

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНОМУ УНІВЕРСИТЕТІ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
Кафедра систем штучного інтелекту



Лабораторна робота
з дисципліни
«Комп'ютерна графіка»

Виконав:
студент групи КН-314
Ляшеник Остап
Викладач:
Лясковська С.Є.

2024 р.

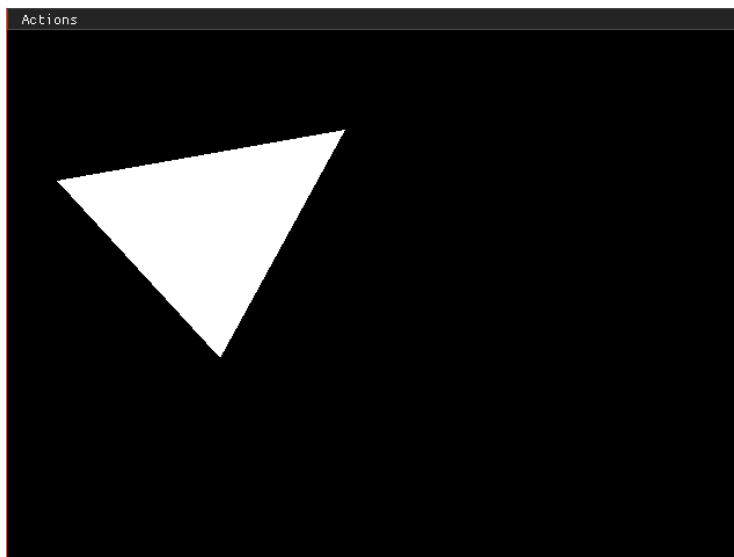
Мета роботи: ознайомитись з основами використання бібліотеки OpenGL і роботою з примітивами

Хід роботи

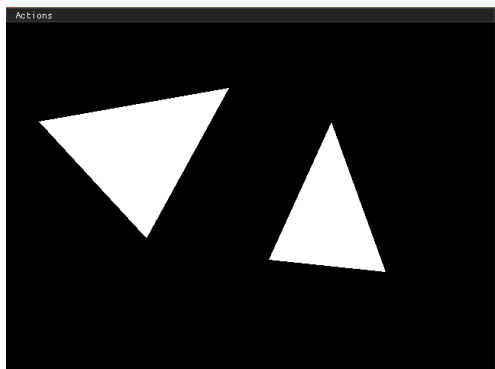
1. Відобразити у вікні набір примітивів (вершини яких задаються кліками миші) у

відповідності з варіантом.

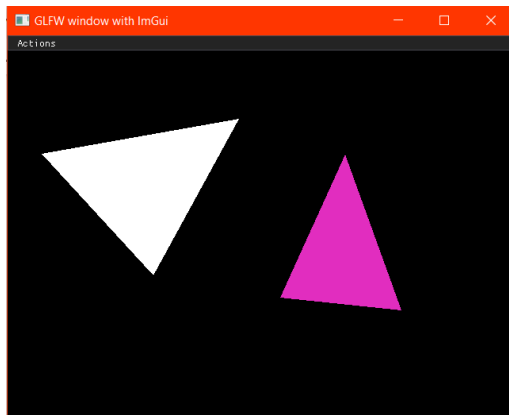
7	GL_TRIANGLE_FAN
---	-----------------



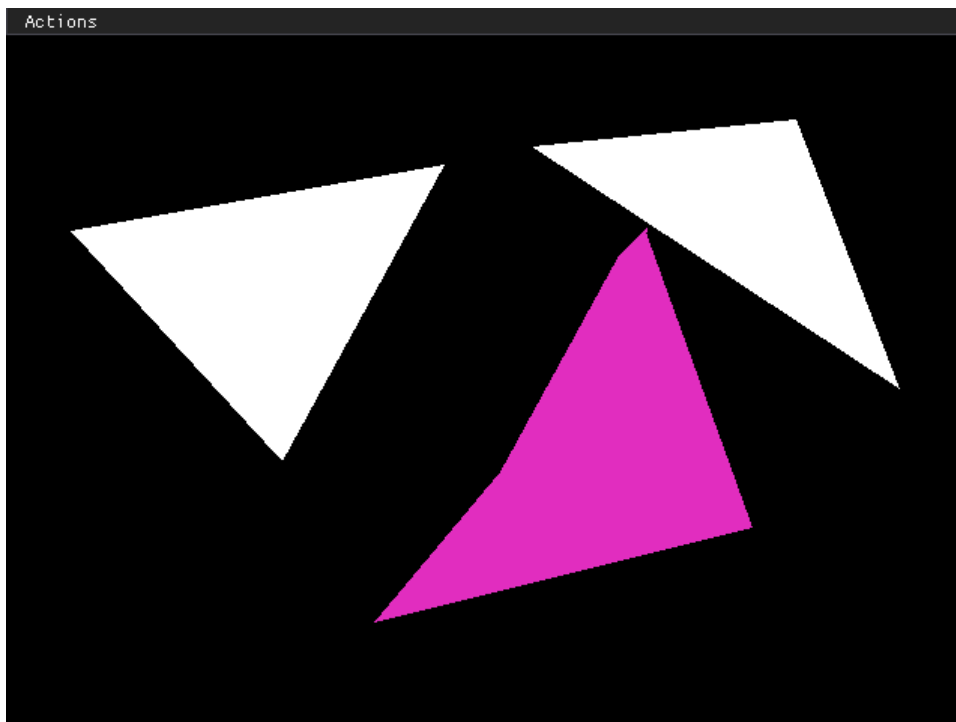
2. Для завершення відображення поточного набору примітивів і початку нового зарезервувати спеціальну клавішу (наприклад пробіл або правий клік миші).



