

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

Игнатъев 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

3.	Домашний питомец
----	------------------

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 изобра-
жения 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, 'По-
играть') # Увеличение счастья
        self.hunger_button = Button(self.screen, RED, 600, 250, 150, 50, 'Покор-
мить') # 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 чет-
ных 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 зна-
чения 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 сча-
стья
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)

# Заккрытие 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 го-
лода 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 изобра-
жения 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 "Вы-
ход"

        pygame.display.flip()

    def start_game(self):
        self.game = Game(self.size)
        self.game.run()
        self.running = False

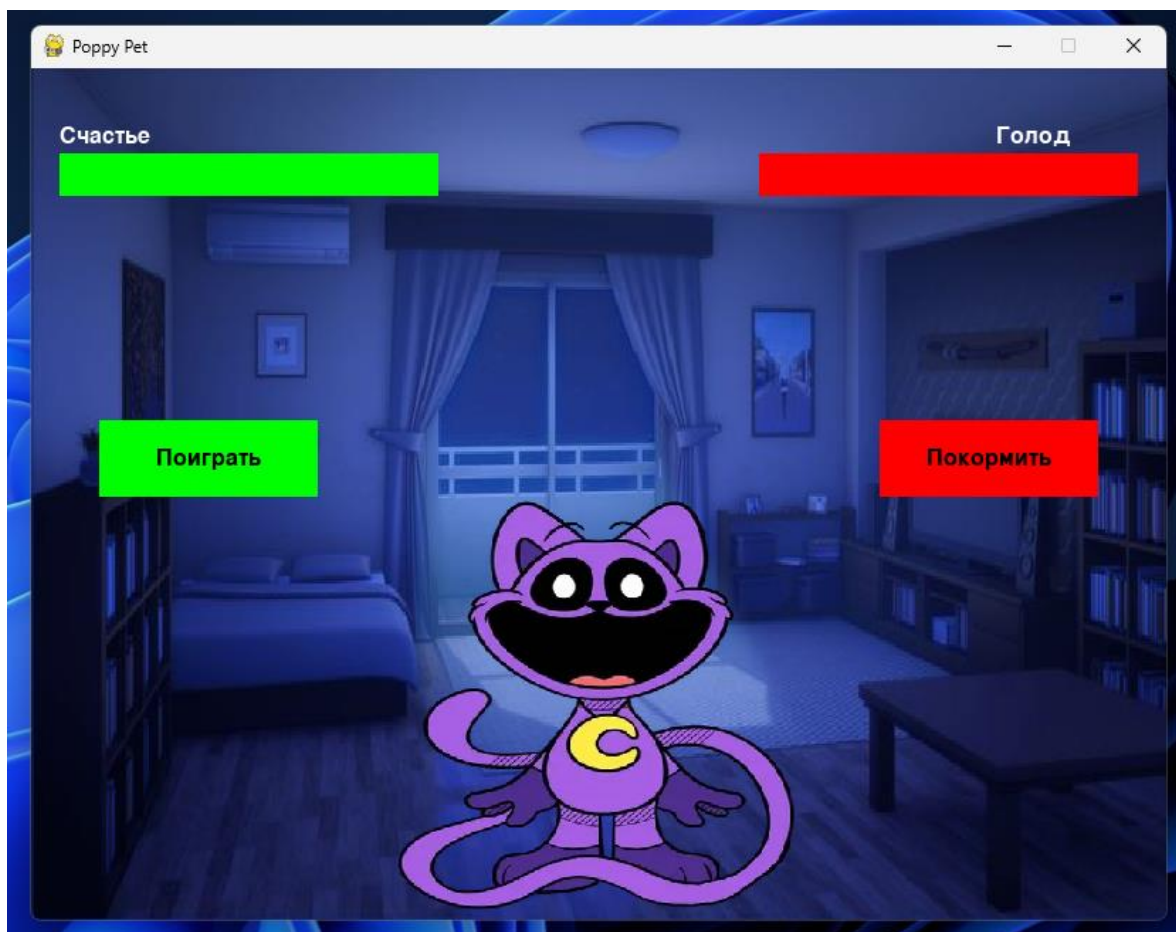
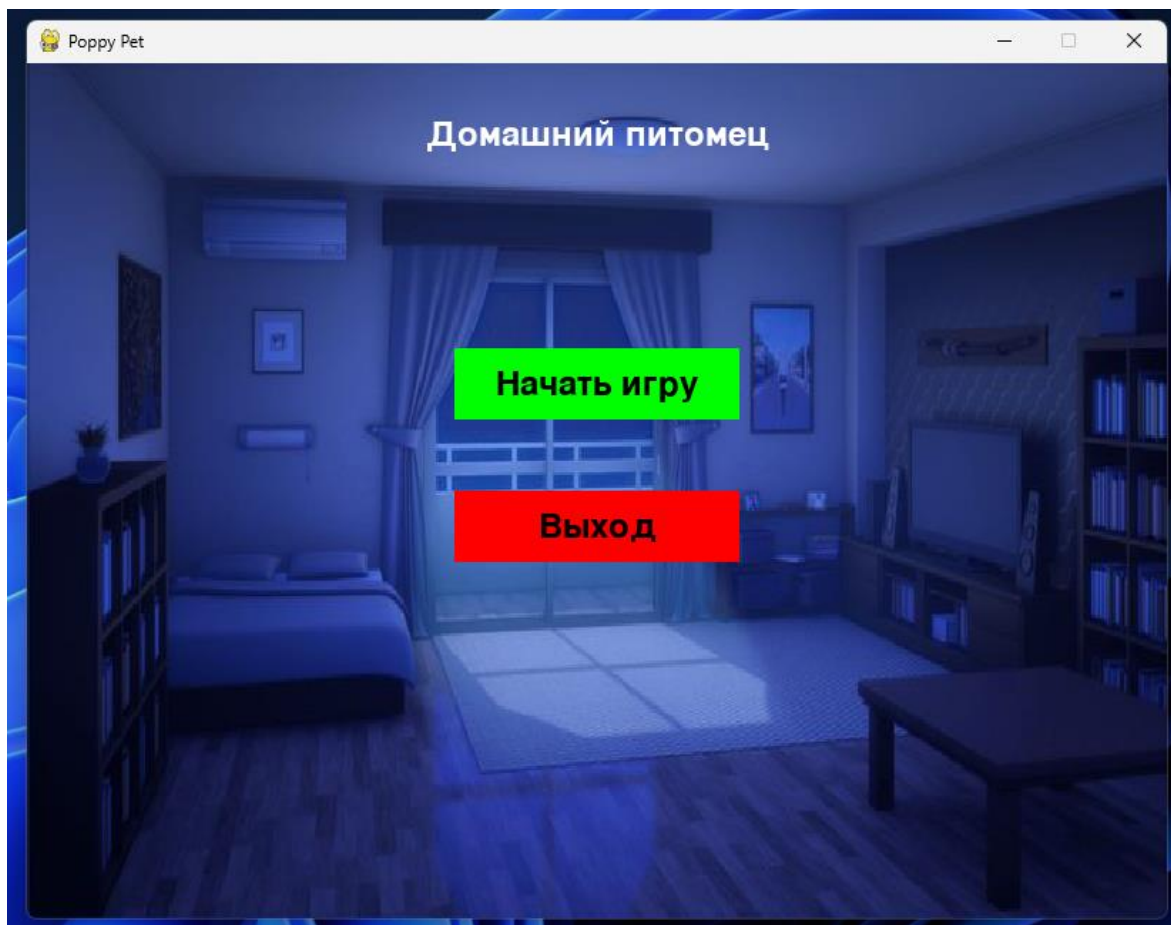
def main():
    pygame.init()

