

Orbital Mechanics Simulator

Lucius Kwok (VC1B)

<lk@feltp.com>

Supervisor: Brian Papa

<bpapa@icloud.com>

Abstract

Orbital mechanics are an often misunderstood aspect of spaceflight, but one which can be explained through hands-on piloting of a simulated spacecraft which follow a few basic mathematical equations. In this project, I will focus on teaching how a few basic laws of Newtonian physics can result in many types of orbits around celestial bodies, and how to figure out the most efficient way to get from one planet or moon to another. Along the way, I may touch upon the Oberth effect, Hohmann transfers, gravity assists, patched conics, the rocket equation, and other spaceflight-related concepts. By turning it into a game to find the most efficient solution, learning becomes fun and accessible to everyone.

Tools

- Software
 - Unity 6 engine
 - Microsoft Visual Studio Community 2022
 - Autodesk Maya 2025
 - Adobe Photoshop 26
 - Panic Nova (text editor)
 - GitHub suite
 - Google suite
- Hardware
 - Development:
 - Windows PC with Intel Core i7-11700 & NVIDIA GTX 1080 Ti
 - Apple MacBook Air (M3, 2024)
 - Target Platforms: Windows, macOS, WebGL

Diagram 1: Screen UX Flow

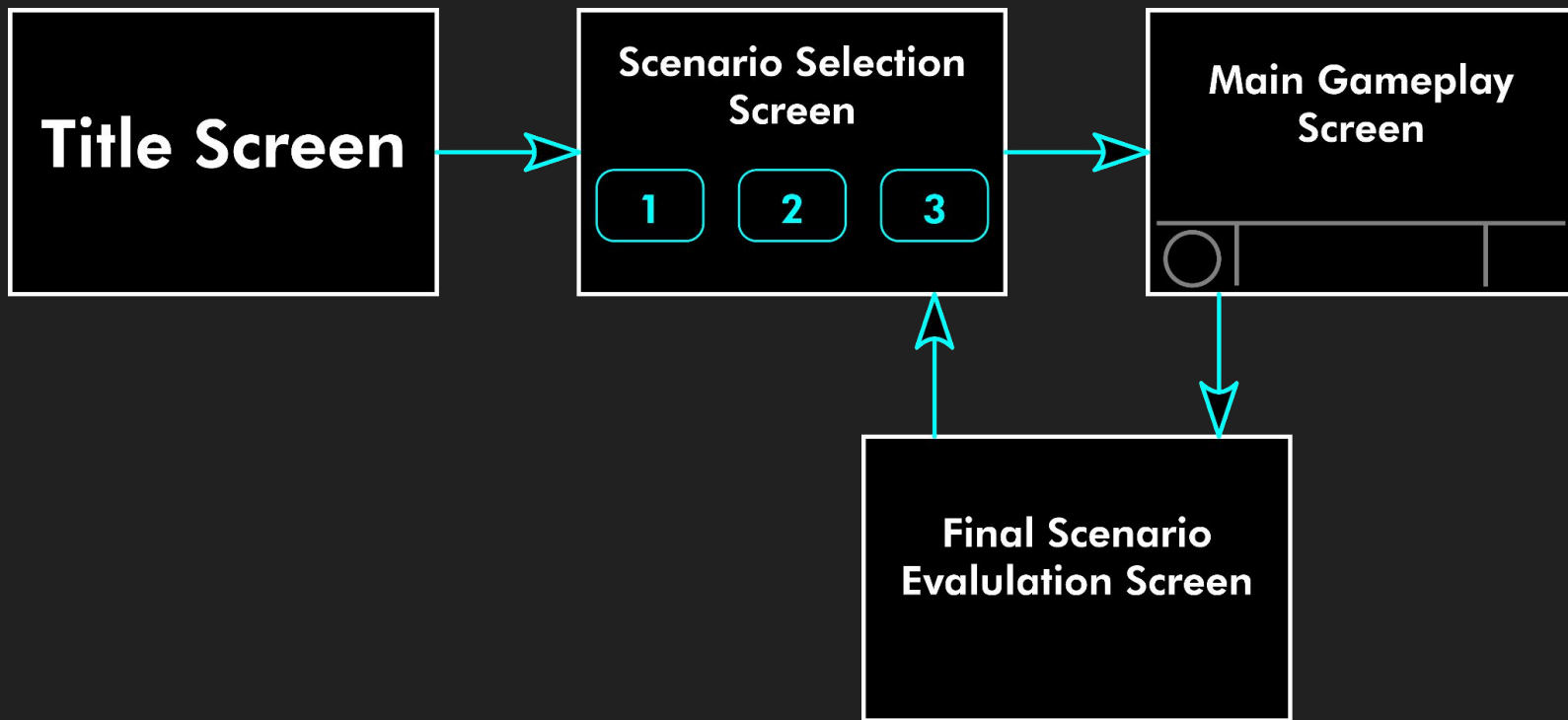
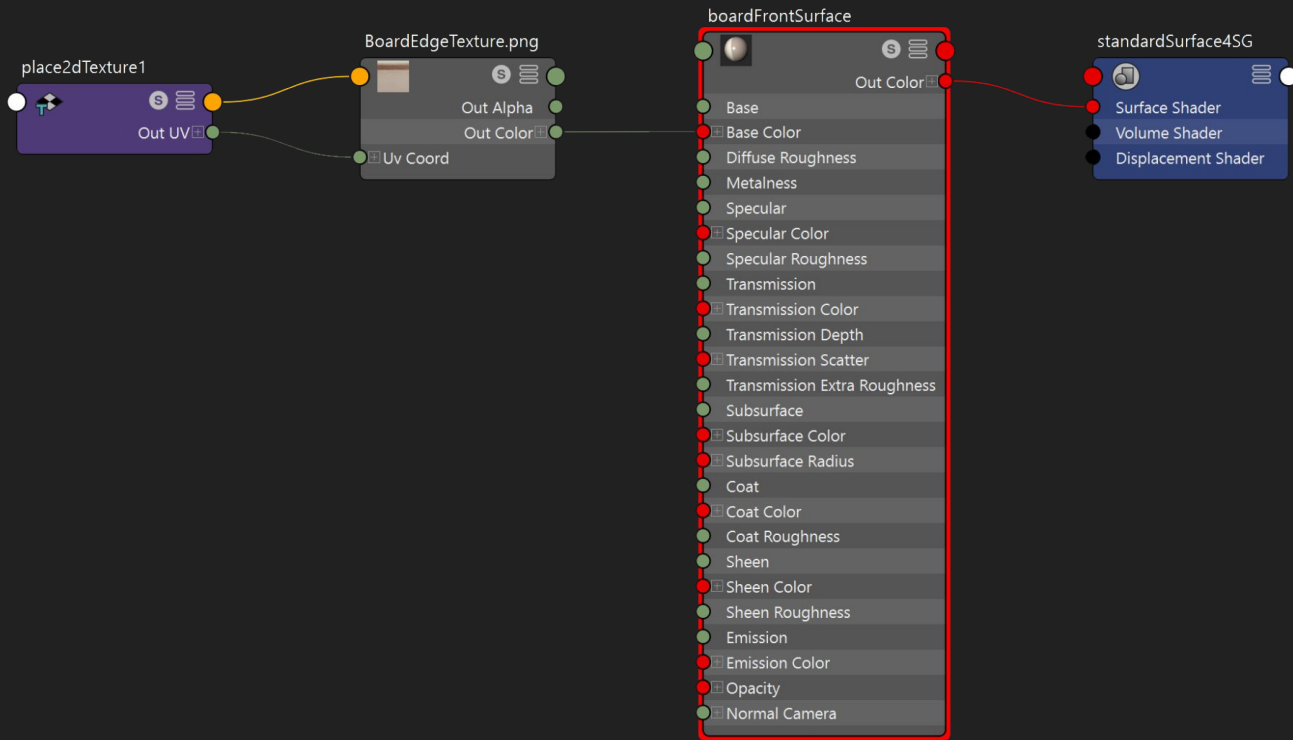


Diagram 2: Example Shader Network



Simple texture mapping shader for 3D rendering

Diagram 3: Example of Patched Conics

Transfer orbit with entry and exit from different spheres of influence.

