

Лабораторная работа 1: Основы 2D графики

Цель:

Ознакомиться с основами представления объектов в 2D и работы с графическим ПО.

Задачи:

1. Используя библиотеку **Pygame**, реализовать простую 2D графику, отображающую основные геометрические фигуры: точки, линии, полигоны.
2. Создать простое приложение, где пользователь может рисовать на экране, используя мышь.
3. Сохранить созданное изображение в растровом формате (например, PNG).

Введение

В данной лабораторной работе студенты познакомятся с основами 2D графики с использованием библиотеки **Pygame**. Работа включает в себя создание простых геометрических фигур, разработку приложения для рисования и сохранение изображений в растровом формате.

Используемое ПО:

- библиотека **Pygame**
- среда разработки **PyCharm**

Порядок выполнения работы

1. Создание нового проекта в **PyCharm**

Запускаем среду разработки **PyCharm**. На главной странице выбираем вкладку **Projects**. В верхнем правом углу нажимаем кнопку **New project**. В списке слева выбираем **Pure Python**.

Называем новый проект и задаём ему расположение. В пункте **Interpreter type** выбираем **Custom environment**. В поле **Environment** выбираем **Generate new**, в поле **Type** – **Conda**. Выбираем последнюю версию **Python** и называем нашу новую виртуальную среду.

2. Установка **Pygame**

Для установки библиотеки необходимо в терминал написать команду **“pip install pygame”**. Убедитесь, что виртуальная среда активна. В терминале перед путём вашего текущего расположения в скобках должно быть название вашей виртуальной среды, например (gameEnv).

3. Создание скрипта для работы

Нажимаем правой кнопкой мыши по папке вашего проекта в левом списке. Выбираем пункт **New – Python file**.

4. Создание окна

Напишем код для создания основного окна приложения с заданными размерами и заголовком.

```
import pygame

def start():
    pygame.init()
    screen = pygame.display.set_mode((800, 600))
    pygame.display.set_caption("2D Graphics Lab")
```

4. Отображение фигур

Реализуем функции для рисования точек, линий и полигонов. Используйте различные цвета и размеры для фигур.

```
def draw_shapes(screen):
    # Рисуем точку
    pygame.draw.circle(screen, (255, 0, 0), (400, 300), 5) # Красная точка
    # Рисуем линию
    pygame.draw.line(screen, (0, 255, 0), (100, 100), (700, 500), 3) # Зеленая линия
    # Рисуем многоугольник
    pygame.draw.polygon(screen,
                        (0, 0, 255),
                        [(300, 200), (400, 100), (500, 200), (450, 300), (350, 300)]
                        ) # Синий многоугольник
```

4. Рисование с помощью мыши

Позволим пользователю рисовать на экране, удерживая левую кнопку мыши. Для этого создадим класс Drawer.

```
class Drawer:
    def __init__(self):
        self.drawing = False
        self.last_pos = None

    def handle_events(self, screen):
        for event in pygame.event.get():
            if event.type == pygame.QUIT:
                return False
            if event.type == pygame.MOUSEBUTTONDOWN:
                if event.button == 1: # Левая кнопка мыши
                    self.drawing = True
                    self.last_pos = event.pos
            if event.type == pygame.MOUSEBUTTONUP:
                if event.button == 1:
                    self.drawing = False
                if event.button == 3:
```

```
        self.save_image(screen)
    if event.type == pygame.MOUSEMOTION:
        if self.drawing:
            pygame.draw.line(screen, (144, 255, 144), self.last_pos, event.pos, 5) # Зеленая
линия
            self.last_pos = event.pos
    return True
```

5. Сохранение изображения

Добавим возможность сохранения состояния экрана в PNG. Для этого добавим в класс Drawer функцию `save_image`.

```
def save_image(self, screen):
    print('image saved')
    pygame.image.save(screen, "drawing.png")
```

6. Тестирование

Проверьте работоспособность приложения, исправьте обнаруженные ошибки.

Варианты

- 1 Рисование линий с выбором цвета. Пользователь может выбрать цвет линии перед рисованием.
- 2 Рисование кругов по клику мыши. Пользователь может выбрать радиус для круга и, кликнув на экране, нарисовать круг заданного радиуса.
- 3 Рисование прямоугольников с использованием двух кликов. Первый клик задает один угол, второй — противоположный.
- 4 Стирание нарисованных фигур. Пользователь может нажать левой кнопкой мыши, чтобы стереть нарисованную фигуру. Фигуру можно нарисовать или вручную, или с помощью кода.
- 5 Рисование многоугольников с заданным количеством вершин. Пользователь может задать количество вершин многоугольника перед рисованием.
- 6 Изменение толщины линий с помощью клавиатуры. Пользователь может использовать клавиши «←» и «+» для увеличения или уменьшения толщины линий.
- 7 Функция заливки фигур. Пользователь может выбрать цвет заливки и залить фигуру, кликнув внутри нее. Фигуру можно нарисовать или вручную, или с помощью кода.

- 8 Рисование эллипсов с цветом заливки. Пользователь может кликнуть на экране, чтобы нарисовать эллипс с заданным цветом заливки.
- 9 Изменение прозрачности фигур. Пользователь может задать уровень прозрачности для рисуемых фигур.