***Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования***

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

**Факультет ИУ**

**Кафедра ИУ10**

**Отчет**

**по лабораторной работе № 3**

**Вариант**

**Дисциплина: ООП**

**Тема: Git, GitHub и pygame**

Работу выполнила: Захарова Арина ИУ10-22

Проверил: Буркацкий Кирилл Александрович

Москва, 2022

Оглавление

[**Отчет** 1](#_Toc99987246)

[**Цель работы:** 2](#_Toc99987247)

[**Ход работы** 2](#_Toc99987248)

[Задача №1 2](#_Toc99987249)

[*Цель:* 3](#_Toc99987250)

[*Ход решения задачи* 3](#_Toc99987251)

[*Вывод:* 4](#_Toc99987252)

[Задание 2 и 3 4](#_Toc99987253)

[*Цель:* 4](#_Toc99987254)

[*Ход решения задачи* 4](#_Toc99987255)

[*Вывод:* 7](#_Toc99987256)

[Пользовательский интерфейс и структура программы 7](#_Toc99987257)

[**Вывод по проделанной работе:** 7](#_Toc99987258)

**Цель работы:** в ходе решения данных задач познакомиться с библиотекой pygame.

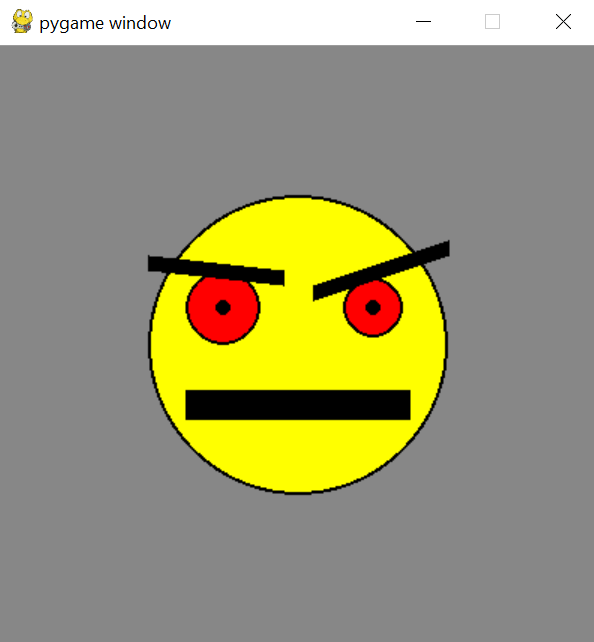
# **Ход работы**

## Задача №1

*Цель:* Нарисовать злой смайлик.

### *Ход решения задачи*

1. Написание программы
2. import pygame  
   from pygame.draw import \*  
     
   pygame.init()  
     
   FPS = 30  
   screen = pygame.display.set\_mode((400, 400))  
   screen.fill((135, 135, 135))  
     
   circle(screen, (255, 255, 0), (200, 200), 100) # мордочка  
   circle(screen, (0, 0, 0), (200, 200), 100, 2) # обводка мордочки  
   circle(screen, (255, 0, 0), (250, 175), 20) # глаз 1  
   circle(screen, (255, 0, 0), (150, 175), 25) # глаз 2  
   circle(screen, (0, 0, 0), (250, 175), 5) # зрачки  
   circle(screen, (0, 0, 0), (150, 175), 5)  
   circle(screen, (0, 0, 0), (250, 175), 20, 2) # обводка глаза 1  
   circle(screen, (0, 0, 0), (150, 175), 25, 2) # обводка глаза 2  
   rect(screen, (0, 0, 0), (125, 230, 150, 20)) # рот  
   polygon(screen, (0, 0, 0), ((190, 160), (100, 150), (100, 140), (190, 150))) # бровь 1  
   polygon(screen, (0, 0, 0), ((300, 130), (210, 160), (210, 170), (300, 140))) # бровь 2  
   pygame.display.update()  
   clock = pygame.time.Clock()  
   finished = False  
     
   while not finished:  
    clock.tick(FPS)  
    for event in pygame.event.get():  
    if event.type == pygame.QUIT:  
    finished = True  
     
   pygame.quit()
3. Тестирование



*Вывод:* разработана программа, рисующая смайлик.