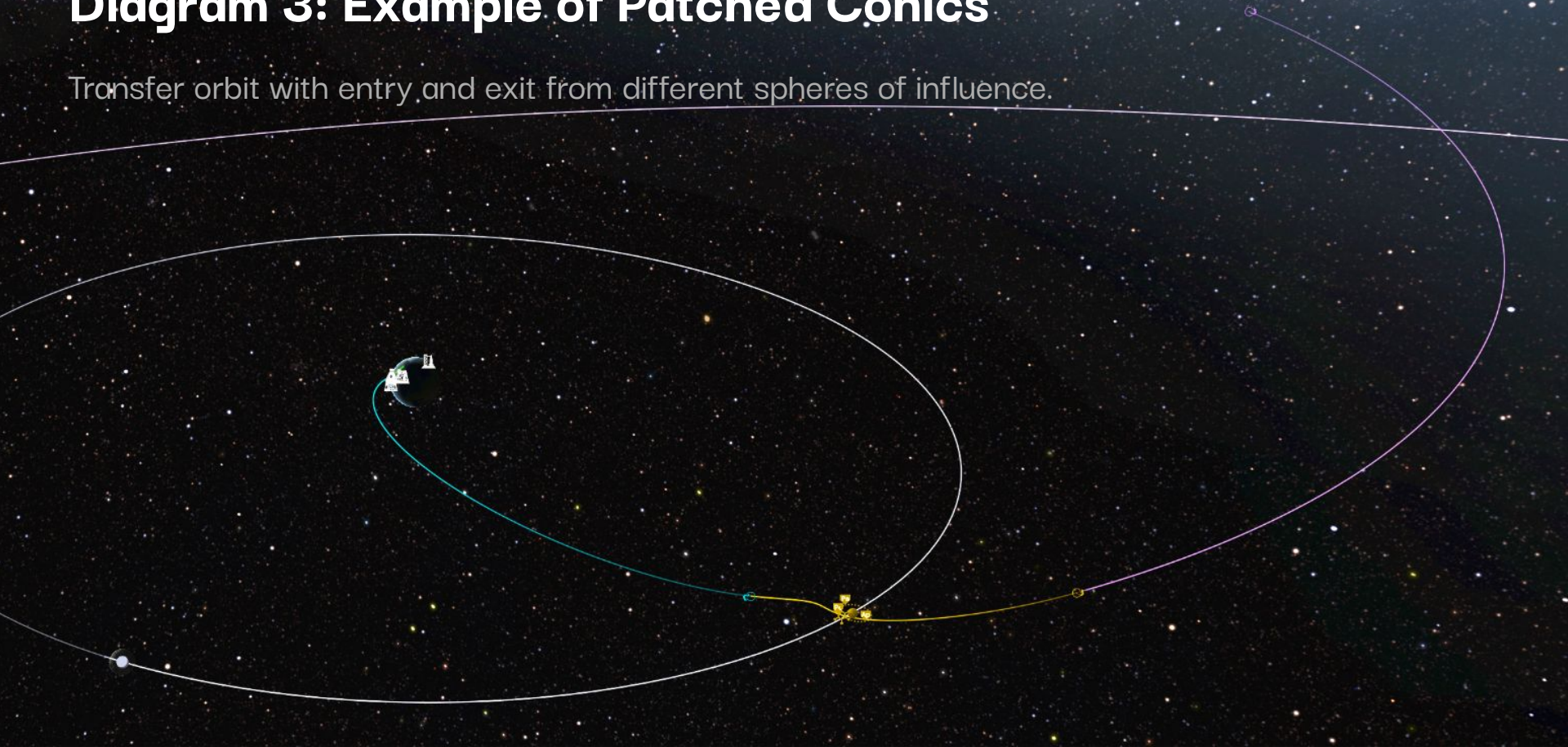
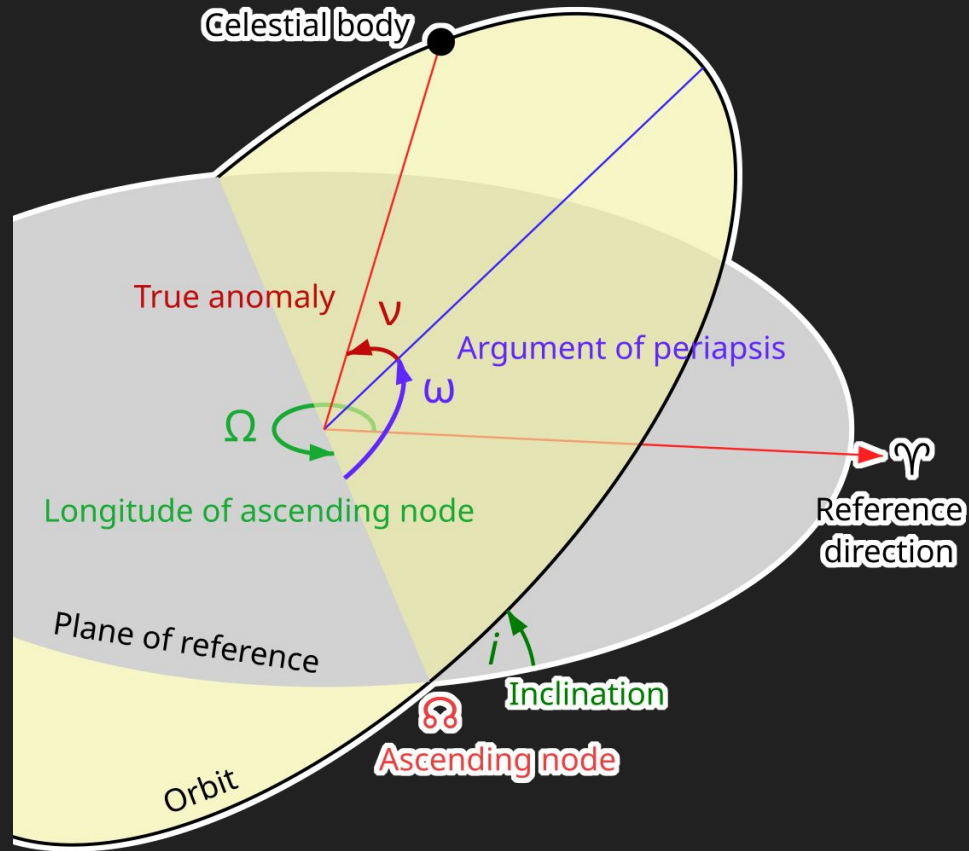


