|  |  |  |  |
| --- | --- | --- | --- |
| **Politechnika Świętokrzyska w Kielcach**  **Wydział Elektrotechniki, Automatyki i Informatyki**  **Katedra Informatyki, Elektroniki i Elektrotechniki** | | | |
| Kierunek  **Informatyka** | Laboratorium  **Grafika komputerowa** | | |
| Temat ćwiczenia  **Silnik 3D** | | Wykonał:  Dominik Grudzień  Patryk Grzywacz |
| Grupa dziekańska  **3ID11B** |
| Data wykonania  29.01.2022r | | Data oddania  29.01.2022r | Ocena i podpis |

1. *Cel projektu*

Celem projektu było wykonanie silnika 3D za pomocą wybranej biblioteki graficznej i zaprezentowanie jego funkcjonalności na demie lub napisanie na jego podstawie gry. Na koniec semestru należało również oddać sprawozdanie w formacie PDF oraz techniczną dokumentacje kodu.

1. *Wybrane technologie*

Do wykonania tego projektu użyliśmy biblioteki FrameWorka GLFW do OPENGL, środowiska C-Lion 2021, języka C++, programu Doxygen do wygenerowania dokumentacji oraz GitHub’a jako repozytorium kodu.

1. *Wnioski*

Do projektu udało się wykonać bardzo dużo funkcjonalności, w skład które wchodzą:

* Zaimplementowanie klasy silnika wyposażonego w funkcjonalności tk.j.:
  + inicjacja biblioteki graficznej (wykonana w środowisku C-LION z CMake),
  + parametryzowanie trybu graficznego (np. w oknie, pełny ekran, wybór rozdzielczości),
  + parametryzowanie innych parametrów (np. liczba klatek animacji na sekundę, uruchomienie obsługi myszy/klawiatury),
  + główna pętla gry korzystająca z czasomierza,
  + obsługa klawiatury i myszy,
  + obsługa czyszczenia ekranu do zadanego koloru,
  + obsługa błędów z możliwością wyświetlania (
  + zamknięcie gry (deinicjacja biblioteki graficznej, sprzątanie pamięci).
  + obracanie kamery myszą
  + przemieszczanie za pomocą klawiszy WASD z dwoma prędkościami (z klawiszem SHIFT)
  + zmiana perspektywy widzenia ( ortograficzna oraz perspektywistyczna za pomocą klawisza P)
  + zamykanie i czyszczenie aplikacji przyciskiem ESCAPE
* Zastosowanie wielokrotnego buforowania i głębi bitowej w silniku
* Zaimplementowanie klasy Sphere, która rysuje obiekty kuliste opisane za pomocą pozycji 4 wymiarowych vektorów, shaderów vertexowych oraz fragmentowych, nakładania na nie tekstur oraz transformacji.
* Napisanie shaderów GLSL w wersji 330 core do obliczeń vertexowych (geometria i pozycje) oraz fragmentowych do obliczeń kolorów pikseli.

Kod silnika kompiluje się oraz włącza i zamyka się bezbłędnie.