy Pet")

    def draw\_menu(self):

        self.background\_image = pygame.image.load('back.jpg')  # Загрузка изображения фона меню

        self.background\_image = pygame.transform.scale(self.background\_image, self.size)  # Масштабирование изображения фона

        running = True

        while running:

            for event in pygame.event.get():

                if event.type == pygame.QUIT:

                    running = False

            self.screen.blit(self.background\_image, (0, 0))  # Рисуем фон меню

            font = pygame.font.Font(None, 36)

            text = font.render("Домашний питомец", True, WHITE)

            text\_rect = text.get\_rect(center=(self.size[0] // 2, 50))

            self.screen.blit(text, text\_rect)

            button\_1 = pygame.Rect(300, 200, 200, 50)

            pygame.draw.rect(self.screen, GREEN, button\_1)

            text = font.render("Начать игру", True, BLACK)

            text\_rect = text.get\_rect(center=button\_1.center)

            self.screen.blit(text, text\_rect)

            button\_2 = pygame.Rect(300, 300, 200, 50)

            pygame.draw.rect(self.screen, RED, button\_2)

            text = font.render("Выход", True, BLACK)

            text\_rect = text.get\_rect(center=button\_2.center)

            self.screen.blit(text, text\_rect)

            mouse\_pos = pygame.mouse.get\_pos()

            if button\_1.collidepoint(mouse\_pos):

                if pygame.mouse.get\_pressed()[0]:

                    self.start\_game()  # Вызываем метод start\_game при нажатии на кнопку "Начать игру"

            if button\_2.collidepoint(mouse\_pos):

                if pygame.mouse.get\_pressed()[0]:

                    running = False  # Закрываем меню при нажатии на кнопку "Выход"

            pygame.display.flip()

    def start\_game(self):

        self.game = Game(self.size)

        self.game.run()

        self.running = False

def main():

    pygame.init()

    size = weight, height = 800, 600

    menu = GameMenu(size)

    menu.draw\_menu()

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Результат работы программы:

Изображение выглядит как текст, снимок экрана, мебель, в помещении

Автоматически созданное описание

Изображение выглядит как снимок экрана, мультфильм, Анимация, текст

Автоматически созданное описание

Изображение выглядит как снимок экрана, мультфильм, текст, Анимация

Автоматически созданное описание

**Вывод:** на этой лабораторной работе приобрёл практические навыки создания приложений на языке Python, быструю разработку 2d игр.