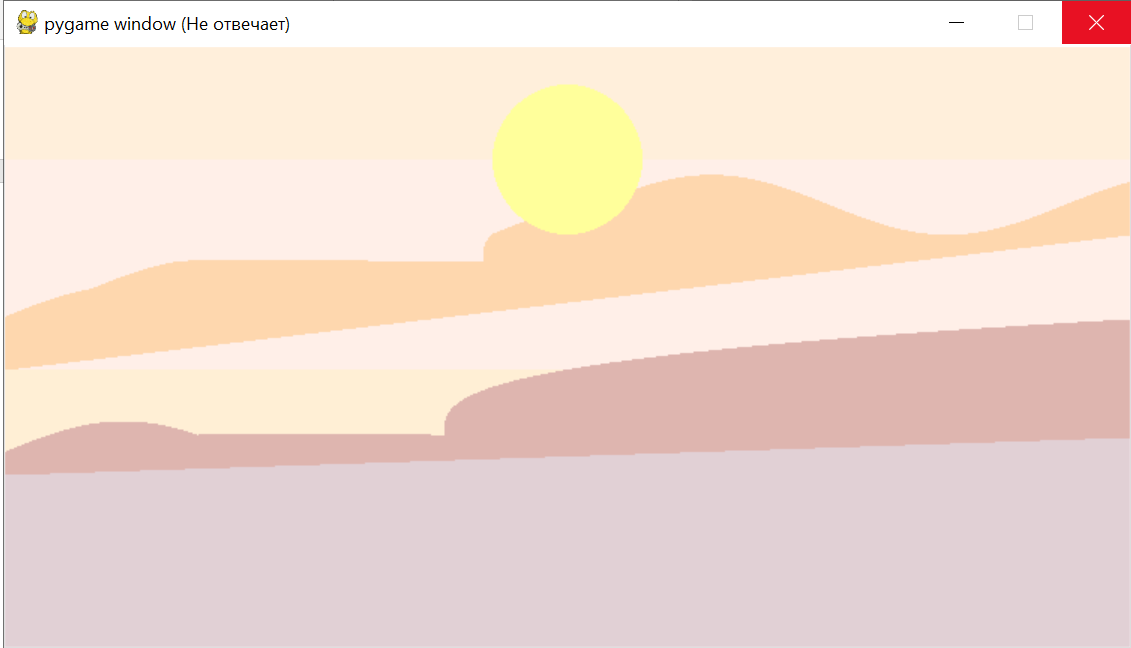
## Задание 2 и 3

*Цель:* разработать программу, рисующую горы с птичками или без.

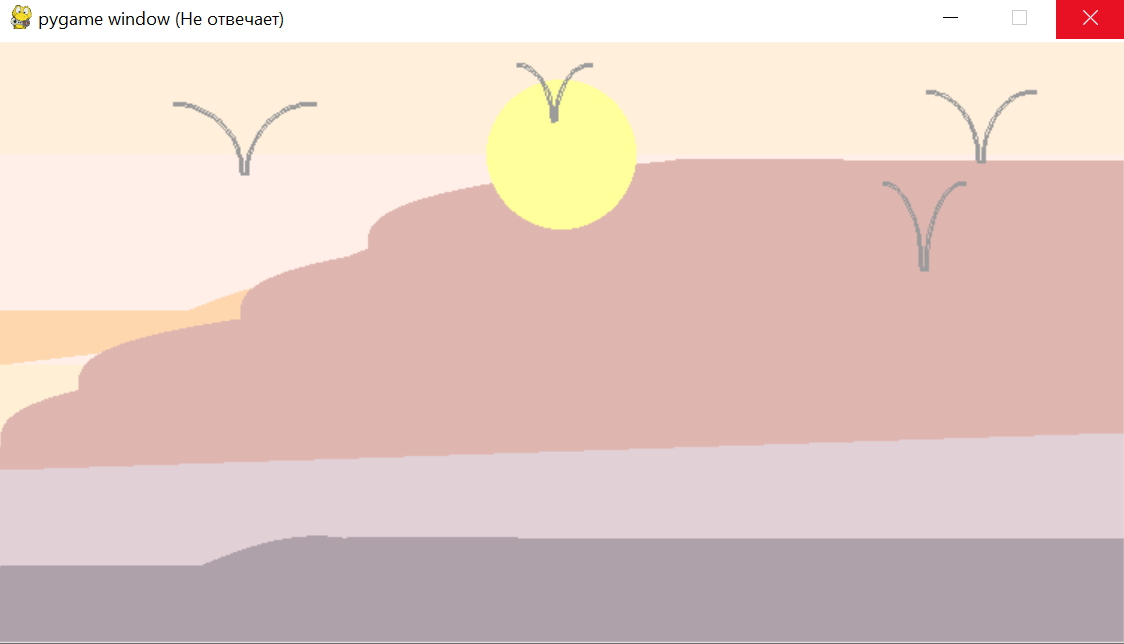
### *Ход решения задачи*

1. Написание программы
2. import pygame  
   import math  
   from pygame.draw import \*  
   from pygame.color import \*  
   from random import randint, choice  
   from math import pi  
     
   pygame.init()  
     