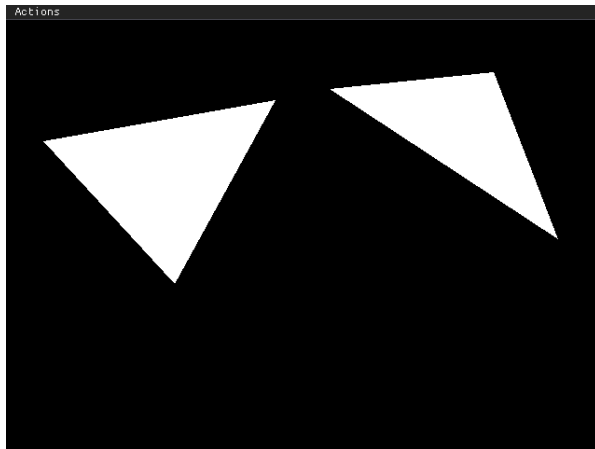
3. Для поточного набору примітивів надати можливість змінювати колір і координати його вершин.



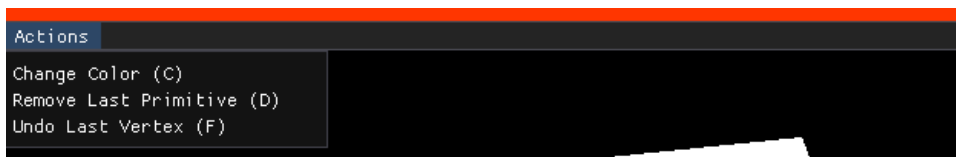
4. Поточний набір примітивів виділяти серед інших, наприклад, зміною розміру його вершин командою `glPointSize()`.



6. Передбачити можливість видалення останнього примітива і останнього набору примітивів.



7. Продублювати реалізовані команди в меню, яке створити за допомогою бібліотеки GLUT.



Функціональний код:

```
import glfw
from OpenGL.GL import *
import imgui
from imgui.integrations.glfw import GlfwRenderer
import random

class Primitive:
    def __init__(self):
        self.vertices = []
        self.color = [1.0, 1.0, 1.0]

if not glfw.init():
    raise Exception("glfw can not be initialized!")

window = glfw.create_window(640, 480, "GLFW window with ImGui", None,
None)
```

```
if not window:
    glfw.terminate()
    raise Exception("glfw window can not be created!")

glfw.set_window_pos(window, 400, 200)
glfw.make_context_current(window)

imgui.create_context()
impl = GlfwRenderer(window)

current_primitive = Primitive()
primitives = []

def window_resize(window, width, height):
    glViewport(0, 0, width, height)

glfw.set_window_size_callback(window, window_resize)

def add_vertex(x, y):
    global current_primitive
    current_primitive.vertices.append((x, y))

def change_color():
    global current_primitive
    current_primitive.color = [random.random() for _ in range(3)]

def remove_last_primitive():
    if primitives:
        primitives.pop()

def undo_last_vertex():
    if current_primitive.vertices:
```

```

        current_primitive.vertices.pop()

def mouse_button_callback(window, button, action, mods):
    if imgui.get_io().want_capture_mouse:
        return
    global current_primitive, primitives
    if button == glfw.MOUSE_BUTTON_LEFT and action == glfw.PRESS:
        x, y = glfw.get_cursor_pos(window)
        # Додавання координат вершини до поточного примітива
        current_primitive.vertices.append((x, y))
    elif button == glfw.MOUSE_BUTTON_RIGHT and action == glfw.PRESS:
        # Додавання поточного примітива до списку примітивів і
        # створення нового поточного примітива
        if current_primitive.vertices: # Переконаємося, що у
            примітива є хоча б одна вершина
            primitives.append(current_primitive)
            current_primitive = Primitive()

glfw.set_mouse_button_callback(window, mouse_button_callback)

def key_callback(window, key, scancode, action, mods):
    if imgui.get_io().want_capture_keyboard:
        return
    if key == glfw.KEY_C and action == glfw.PRESS:
        change_color()
    elif key == glfw.KEY_D and action == glfw.PRESS:
        remove_last_primitive()
    elif key == glfw.KEY_F and action == glfw.PRESS:
        undo_last_vertex()

glfw.set_key_callback(window, key_callback)

while not glfw.window_should_close(window):
    glfw.poll_events()

