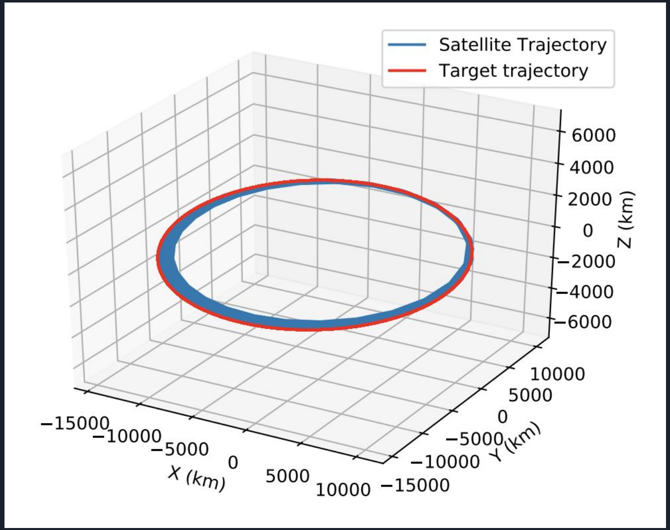
Reinforcement Learning for Spacecraft Orbits

Christopher Figueroa, Sai Shreya Jillella,
Damini Singh Thakur, Aishwarya Uyyala

Inspiration For Project

- Reinforcement learning research at Western Michigan University
- Determines maneuvers needed to change orbit of spacecraft
- Optimized for orbit of near Earth Satellites
- Set on a real-life scale
- Realistic complex physics



3D Diagram of Satellite orbiting Earth [1]

Abstract

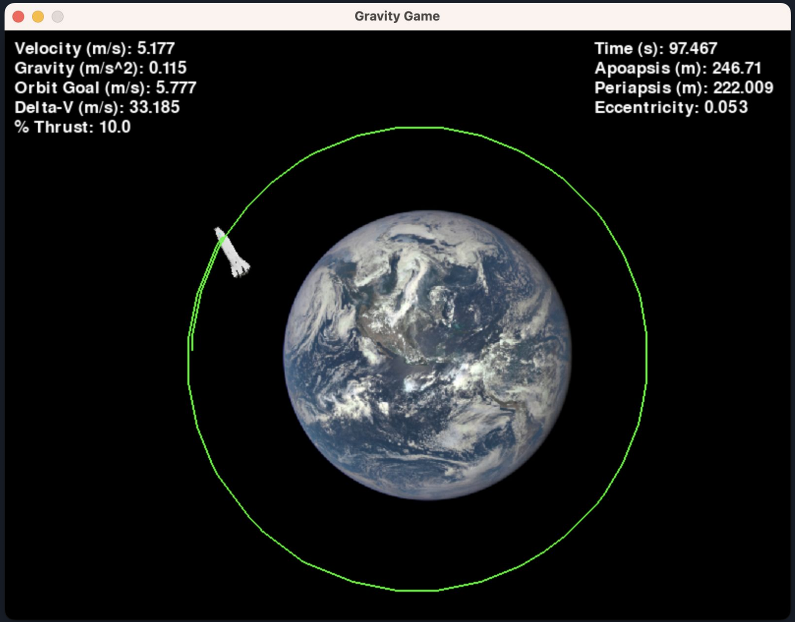
- A simplified version of the Western Michigan University program
- Also uses reinforcement learning
- Essential equations:
    - Gravity: G * (M / R^2)
    - Orbital velocity: √((M*G) / R)
    - (A) Apoapsis: highest point in orbit
    - (P) Periapsis: lowest point in orbit
    - Eccentricity: (A-P) / (A+P)
- Smaller scale, less computationally intensive