Diagram 4: Keplerian Orbital Elements

Showing an inclined elliptical orbit



Tentative Schedule

Week 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Project planning & research. Set up project repo & issue tracker.															
		Identify users, use cases, and proposed solutions. Sketch UI/UX.													
				Develop prototype/demo. User testing, presentation.					Further testing			Final testing			
					Refine and build out UI and features based on user and presentation feedback.										
								Triage features and issues to decide on MVP							
											Feature lock & focus on fixing issues.				
													Prepare final presentation & submission		

GitHub Projects

The screenshot shows a GitHub Projects page for the 'Lucius Kwok CISC 4900 Project'. The page is in dark mode. At the top, the browser address bar shows the URL 'github.com/users/luciuskwok/projects/2/views/1'. The page header includes the user's name 'luciuskwok', the project name 'Lucius Kwok CISC 4900 Project', and a search bar. Below the header, there's a section for 'View 1' with a '+ New view' button. A filter bar allows filtering by keyword or by field, with 'Discard' and 'Save' buttons. The main content is a table with columns: Title, Assignees, and Status. The table lists 10 items, each with a status (Done or In Progress). The first item is 'Delete old repo and free up space on LFS #1' with status 'Done'. The remaining 9 items are 'In Progress'. At the bottom, there's a '+ You can use Control + Space to add an item' button.

