

# Układ Słoneczny

Sławomir Stuglik, Jarosław Czepczor

19 czerwca 2017

## 1 Wstęp

### 1.1 Wprowadzenie

- Raport został stworzony w celu przedstawienia założeń projektu "Układ Słoneczny" i jego kroków realizacji.
- Program został napisany przy użyciu programu OpenGL w C++.
- Jest to realistyczna wizualizacja 3D.

### 1.2 Cel projektu

- Zamierzeniem projektu była wizualizacja Układu Słonecznego zgodnie z grawitacją Newtona.
- Otekstrowanie planet.
- Realistyczna symulacja z aktualnymi parametrami orbit.

## 2 Możliwości wizualizacji:

- Podróż po wirtualnym Układzie Słonecznym.
- Możliwość obserwacji planet z bliska.
- Obserwacja wszechwiata o 360 stopni.
- Powiększanie i zmniejszanie planet.
- Wyświetlanie orbit po których krążą planety.

### 3 Opis kodu programu:

- Główny plik projektu to main.cpp do którego dołączone zostały moduły są to:
- camera-odpowiada za widok i obserwację układu.
- moon-służy za utworzenie Księżyca, tylko dla Ziemi.
- planet-możliwość tworzenia planet.
- skala-skala całego układu(w tym przypadku 1:100000000)
- solarsystem-moduł odpowiedzialny za dodawanie planet i Księżyców i ich pozycjonowanie.
- tga-moduł pozwalający na teksturowanie obiektów

```

int main(int argc, char** argv)
{
    glutInit(&argc, argv); // inicjalizacja biblioteki GLUT
    glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB|GLUT_DEPTH);
    glutInitWindowSize(1200, 750); // rozmiary gŁ,Åtwnego okna programu
    glutInitWindowPosition(100, 100);
    glutCreateWindow( "Uklad Sloneczny" ); // utworzenie gŁ,Åtwnego okna programu z nazwa
    init();
    glutDisplayFunc( display ); // doŁ,Åczenie funkcji generujÅcej scenÅ 3D
    glutReshapeFunc( reshape ); // doŁ,Åczenie funkcji wywoŁ,ywanej przy zmianie rozmiaru okna
    glutKeyboardFunc(keyDown);
    glutKeyboardUpFunc(keyUp);
    glutMainLoop(); // wprowadzenie programu do obsŁ,ugi pÅ™tli komunikatÅw - coŁ> jak gŁ,Åtwna petla GLUT
    return 0;
}

// wczytanie tekstur
help = new TGA("images/menu.tga");
tlo = new TGA("images/tlo.tga");
moon = new TGA("images/moon.tga");

TGA* slonce = new TGA("images/slonce.tga");
TGA* merkury = new TGA("images/merkury.tga");
TGA* wenus = new TGA("images/wenus.tga");
TGA* earth = new TGA("images/earth.tga");
TGA* mars = new TGA("images/mars.tga");
TGA* jowisz = new TGA("images/jowisz.tga");
TGA* saturn = new TGA("images/saturn.tga");
TGA* uran = new TGA("images/uran.tga");
TGA* neptun = new TGA("images/neptun.tga");

// dodanie wszystkich planet z dokŁ,adnymi danymi. OdlegŁ,oŁ>Å mierzona w km, czas mierzony w dniach z
//1. (odlegŁ,osc od odbiektu przyciagania - gdy jest to planeta to obiekt przyciagany przez slonce)
//2. czas pelnego pokonania drogi wokol slonca (po orbicie)
//3. czas obrotu wokol wlasnej osi
//4. promien
UkladSloneczny.addPlanet(0, 1, 500, 695500, slonce->getTextureHandle()); // slonce
UkladSloneczny.addPlanet(57910000, 88, 58.6, 2440, merkury->getTextureHandle()); // merkury
UkladSloneczny.addPlanet(108200000, 224.65, 243, 6052, wenus->getTextureHandle()); // wenus
UkladSloneczny.addPlanet(149600000, 365, 1, 6371, earth->getTextureHandle()); // ziemia
UkladSloneczny.addPlanet(227939100, 686, 1.03f, 3389, mars->getTextureHandle()); // mars
UkladSloneczny.addPlanet(778500000, 4332, 0.4139, 69911, jowisz->getTextureHandle()); // jowisz
UkladSloneczny.addPlanet(1433000000, 10759, 0.44375, 58232, saturn->getTextureHandle()); // saturn
UkladSloneczny.addPlanet(2877000000, 30685, 0.718056, 25362, uran->getTextureHandle()); // uran
UkladSloneczny.addPlanet(4503000000, 60188, 0.6713, 24622, neptun->getTextureHandle()); // neptun

UkladSloneczny.addMoon(3, 7000000, 27.3, 27.3, 1738, moon->getTextureHandle()); // ksiezyc ziemi

```

## 4 Bibliografia

1. Opis planet
2. Programy z zajęc