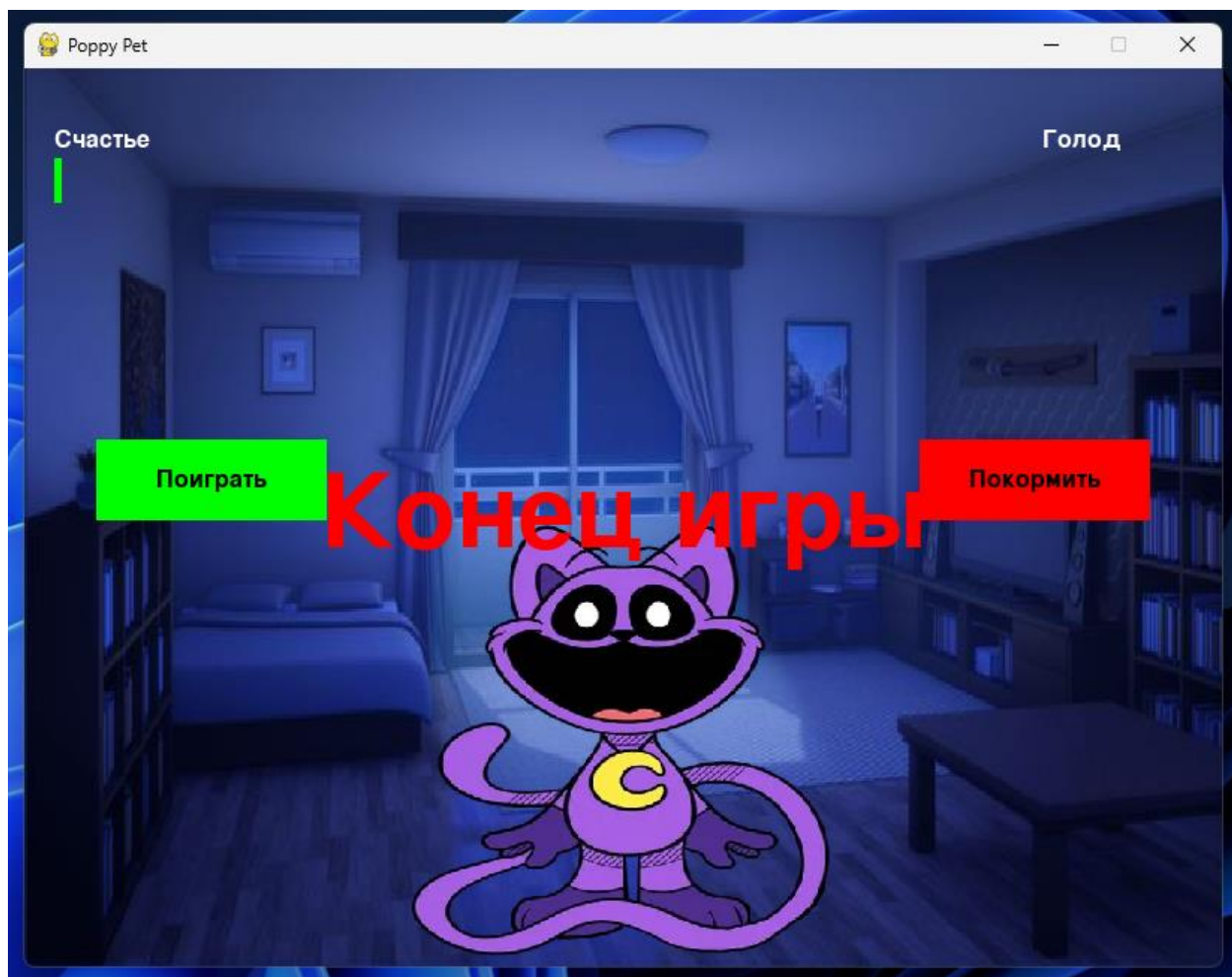
    size = weight, height = 800, 600

    menu = GameMenu(size)
    menu.draw_menu()

if __name__ == "__main__":
    main()
```

Результат работы программы:





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