```

```

impl.process_inputs()

imgui.new_frame()

if imgui.begin_main_menu_bar():
    if imgui.begin_menu("Actions", True):

        if imgui.menu_item("Change Color (C)", None, False,
True)[0]:
            change_color()

        if imgui.menu_item("Remove Last Primitive (D)", None,
False, True)[0]:
            remove_last_primitive()

        if imgui.menu_item("Undo Last Vertex (F)", None, False,
True)[0]:
            undo_last_vertex()

    imgui.end_menu()
imgui.end_main_menu_bar()

glClear(GL_COLOR_BUFFER_BIT)
glLoadIdentity()
glOrtho(0, 640, 0, 480, -1, 1)

for primitive in primitives + [current_primitive]:
    glColor3fv(primitive.color)
    glBegin(GL_TRIANGLE_FAN)
    for x, y in primitive.vertices:
        glVertex2f(x, 480 - y)
    glEnd()

imgui.render()
impl.render(imgui.get_draw_data())
glfw.swap_buffers(window)

impl.shutdown()

```

```

glfw.terminate() import glfw
from OpenGL.GL import *
import imgui
from imgui.integrations.glfw import GlfwRenderer
import random

class Primitive:
    def __init__(self):
        self.vertices = []
        self.color = [1.0, 1.0, 1.0]

if not glfw.init():
    raise Exception("glfw can not be initialized!")

window = glfw.create_window(640, 480, "GLFW window with ImGui", None,
None)

if not window:
    glfw.terminate()
    raise Exception("glfw window can not be created!")

glfw.set_window_pos(window, 400, 200)
glfw.make_context_current(window)

imgui.create_context()
impl = GlfwRenderer(window)

current_primitive = Primitive()
primitives = []

def window_resize(window, width, height):
    glViewport(0, 0, width, height)

glfw.set_window_size_callback(window, window_resize)

```



```

def add_vertex(x, y):
    global current_primitive
    current_primitive.vertices.append((x, y))

def change_color():
    global current_primitive
    current_primitive.color = [random.random() for _ in range(3)]

def remove_last_primitive():
    if primitives:
        primitives.pop()

def undo_last_vertex():
    if current_primitive.vertices:
        current_primitive.vertices.pop()

def mouse_button_callback(window, button, action, mods):
    if imgui.get_io().want_capture_mouse:
        return
    global current_primitive, primitives
    if button == glfw.MOUSE_BUTTON_LEFT and action == glfw.PRESS:
        x, y = glfw.get_cursor_pos(window)
        # Додавання координат вершини до поточного примітива
        current_primitive.vertices.append((x, y))
    elif button == glfw.MOUSE_BUTTON_RIGHT and action == glfw.PRESS:
        # Додавання поточного примітива до списку примітивів і
        # створення нового поточного примітива
        if current_primitive.vertices: # Переконуємося, що у
            примітива є хоча б одна вершина
            primitives.append(current_primitive)
            current_primitive = Primitive()

```

```
glfw.set_mouse_button_callback(window, mouse_button_callback)
```

```
def key_callback(window, key, scancode, action, mods):  
    if imgui.get_io().want_capture_keyboard:  
        return  
    if key == glfw.KEY_C and action == glfw.PRESS:  
        change_color()  
    elif key == glfw.KEY_D and action == glfw.PRESS:  
        remove_last_primitive()  
    elif key == glfw.KEY_F and action == glfw.PRESS:  
        undo_last_vertex()
```

```
glfw.set_key_callback(window, key_callback)
```

```
while not glfw.window_should_close(window):  
    glfw.poll_events()  
    impl.process_inputs()
```

```
    imgui.new_frame()
```

```
    if imgui.begin_main_menu_bar():  
        if imgui.begin_menu("Actions", True):  
            if imgui.menu_item("Change Color (C)", None, False,  
True)[0]:  
                change_color()  
            if imgui.menu_item("Remove Last Primitive (D)", None,  
False, True)[0]:  
                remove_last_primitive()  
            if imgui.menu_item("Undo Last Vertex (F)", None, False,  
True)[0]:  
                undo_last_vertex()  
        imgui.end_menu()
```

```
    ImGui.EndMainMenuBar()

    glClear(GL_COLOR_BUFFER_BIT)
    glLoadIdentity()
    glOrtho(0, 640, 0, 480, -1, 1)

    for primitive in primitives + [current_primitive]:
        glColor3fv(primitive.color)
        glBegin(GL_TRIANGLE_FAN)
        for x, y in primitive.vertices:
            glVertex2f(x, 480 - y)
        glEnd()

    ImGui.Render()
    impl.render(ImGui.GetDrawData())
    glfw.swap_buffers(window)

impl.shutdown()
glfw.terminate()
```

Висновок:

В ході роботи була опрацьована бібліотека OpenGL, виявлені проблеми та знайдені оптимальні рішення до виконання завдання.