$$
\begin{aligned}
\frac{da}{dt} &= 2\sqrt{\frac{a}{\mu_E}}\left[F_R \frac{ae}{\sqrt{1-e^2}}\sin\theta + F_S \frac{a^2\sqrt{1-e^2}}{a(1-e\cos E)}\right] \\
\frac{de}{dt} &= \frac{h}{\mu_E}\sin\theta F_R + \frac{1}{\mu_E h}\left[(h^2+\mu_E r)\cos\theta + \mu_E r\right]F_S \\
\frac{di}{dt} &= \frac{r}{h}\cos(\omega+\theta)F_W \\
\frac{d\Omega}{dt} &= \frac{r}{h\sin i}\sin(\omega+\theta)F_W \\
\frac{d\omega}{dt} &= -\frac{1}{eh}\left[\frac{h^2}{\mu_E}\cos\theta F_R - \left(r+\frac{h^2}{\mu_E}\right)\sin\theta F_S\right] - \frac{r\sin(\omega+\theta)}{h\tan i}F_W \\
\frac{d\theta}{dt} &= \frac{h}{r^2} + \frac{1}{eh}\left[\frac{h^2}{\mu_E}\cos\theta F_r - \left(r+\frac{h^2}{\mu_E}\right)\sin\theta F_s\right]
\end{aligned}
$$

Examples of unneeded equations [1]

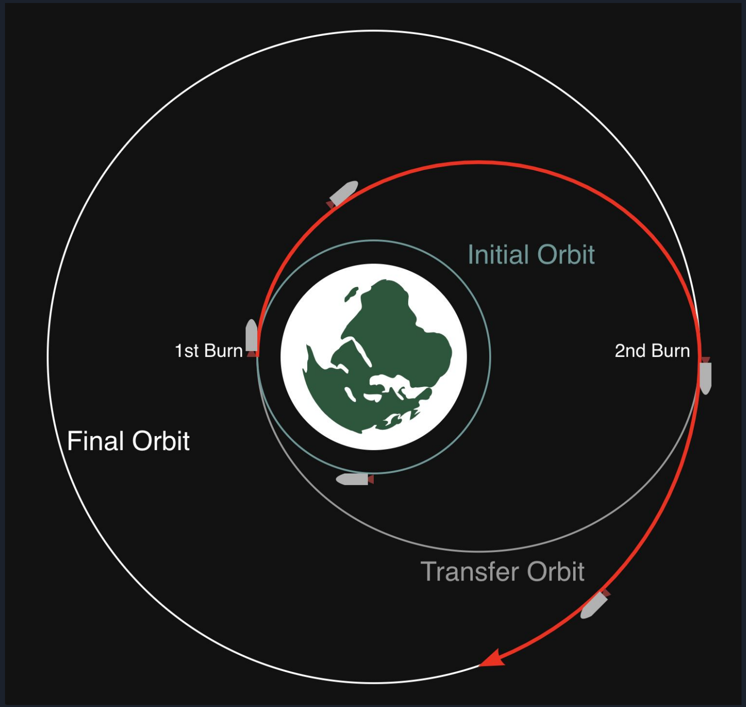
What Will Be Done

- 2D Python orbital mechanics game
- Uses simplified game-optimized physics with relevant parameters
- Scaled Earth down to 150 meters
- Real Earth is 6,371,000 meters in radius
- Gymnasium library for reinforcement learning
- Will set orbit eccentricity as measurement for loss



