Sprawozdanie – Symulacja Układu Słonecznego w OpenGL Maciej Gołębiowski 280693

1. Cel projektu

Celem projektu było stworzenie trójwymiarowej symulacji Układu Słonecznego w OpenGL z realistycznymi orbitami, obrotami planet wokół własnej osi, ruchem księżyca Ziemi oraz pierścieniami Saturna. Dodatkowo wdrożono sterowanie kamerą i obsługę tekstur dla wszystkich ciał niebieskich.

2. Środowisko i narzędzia

- **Język:** C++
- **Biblioteka graficzna:** OpenGL, FreeGLUT
- **Obsługa tekstur:** stb_image.h
- **System operacyjny:** Windows/Linux

3. Założenia fizyczne

3.1 Promienie i odległości

Wszystkie wartości, takie jak promienie planet i odległości orbitalne, zostały podane w jednostkach umownych. Dzięki temu wszystkie planety mieszczą się w scenie i są dobrze widoczne. Skala została dobrana wizualnie, aby zachować proporcje między planetami i umożliwić płynne obserwowanie ich ruchu.

Planeta	Promień [j.u.]	Odległość od Słońca [j.u.]
Merkury	0.25	4.0
Wenus	0.6	6.0
Ziemia	0.65	8.0
Mars	0.35	10.0
Jowisz	1.4	13.5
Saturn	1.2	17.0

Uran	0.9	20.5
Neptun	0.85	24.0
Księżyc	0.18	1.5 (od Ziemi)

3.2 Prędkości orbitalne i obroty wokół własnej osi

Wartości dobrano tak, aby ruch był realistyczny wizualnie:

Planeta	**Prędkość orbitalna [°/s]**	**Prędkość obrotu wokół osi [°/s]**
Merkury	4.15	6.0
Wenus	1.62	-2.0
Ziemia	1.0	5.0
Mars	0.53	4.0
Jowisz	0.083	8.0
Saturn	0.033	7.0
Uran	0.012	6.0
Neptun	0.006	5.0
Księżyc	5.0	0.0

3.3 Ekscentryczność orbit

Orbity niektórych planet nie są idealnie kołowe:

- Merkury: 0.205
- Wenus: 0.006
- Ziemia: 0.017
- Mars: 0.093
- Jowisz: 0.049
- Saturn: 0.056
- Uran: 0.047
- Neptun: 0.009
- Księżyc: 0.055

aktualna odległość planety od środka orbity obliczana wzorem:

$$r(\theta) = \frac{a(1 - e^2)}{1 + e \cdot \cos(\theta)}$$

gdzie:

- a – półoś wielka elipsy
- e – ekscentryczność
- θ – kąt orbitalny w radianach

4. Implementacja graficzna

4.1 Planety i księżyc

Każda planeta i księżyc są reprezentowane jako sfery (gluSphere) z przypisaną teksturą. Ich **pozycja w przestrzeni 3D** nie jest statyczna – obliczana jest dynamicznie na podstawie ich ruchu orbitalnego wokół Słońca (lub Ziemi w przypadku Księżyca).Do tego celu użyto wzorów geometrycznych opisujących **eliptyczną orbitę**:

```
float r = (a*(1-e*e)) / (1 + e*cos(theta));
float x = r*cos(theta);
float z = r*sin(theta);
glTranslatef(x, 0, z);
```

- a – **wielka półoś orbity**. Jest to średnia odległość planety od Słońca w naszej scenie, wyrażona w jednostkach umownych (skalowanych tak, aby wszystkie planety mieściły się w widocznej scenie).
- e – **mimośród orbity**, określający, jak bardzo orbita różni się od koła (e=0 oznacza orbitę kołową, większe e – bardziej eliptyczną).
- theta – **kąt orbitalny**, zmieniający się w czasie symulacji:

$$\theta = \text{timeSim} \times \text{orbitSpeed}$$

4.2 Oświetlenie

- Źródło światła w pozycji Słońca (0,0,0)
- Typ światła: GL_LIGHT0, włączone: diffuse i specular
- Planety reagują na światło dzięki GL_MODULATE i GLU_SMOOTH

4.3 Kamera

Obsługa dwóch trybów:

1. **Wolna kamera** – sterowanie: WASD (ruch), strzałki (obrót), +/- (zoom), mysz (obrót)

2. **Kamera na Ziemi** – C przełącza na widok z powierzchni Ziemi, śledzi ruch księżyca

5. Sterowanie

= / -: przybliżanie / oddalanie
W / S: ruch do przodu / do tyłu
A / D: ruch w lewo / w prawo
Strzałki: obrót kamery
Lewy przycisk + ruch myszy : obrót kamery myszką
C: przełączanie na kamerę na Ziemi
ESC: wyjście z programu

6. Uwagi i obserwacje

- **Skalowanie i proporcje**
  - Wszystkie planety i odległości zostały przeskalowane dla czytelności w wizualizacji 3D.
  - Promienie planet i orbit nie odpowiadają dokładnie rzeczywistym wartościom astronomicznym, aby obiekty były widoczne w oknie programu.
- **Ruch planet i księżyca**
  - Planety poruszają się po orbitach eliptycznych, wyznaczonych równaniem: $$r = \frac{a(1-e^2)}{1+e\cos(\theta)}$$
  - Prędkości orbitalne i obroty wokół własnej osi zostały dobrane tak, aby były widoczne wizualnie, Przeskalowane z rzeczywistymi wartościami astronomicznymi, lecz nie w pełni zgodne z nimi. Przykład: W programie Ziemia: orbitSpeed = 1.0, Merkury: orbitSpeed = 4.15. Relacja ≈ 4:1, czyli Merkury krąży ~4 razy szybciej niż Ziemia w animacji. To jest przeskalowane dla wizualizacji, ale odwzorowuje różnice w czasie obrotu planet.
  - Księżyc krąży wokół Ziemi, nie ma własnej orbitowanej osi wokół Słońca.
- **Oświetlenie**
  - Światło pochodzi ze Słońca, ustawione jako punktowe źródło w centrum układu.
  - Wszystkie planety reagują na światło (diffuse i specular), co pozwala na realistyczne cieniowanie i efekt dnia i nocy na planetach.

- **Sterowanie kamerą**
    - Kamera może się obracać myszką i za pomocą strzałek.
    - Kamera może przybliżać i oddalać scenę (= / -) oraz przesuwać się (W/A/S/D).
    - Dodatkowo zaimplementowano kamerę “na Ziemi” – po wciśnięciu C kamera ustawia się na powierzchni Ziemi i podąża za jej ruchem orbitalnym.
- **Ograniczenia i uproszczenia**
    - Obroty planet wokół własnej osi są przybliżone i dobrane dla efektu wizualnego.
    - Odległości i promienie orbit nie są w skali astronomicznej.
    - Nie uwzględniono grawitacyjnych interakcji między planetami ani efektów atmosferycznych.
    - Animacja jest uproszczona, ale pozwala na wizualną demonstrację ruchu planet i księżyca.
- **Wnioski**
    - Program pozwala wizualizować Układ Słoneczny w 3D, z realistycznym oświetleniem i animacją ruchu planet.
    - Wprowadzenie kamery na powierzchni Ziemi daje możliwość obserwacji z perspektywy planety.
    - Użycie tekstur i shaderów (modulacja) zwiększa wrażenie realizmu sceny.