

1) Instrukcja obsługi programu:

Program, można uruchomić poprzez wybrane środowisko IDE (W moim przypadku było to Visual Studio 2019) dzięki plikowi **OpenGL2.sln**. Natomiast nie zalecam korzystania z tego sposobu ponieważ wymaga ono dodatkowego zlinkowania wszystkich plików bibliotek przedstawionych w 3 punkcie. **Zalecam natomiast uruchamiać program poprzez dołączony plik OpenGL2.exe, znajdujący się w folderze EXE.** W samej aplikacji sterujemy przy pomocy następujących przycisków:

W – Poruszanie do przodu

S – Poruszanie do tyłu

A – Poruszanie w lewo

D – Poruszanie w prawo

Mysz – Poruszanie myszą pozwala na obracanie kamerą według kierunku myszy

2) Sposób konfiguracji programu:

W programie, można dowolnie modyfikować istniejącą scenę czy sposób widzenia dzięki zaimplementowanym funkcjom. Przede wszystkim, można manipulować dowolnymi parametrami kamery. Ustawia się je w linii 65 pliku **Opengl2.cpp** oraz samej klasie **Camera.hpp**. Jeżeli chcemy natomiast modyfikować wygląd sceny to wszystko ustawia się w pliku **Database.hpp**. W celu dodania nowej tekstury, tworzy się nowy wektor obiektów klasy **Texture**, w linii 3, a następnie dodajemy do niego obiekt tejże klasy w konstruktorze klasy **TextureDatabase** według parametrów jej konstruktora. Następnie jeżeli chcielibyśmy by nasz obiekt był stworzony na podstawie tablicy vertexów i ewentualnie indeksów wymagane jest utworzenie nowych wektorów struktury **Vertex** oraz **GLuint** w linii 45 i 46. Podobnie jak w przypadku tekstur, ponownie dodajemy obiekt do wektora w obrębie konstruktora klasy **VertexDatabase** według wskazanych argumentów. By utworzyć już sam model w programie, dodajemy kolejny obiekt klasy:

1. **Mesh**, jeżeli chcemy by nasz obiekt był zwykłym elementem na scenie
2. **Light**, jeżeli chcemy by nasz obiekt był źródłem światła
3. **Window**, jeżeli chcemy by nasz obiekt był częściowo przezroczysty

W każdym z wymienionych klas postępuje wedle wskazanych parametrów konstruktorów bazując na poprzednio dodanych teksturach czy zbiorach vertexów i indeksów. Dodatkowo każdym obiektem z osobna, można dowolnie manipulować na podstawie wywoływania metod **SetRotation**, **SetPosition** oraz **SetScale** na rzecz tego obiektu. Ten element również wykonujemy w obrębie konstruktora.

3) Wymagania programu:

Program w celu uruchomienia go poprzez program IDE wymaga dodatkowego podłączenia następujących bibliotek w ustawieniach projektu:

1. GLFW
2. GLEW
3. GLM
4. stb

4) Przykłady działania ze zrzutami ekranu :

