*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»*

*КАФЕДРА «ИУ-10»*

**Отчет**

**по лабараторной работе № 3**

**Дисциплина:** Объектно – ориентированное программирование

**Тема:** «PyGame и GitHub»

*Выполнила:*

Студент 1 курса группы ИУ10-22

Леденева Вера Сергеевна

*Проверил:* Буркацкий Кирилл Александрович

МОСКВА, 2022

# **ЦЕЛЬ РАБОТЫ**

Изучить работу с git и github, а также с библиотекой pygame.draw

# **ХОД РАБОТЫ**

## ***Задача №1:***

Написание вспомогательных функций и функций рисования изображений в файлы task1.py, task2.py, task3.py:

## *Код программы task1.py:*

import pygame  
from pygame.draw import \*  
  
pygame.init()  
FPS = 30  
  
  
def task1():  
 screen = pygame.display.set\_mode((500, 500))  
 screen.fill((255, 255, 255))  
 circle(screen, (255, 255, 0), (250, 250), 200)  
 circle(screen, (0, 0, 0), (250, 250), 200, 3)  
 circle(screen, (255, 0, 0), (175, 200), 40)  
 circle(screen, (0, 0, 0), (175, 200), 40, 3)  
 circle(screen, (255, 0, 0), (330, 200), 30)  
 circle(screen, (0, 0, 0), (330, 200), 30, 3)  
 circle(screen, (0, 0, 0), (175, 200), 15)  
 circle(screen, (0, 0, 0), (330, 200), 15)  
 rect(screen, (0, 0, 0), (150, 350, 200, 25))  
 line(screen, (0, 0, 0), (80, 100), (220, 175), 13)  
 line(screen, (0, 0, 0), (280, 180), (440, 130), 10)  
  
 pygame.display.update()

## *Код программы task2.py:*

import pygame  
from pygame.draw import \*  
  
pygame.init()  
FPS = 30  
  
  
def task2():  
 screen = pygame.display.set\_mode((800, 500))  
 screen.fill((255, 211, 153))  
 line(screen, (179, 134, 148), (0, 500), (800, 490), 300)  
 line(screen, (255, 226, 188), (0, 180), (800, 180), 120)  
 circle(screen, (252, 255, 87), (400, 120), 50)  
 line(screen, (250, 155, 47), (0, 280), (800, 180), 5)  
 polygon(screen, (250, 155, 47),  
 [(165, 108), (196, 119), (205, 140), (307, 220), (350, 210), (375, 230), (410, 200), (430, 210), (450, 190),  
 (470, 223), (165, 260)])  
  
 A = []  
 for i in *range*(0, 165, 1):  
 A.append((i, (-1 / 200) \* (i - 17) \*\* 2 + 217))  
 A.append((165, 260))  
 A.append((0, 280))  
 polygon(screen, (252, 153, 45), A)  
  
 B = []  
 for i in *range*(470, 580, 1):  
 B.append((i, (-1 / 4000) \* (i - 510) \*\* 3 + 180))  
 B.append((590, 205))  
 B.append((440, 225))  
  
 polygon(screen, (252, 153, 45), B)  
  
 C = []  
 for i in *range*(570, 620, 1):  
 C.append((i, (1 / 7) \* (i - 592) \*\* 2 + 70))  
 C.append((620, 200))  
 C.append((520, 215))  
 polygon(screen, (252, 153, 45), C)  
  
 polygon(screen, (252, 153, 45), [(610, 205), (595, 70), (670, 165), (700, 150), (740, 175), (770, 160), (800, 180)])  
  
 D = []  
 for i in *range*(0, 200, 1):  
 D.append((i, (1 / 120) \* (i - 80) \*\* 2 + 220))  
 D.append((200, 350))  
 D.append((0, 350))  
 polygon(screen, (172, 67, 52), D)  
  
 polygon(screen, (172, 67, 52),  
 [(180, 350), (225, 300), (270, 320), (320, 250), (370, 265), (400, 300), (460, 285), (460, 350), ])  
  