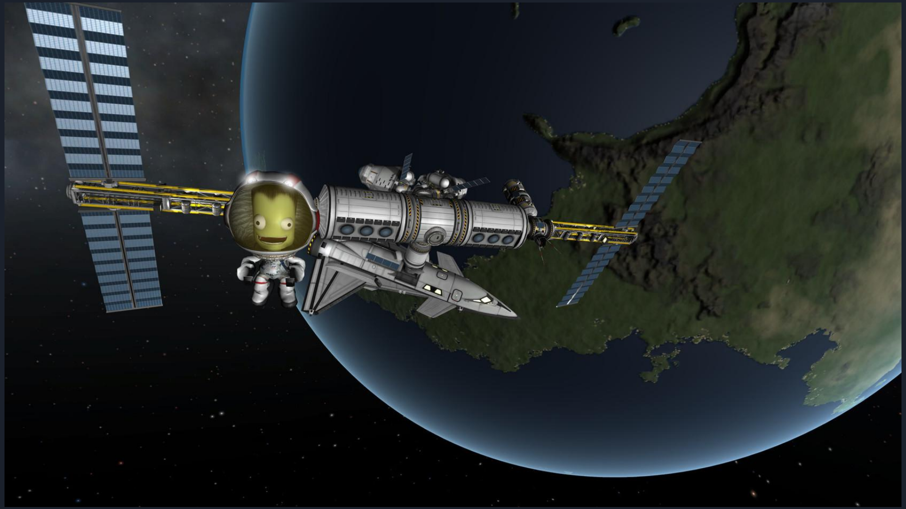
Functions in Code

- Function to rotate spacecraft
- Accelerate in <u>pointed</u> direction
- Control throttle level
- Reset game if too far or out of fuel
- All functions called <u>automatically</u> by reinforcement learning



Novelty of Project

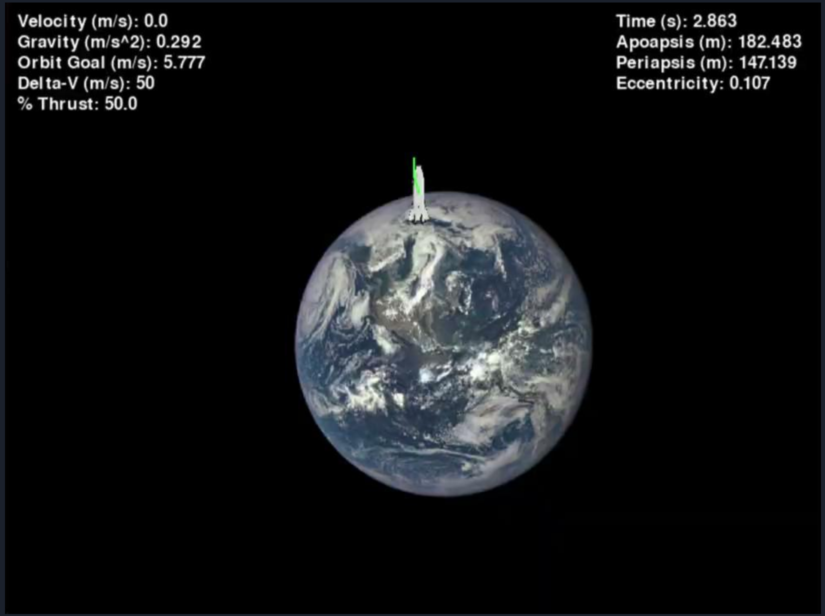
- Uses custom environment with Gymnasium library
- Optimized for use in 2D video games
- Trained to orbit planets of varying sizes and gravity
- Potential to scale up to real-life
- Could be further expanded for use in 3D games



Kerbal Space Program

Project Timeline

- 10/30/23 - Python game file created
- 11/2/23 - game physics optimized
- 11/5/23 - shapes replaced with photos
- 11/8/23 - parameters optimized
- 11/9/23 - game functions completed
- 11/11/23 - initialization of reinforcement learning
- 11/13/23 - project idea presentation
- 11/16/23 - optimization of reinforcement learning
- 11/19/23 - additional research
- 11/26/23 - debugging of program
- 12/1/23 - finalization of project
- 12/6/23 - project final presentation



Sources

- [1] Western Michigan University research: https://scholarworks.wmich.edu/cgi/viewcontent.cgi?article=4537&context=dissertations
- [2] Orbital parameters: https://courses.lumenlearning.com/suny-osuniversityphysics/chapter/13-5-keplers-laws-of-planetary-motion/
- [3] Gravity and momentum: https://oer.pressbooks.pub/lynnanegeorge/chapter/chapter-1/
- [4] Gymnasium AI: https://gymnasium.farama.org/