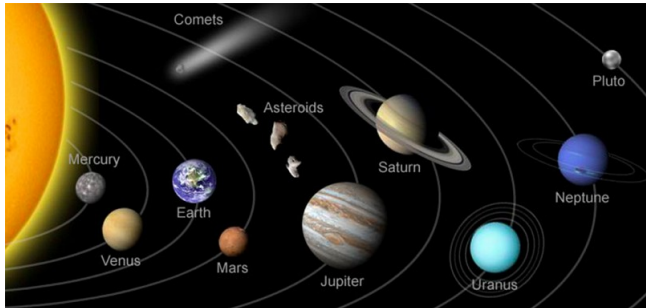


Symulacja Układu Słonecznego

Noga Wojciech, Grzegorz Wolanin, Krzysztof Jaje

19 czerwca 2017

- 1 Opis problemu
- 2 Widoki z programu
- 3 Kodu
- 4 Uwagi



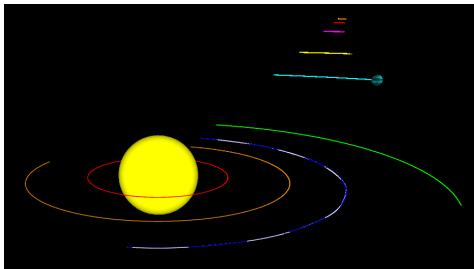
Problem Symulacji Układu Słonecznego

Symulacje Układu Słonecznego zaczynamy od wprowadzeniu początkowych parametrów. Planety są ułożone w jednej linii i zaczynają z tego samego kąta. Oczywiście proporcje dobrane są tak, by było wszystko widoczne. Dodatkowo w symulacji uwzględniono Księżyc i Pluton.

Przyjęto w programie:

- Skala odległości: 1 : 17240000
- Skala wielkości obiektów: 1190 : 1
- Stała grawitacji dla symulacji: $6.7 * 10^{-4} [\frac{Nm^2}{kg^2}]$

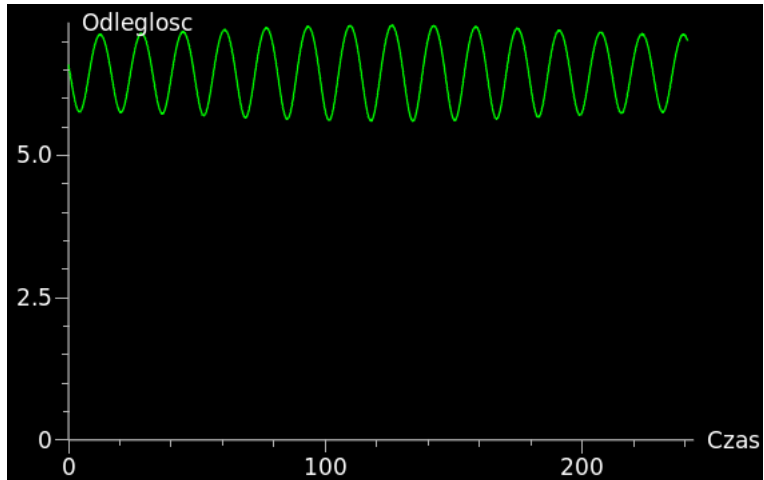
Widok wizualizacji pozycji planet.



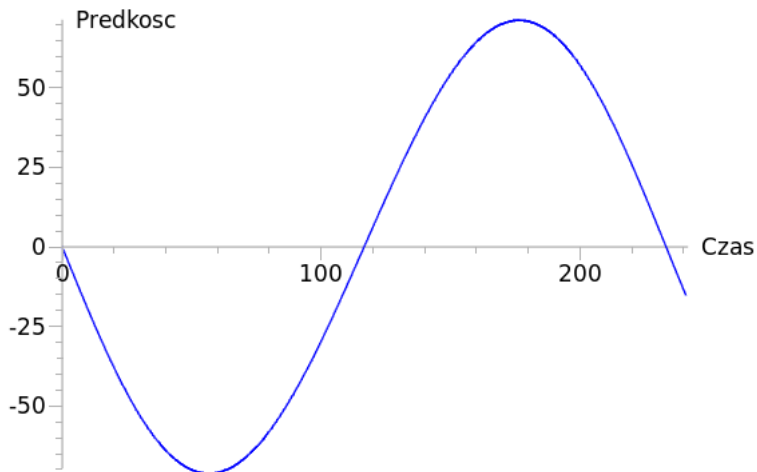
Legenda:

