**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:**

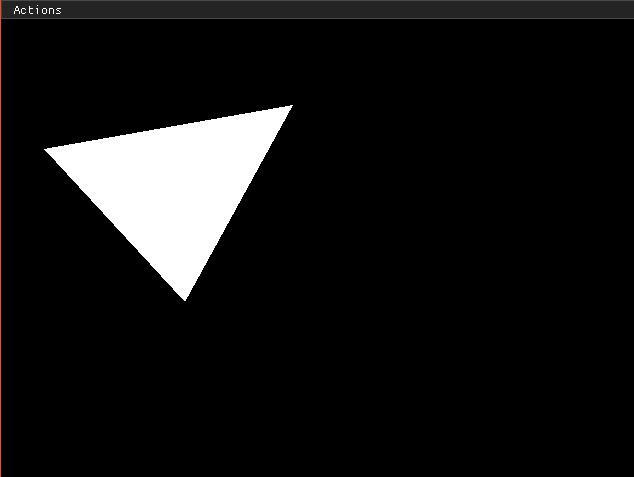
Лясковська С.Є.

2024 р.

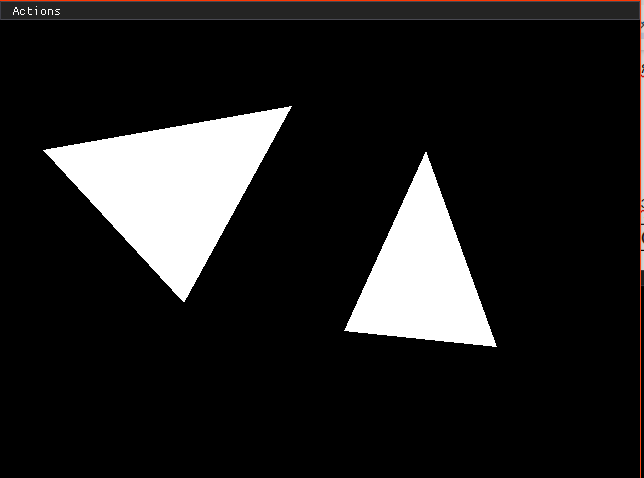
**Мета роботи**: ознайомитись з основами використання бібліотеки OpenGL і роботою з примітивами

**Хід роботи**

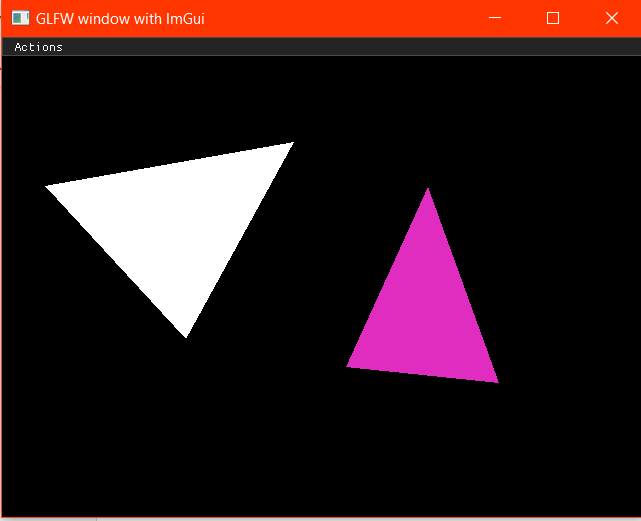
1. Відобразити у вікні набір примітивів (вершини яких задаються кліками миші) у відповідності з варіантом.



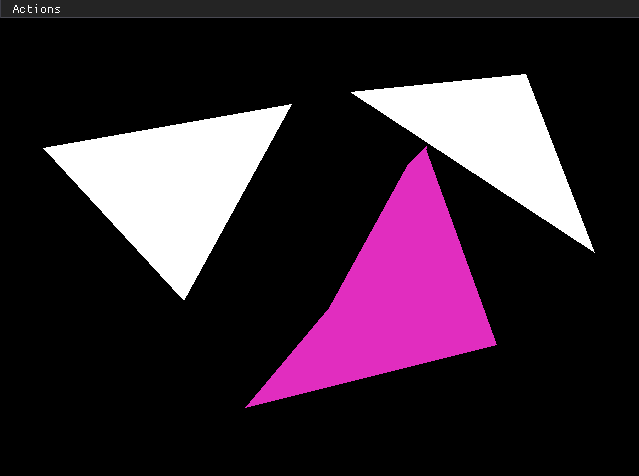
2. Для завершення відображення поточного набору примітивів і початку нового зарезервувати спеціальну клавішу (наприклад пробіл або правий клік миші).



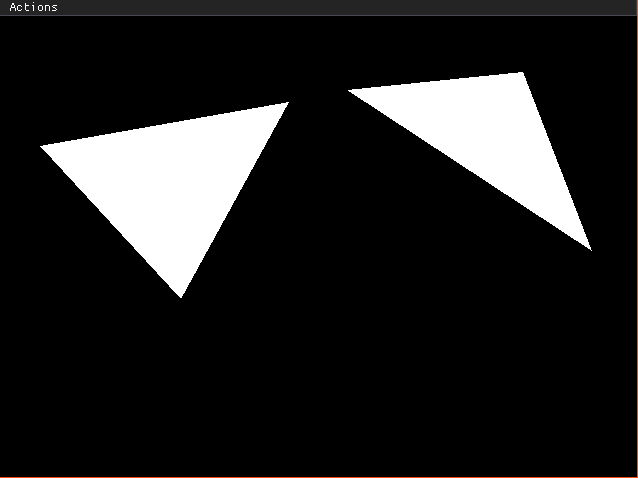
3. Для поточного набору примітивів надати можливість змінювати колір і координати його вершин.



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: # 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.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.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()

**Висновок:**

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