

# Компьютерная графика

## Лабораторная 1

# Отчет

- Титул
- Задача
- Основная теория
- Практическая реализация
- Заключение

# Init

```
import glfw  
from OpenGL.GL import *
```

```
delta = 0.1
```

```
angle = 0.0
```

```
posx = 0.0
```

```
posy = 0.0
```

```
size = 0.0
```

# Main

```
def main():
    if not glfw.init():
        return
    window = glfw.create_window(640, 640, "Lab1", None, None)
    if not window:
        glfw.terminate()
        return
    glfw.make_context_current(window)
    glfw.set_key_callback(window, key_callback)
    glfw.set_scroll_callback(window, scroll_callback)
    while not glfw.window_should_close(window):
        display(window)
    glfw.destroy_window(window)
    glfw.terminate()
```

# Display 1

```
def display(window):  
    global angle  
    glClear(GL_COLOR_BUFFER_BIT)  
    glLoadIdentity()  
    glClearColor(1.0, 1.0, 1.0, 1.0)  
    glPushMatrix()  
    glRotatef(angle, 0, 0, 1)  
    glBegin(GL_POLYGON)  
    glColor3f(0.1, 0.1, 0.1)  
    glVertex2f(posx + size + 0.5, posy + size + 0.5)  
    glColor3f(0.35, 0.0, 0.89)  
    glVertex2f(posx - size + -0.5, posy + size + 0.5)  
    glColor3f(0.0, 1.0, 1.0)  
    glVertex2f(posx - size + -0.5, posy - size + -0.5)  
    glColor3f(0.78, 0.23, 1.0)  
    glVertex2f(posx + size + 0.5, posy - size + -0.5)  
    glEnd()
```

# Display 2

```
glPopMatrix()
```

```
angle += delta
```

```
glfw.swap_buffers(window)
```

```
glfw.poll_events()
```

# Key callback

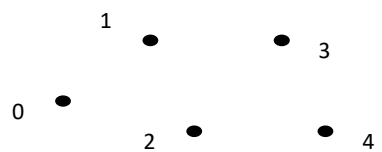
```
def key_callback(window, key, scancode, action,
mods):
    global delta
    global angle
    if action == glfw.PRESS:
        if key == glfw.KEY_RIGHT:
            delta = -3
        if key == 263: # glfw.KEY_LEFT
            delta = 3
```

# Scroll callback

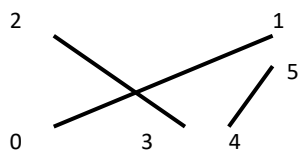
```
def scroll_callback(window, xoffset, yoffset):  
    global size  
    if (xoffset > 0):  
        size -= yoffset/10  
    else:  
        size += yoffset/10
```



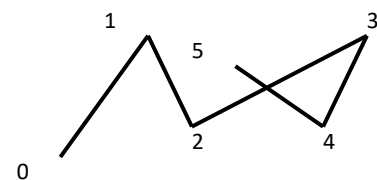
# Основные графические примитивы OpenGL



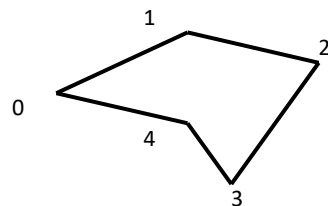
GL\_POINTS



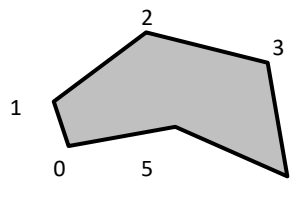
GL\_LINES



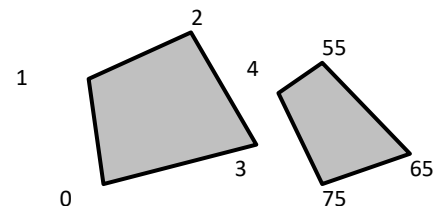
GL\_LINE\_STRIP



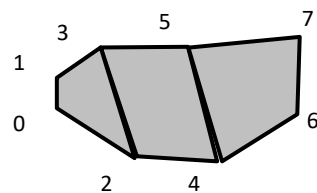
GL\_LINE\_LOOP



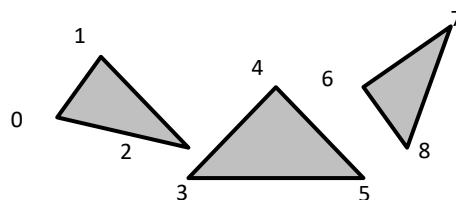
GL\_POLYGON



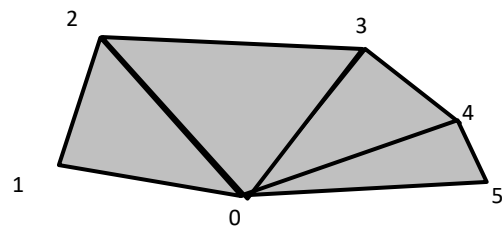
GL\_QUADS



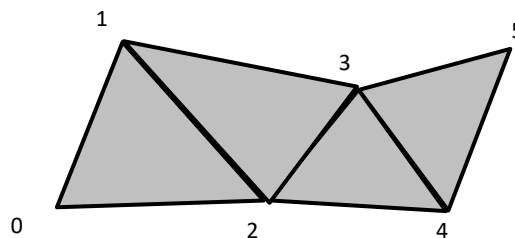
GL\_QUAD\_STRIP



GL\_TRIANGLES



GL\_TRIANGLE\_FAN



GL\_TRIANGLE\_STRIP

# Лабораторная номер 1

- Реализовать любой графический примитив
- Добавить любое геометрическое преобразование (сдвиг, поворот и т.д.)
- Добавить обработку события (нажатия на кнопку и т.д.)