- Merkury
- Wenus
- Ziemia
- Księżyc
- Mars
- Jowisz
- Saturn
- Neptun
- Pluton

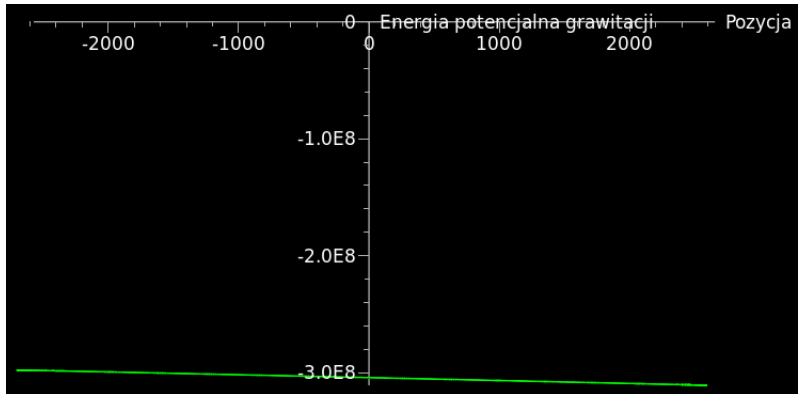
Odległość Ziemi od Księżyca



Zmiana prędkości Ziemi w czasie



Energia potencjalna grawitacji i pozycja Ziemi względem Słońca



Tworzenie planety

```
PLANET = sphere(pos=(D,0,0), radius=R, material =  
materials.TYPE, color=color.COLOR)  
PLANET.trail = curve(color=PLANET.COLOR)  
PLANET.velocity = vector (0,0,V)
```

- PLANET - planeta, nazwa zmiennej
- D - odleglosc od Słońca
- R - promień planety
- TYPE - typ materiału
- COLOR - kolor
- V - prędkość początkowa

Zmienne do rysowania na wykresach

*PLANETRATIO = gcurve (gdisplay=graphSpeedRatios,
color=color.COLOR)*

*PLANETREALRATIO = gdots
(gdisplay=graphSpeedRatios, size=2,
color=color.COLOR)*

- PLANETRATIO - krzywa (linia) prędkości planety w stosunku do Ziemi
- PLANETREALRATIO - uśredniony stosunek prędkości planety do Ziemi
- R - promień planety
- COLOR - kolor

Kod aktualizujący parametry planety

```
PLANET.trail.append(pos=PLANET.pos)
distancePLANET = mag(PLANET.pos)
UnitVectorPLANET = (PLANET.pos -
sun.pos)/distancePLANET
FgravPLANET =
(G*m_sun*m_mercury*UnitVectorPLANET)/distancePLANET**2
PLANET.velocity = PLANET.velocity +
(FgravPLANET/m_PLANET)*dt
PLANET.pos = PLANET.pos + PLANET.velocity*dt
PLANET.rotate(angle=radians(0.094), axis=(0,1,0))
if distancePLANET <= sun.radius: break
PLANET_Ratio = mag(PLANET.velocity) /
mag(earth.velocity)
```

Rysowanie wykresów i położenia Ziemi i Księżyca

```
earthposx.plot(pos=(t,earth.pos.x))  
earthvelocity.plot(pos=(t,(earth.velocity.x**2 +  
earth.velocity.z**2)**1/2))  
earthGPE.plot(pos=(earth.pos.x,earth_GPE))  
moonposx.plot(pos=(t,moon.pos.x))  
distanceMoon_earthpos.plot(pos=(t,distanceMoon_earth))
```

Aktualizacja czasu i prędkości Ziemi

```
vlist.append(earth.velocity.x)  
tlist.append(t) t = t + dt
```

Rysowanie wykresów planety zależnych od czasu t

```
PLANETRatio.plot(pos=(t,PLANET_Ratio))  
PLANETRealRatio.plot(pos=(t,PLANET_RealRatio))
```

Problemy podczas symulacji i ograniczenia:

- Program cały czas zapełnia listy, przez co po pewnym czasie zwalnia
- Skala i proporcje musiały być dobrane tak, by dało się to pokazać, jednak stosunki odległości, prędkości i położenia są zachowane
- Na Ziemię nałożona jest jej tekstura, ale jej nie widać z powodu zbyt dużej odległości

Bibliografia

- https://pl.wikipedia.org/wiki/Uk%C5%82ad_S%C5%82oneczny
- http://www.malyindywidualista.com.pl/image.php/uklad_sloneczny.jpg?width=805&image=/upload/articles/uklad_sloneczny.jpg

Dziękujemy za uwagę.