 E = []  
 for i in *range*(430, 550, 1):  
 E.append((i, (1 / 49) \* (i - 500) \*\* 2 + 250))  
 E.append((550, 320))  
 E.append((550, 350))  
 polygon(screen, (172, 67, 52), E)  
  
 F = []  
 for i in *range*(530, 610, 1):  
 F.append((i, (-1 / 128) \* (i - 610) \*\* 2 + 320))  
 F.append((610, 350))  
 F.append((530, 350))  
 polygon(screen, (172, 67, 52), F)  
  
 polygon(screen, (172, 67, 52),  
 [(610, 350), (590, 318), (630, 260), (680, 280), (700, 255), (750, 260), (800, 200), (800, 350)])  
  
 pygame.display.update()

## *Код программы task3.py:*

import pygame  
from pygame.draw import \*  
from task2 import task2  
  
screen = pygame.display.set\_mode((800, 500))  
pygame.init()  
FPS = 30  
  
  
def bird(*a*, *b*, *c*):  
 Q = []  
 W = []  
 for i in *range*(*a*, *a* + 15 \* *c*, 1):  
 Q.append((i, (0.08 / *c*) \* (i - *a* - 10 \* *c*) \*\* 2 + *b* - 15 \* *c*))  
 W.append((2 \* *a* - i, (0.08 / *c*) \* (i - *a* - 10 \* *c*) \*\* 2 + *b* - 15 \* *c*))  
 polygon(screen, (48, 16, 38), Q)  
 polygon(screen, (48, 16, 38), W)  
 Q = []  
 W = []  
  
  
def task3\_():  
 polygon(screen, (48, 16, 38), [(0, 500), (0, 250), (100, 270), (220, 395), (300, 500)])  
  
 G = []  
 for i in *range*(200, 450, 1):  
 G.append((i, (-1 / 250) \* (i - 370) \*\* 2 + 490))  
 G.append((450, 500))  
 G.append((270, 500))  
 polygon(screen, (48, 16, 38), G)  
  
 polygon(screen, (48, 16, 38), [(445, 500), (445, 467), (530, 420), (580, 435), (575, 500)])  
  
 H = []  
 for i in *range*(560, 800, 1):  
 H.append((i, (-1 / 300) \* (i - 600) \*\* 2 + 435))  
 H.append((800, 500))  
 H.append((550, 500))  
 polygon(screen, (48, 16, 38), H)  
  
 I = []  
 for i in *range*(650, 800, 1):  
 I.append((i, (1 / 100) \* (i - 865) \*\* 2 + 250))  
 I.append((800, 500))  
 I.append((600, 500))  
 polygon(screen, (48, 16, 38), I)  
  
  
def task3():  
 task2()  
 task3\_()  
 bird(320, 210, 1)  
 bird(350, 200, 2)  
 bird(420, 220, 2)  
 bird(420, 250, 2)  
 bird(550, 300, 2)  
 bird(530, 350, 3)  
 bird(600, 325, 2)  
 bird(600, 400, 2)  
  
 pygame.display.update()

## ***Задача №2:***

Написание файла main.py с вызовами функций из task1.py, task2.py, task3.py:

## *Код программы:*

from task1 import \*  
from task2 import\*  
from task3 import \*  
  
z = 0  
while z == 0:  
 task = *int*(*input*("Task № "))  
 if task == 1:  
 task1()  
 elif task == 2:  
 task2()  
 elif task == 3:  
 task3()  
 else:  
 *print*("Error")  
 z = *int*(*input*("Press 0 to continue "))

# 

# **ВЫВОД ПО ПРОДЕЛАННОЙ РАБОТЕ И ПРИМЕЧАНИЯ**

Мною был изучены и поняты основные принципы работы с git, github и pygame.

Все файлы лабораторной работы можно найти в репозитории

[VeraLedeneva/OOP\_2022 (github.com)](https://github.com/VeraLedeneva/OOP_2022)