   FPS = 30  
   X = 750  
   Y = 400  
   screen = pygame.display.set\_mode((X, Y))  
   clock = pygame.time.Clock()  
   finished = False  
     
     
   def birds(n):  
    for i in range(n):  
    a = randint(50, 100)  
    b = randint(50, 150)  
    x = randint(0, X - 2 \* a)  
    y = randint(0, Y - 2 \* b)  
    arc(screen, (0, 0, 0), (x, y, a, b), 0, pi/2, width=3)  
    arc(screen, (0, 0, 0), (x + a, y, a, b), pi/2, pi, width=3)  
     
     
   def curve(y\_0, color): # функция, рисующая кривые гор  
    graph = [lambda x: 10 \* x \*\* (1 / 3), # графики, задающие кривые гор  
    lambda x: 20 \* math.sin(x/50),  
    lambda x: 50 \* math.sin(x/50) / (x + 1)]  
    x\_cords = [0]  
    y\_cords = [y\_0]  
    quantity\_curv = randint(3, 10) # генерация количества смен кривых  
    for i in range(quantity\_curv):  
    # выбор начальных и конечных значений координат для данной кривой  
    x\_start = x\_cords[-1]  
    x\_end = randint(10, (i + 1) \* X // quantity\_curv) if i != quantity\_curv - 1 else X  
    y\_start = y\_cords[-1]  
    current\_graph = choice(graph) # выбор кривой  
    for i in range(x\_start + 1, x\_end + 1):  
    x\_cords.append(i)  
    y\_cords.append(y\_start - current\_graph(i - x\_start))  
    cords = [(0, 400)] + [(x\_cords[i], y\_cords[i]) for i in range(len(x\_cords))] + [(X, Y)]  
    polygon(screen, color, tuple(cords))  
     
     
   def main\_picture(need\_birds): # отрисовка  
    screen.fill((254, 213, 162))  
    rect(screen, (254, 213, 196), (0, 75, X, Y - 75))  
    curve(180, (252, 152, 49))  
    polygon(screen, (254, 213, 196), ((0, 400), (0, 215), (X, 125), (X, Y)))  
    rect(screen, (254, 213, 148), (0, 215, X, Y - 215))  
    curve(270, (172, 67, 52))  
    polygon(screen, (179, 134, 148), ((0, 400), (0, 285), (X, 260), (X, Y)))  
    circle(screen, THECOLORS["yellow"], (X/2, 75), 50)  
    if need\_birds == '1': # если пользователь хочет картину с птичками   
    curve(350, (48, 16, 38))  
    birds(randint(3, 5))  
     
     
   if \_\_name\_\_ == "\_\_main\_\_":  
    print("""Привет!  
   Я РИСОВАТОР2000  
   Я рисую горные пейзажи с птицами или без них.  
   На твоей картине нужны птички?  
   1 - да/ другая кнопка - нет""")  
    main\_picture(input()) # вызов функции отрисовки гор  
    pygame.display.update()  
     
    while not finished:  
    clock.tick(FPS)  
    for event in pygame.event.get():  
    if event.type == pygame.QUIT:  
    finished = True  
     
    pygame.quit()
3. Написаны программы для рисования гор, с учётом требований пользователя.

Тест №1



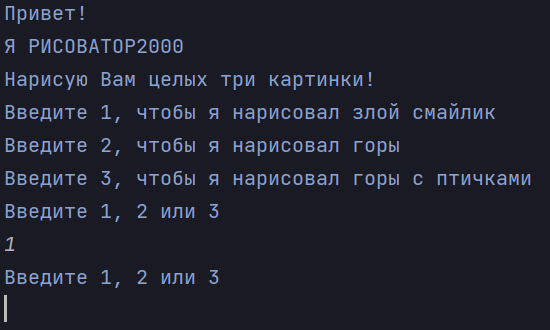
Тест №2



*Вывод:* разработана программа, рисующая картину гор, с учётом пожелания пользователя.

## Пользовательский интерфейс и структура программы

Для того, чтобы, программой было удобно пользователя был реализован интерфейс, где можно вызывать одну их трёх задач. Задачи разбиты на программы, разделенные по файлам в соответствии с исполняемым функционалом.



**Вывод по проделанной работе:** в ходе решения трёх поставленных задач были получены навыки по работе с библиотекой pygame, конечная версия проекта залита на github.