	Title	Assignees	Status
1	✓ Delete old repo and free up space on LFS #1	luciuskwok	Done
2	➡ Add Mercury, Venus, and Mars to scene #10	luciuskwok	In Progress
3	➡ Research how to deal with solar system distances in Unity #2	luciuskwok	In Progress
4	➡ Update slide deck to include link and screenshot to GitHub repo #8	luciuskwok	In Progress
5	➡ Update slide deck to include link and screenshot to GitHub Projects #9	luciuskwok	In Progress
6	➡ Learn how LOD (level of detail) works #3	luciuskwok	In Progress
7	➡ Add controls for moving the camera around in space #4	luciuskwok	
8	➡ Add controls for rotating the spaceship #5	luciuskwok	
9	➡ Add spaceship to scene #6	luciuskwok	
10	➡ Add basic UI to gameplay screen #7	luciuskwok	

+ You can use `Control + Space` to add an item

<https://github.com/users/luciuskwok/projects/2>

GitHub Repository

[https://github.com/luciuskwok/
CISC-4900-Unity](https://github.com/luciuskwok/CISC-4900-Unity)

The screenshot shows the GitHub repository page for **CISC-4900-Unity** by user **luciuskwok**. The repository is public and has 1 commit on the **main** branch. The file list includes **Diagrams**, **OMS**, **.gitattributes**, **.gitignore**, **LICENSE**, and **README.md**, all committed 48 minutes ago. The **README** file is selected, showing the title **Orbital Mechanics Simulator**. The description states it's a spacecraft simulator using Newtonian physics and patched conics. The **Abstract** section explains the project's focus on Newtonian physics and orbital mechanics. The **Tools** section lists **Software** including Unity 6 engine, Microsoft Visual Studio Community 2022, and Autodesk Maya 2025. The right sidebar shows repository statistics: 0 stars, 0 forks, and 1 watching. It also includes sections for **About** (no description), **Releases** (no releases published), **Packages** (no packages published), **Languages** (Mathematica 52.3%, C# 34.8%, ShaderLab 12.9%), and **Suggested workflows** for .NET Desktop and .NET.

github.com/luciuskwok/CISC-4900-Unity

luciuskwok / CISC-4900-Unity

Code Issues Pull requests Discussions Actions Projects Wiki Security Insights Settings

CISC-4900-Unity Public

main 1 Branch 0 Tags

Go to file Add file Code

luciuskwok Initial commit 6497143 · 48 minutes ago 1 Commit

Diagrams	Initial commit	48 minutes ago
OMS	Initial commit	48 minutes ago
.gitattributes	Initial commit	48 minutes ago
.gitignore	Initial commit	48 minutes ago
LICENSE	Initial commit	48 minutes ago
README.md	Initial commit	48 minutes ago

README BSD-3-Clause license

Orbital Mechanics Simulator

Orbital Mechanics Simulator is the title of my project for the CISC 4900 Independent Projects course. It is a spacecraft simulator which models realistic orbits around celestial bodies using Newtonian physics and patched conics.

Abstract

Orbital mechanics are an often misunderstood aspect of spaceflight, but one which can be explained through hands-on piloting of a simulated spacecraft which follow a few basic mathematical equations. In this project, I will focus on teaching how a few basic laws of Newtonian physics can result in many types of orbits around celestial bodies, and how to figure out the most efficient way to get from one planet or moon to another. Along the way, I may touch upon the Oberth effect, Hohmann transfers, gravity assists, patched conics, the rocket equation, and other spaceflight-related concepts. By turning it into a game to find the most efficient solution, learning becomes fun and accessible to everyone.

Tools

- Software
 - Unity 6 engine
 - Microsoft Visual Studio Community 2022
 - Autodesk Maya 2025

About

No description, website, or topics provided.

Readme

BSD-3-Clause license

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Mathematica 52.3% C# 34.8% ShaderLab 12.9%

Suggested workflows

Based on your tech stack

.NET Desktop Configure

Build, test, sign and publish a desktop application built on .NET.

.NET Configure

Build and test a .NET or ASP.NET Core project.

More workflows

Dismiss suggestions

Use Cases

1. Astrophysics course to illustrate different kinds of orbits and how velocity varies with the distance.
2. Casual user who wants to play with spaceships in orbit and solve challenging scenarios such as finding the most efficient transfer from Earth orbit to Moon orbit.
3. Video maker (for YouTube, for example) who wants to generate animations to show the orbits of asteroids, satellites, and celestial bodies in their videos.

Data Sources

NASA: Gravitational constants; distances and diameters of the sun, planets, and moons.

University of Texas: Orbital elements.

Wikipedia: Information on celestial bodies (planets, moons, stars).