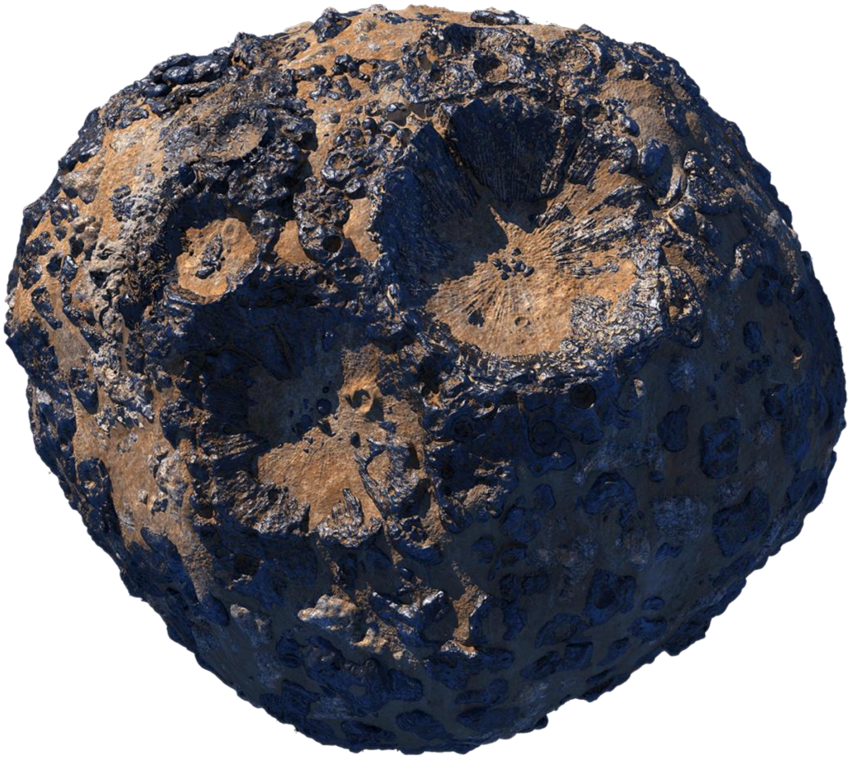
Psyche Sampling Lander Simulator

Version Description Document

Version 2.4



Completed by Teammates of: **Control Alt Elite**

Brigham Young University -Idaho

Arizona State University

NASA

12.14.2024

Overview

We have developed a Psyche Sampling Lander Simulator system that allows users to land a spacecraft on the Asteroids surface. The goal is for the system to accurately simulate the Psyche environment, including gravity, orbital mechanics, spaceflight, and asteroid terrain, providing a realistic landing experience. Novice users can control the spacecraft thrusters in real time. The project includes visual feedback with detailed (currently 2D) graphics and real-time telemetry data to monitor crucial parameters like altitude, speed, and fuel. This system may serve as an educational tool and a research aid for space agencies and enthusiasts interested in space exploration.

Goals

1. Make an asteroid sampling lander simulation that is easy and engaging for almost anyone to use.
2. The simulator environment should be semi-realistic, reflecting the actual environment of Psyche, such as gravity, rotation, diameter, etc.
3. The lander simulation should display the changing environment of a lander as it descends toward the asteroid surface, such as increasing orbital speed to keep the spacecraft in a geosynchronous orbit directly over the chosen landing location, increasing fall velocity due to Psyche's gravity, changing that fall velocity due to thruster effects, tracking the simulation times, etc.
4. A visual representing the descent toward the asteroid surface should give the user the impression of a lander approaching the asteroid while the changing parameters of the environment should be updating and the constant parameters of the displayed for reference.
5. The users should have control over the simulation environment, so they can start, pause, resume, or exit the simulation at will.

6. The simulator should present elements of future features such as Configuration Menu user input, multi-skill levels optimizations, Load Mission, and Save Mission feature elements, which would not be functional for the lowest skill level.
7. The simulator will be functional for the Novice skill level, while presenting non-functional features of Intermediate or Expert skill levels such as a drop-down selector and the features mentioned in goal six above.
8. Novice level operation will consist of upper and lower cursor keys introducing the effects of thrust toward or away from the asteroid.

Contributors

From Brigham Young University – Idaho:

Front End:

- Jerry Lane
- Cayeigh Leishman
- Brycen Williams

Back End:

- Levi Johnson
- Ryan Funk
- Gerin Wilde

Milestones

I. Test Cases

As this was our first professional Project and having only 6 weeks to work on coding the project, our test cases were necessarily restricted to unit tests performed during development. For instance, if the page displayed correctly, we considered the test to be conducted satisfactorily. If a button or cursor key caused the display to change in the way expected, we considered the test completed and functionally passed. Beyond this, we simply did not have time to conduct further formal testing.

II. Installation

Installation instructions can be found in the GitHub repository under “Documents” and “Onboarding Notes.” This document contains instructions on installing the materials needed for the project as well as the IDE, Node.js, React, ASP.NET Core, and anything else needed for the project installation, application use, and development environment. Once you find the PDF file, all the installation instructions can be located between pages 4-23 depending on your needs.

III. Troubleshooting Errors:

Common troubleshooting Errors we ran into as a team can be found in the Onboarding Notes on Page 32 and forward.

IV. Future Features

One thing we leave open for the next team is the ability for the Lander (the blue thing on the simulation) to either travel around the asteroid to “collect Samples” (or if it was just going to land in one spot and drill down, give it the ability to drill the asteroids surface and show the results or fun facts once it reaches different layers of the asteroid.”

More future features include implementation of the Load Mission button, Save Mission button, and Configuration Menu user inputs, enabling the

Intermediate and Expert skill levels to have greater control over what can be set or altered, and making the simulation view more interesting by creating a custom lander and using a spinning star field background (to illustrate the rotation period of Psyche and help the user understand that little in a mission such as this is static).

More information can be found in the Onboarding Notes found in the GitHub repository under "Documents" and "Onboarding Notes."

V. Completed Stories

EPIC - Project Management: Finish Discovery Elicitation Techniques.
Create introspection document
Brainstorming
EPIC - Project Management: Finish Empathy Elicitation Techniques.
Ethnography - Profiles
Ethnography - Scenarios
Domain Analysis: Card Sorting
EPIC - Project Management: Finish Analysis Elicitation Techniques.
Domain: Viewpoint: Top Down: Task Analysis
Stakeholder: Requirement Workshop
EPIC - Project Management: Create SRS/SDD Documents.
Write SRS Document
Create SRS Section 1
Create SRS Section 2
Create SRS Section 3
Create SRS Section 4
Create SRS Section 5
Update SRS to v2.2
EPIC - Product Management: Start User Documentation.
EPIC - Front-End: What OS will the application run on?
Create Operational Frontend White Paper Template
Research and compare Operating Systems
EPIC - Back-End/Server: Where will we be hosting server that front-end logs into?
Create Backend Operational White Paper Template
Research and compare Frameworks
EPIC -Database: Where will we be hosting the DB tables/documents?
Create Database Operational White Paper Template
Fill the White Paper Template with options and information

Research and compare Cloud Systems
EPIC - Project Management: Create three white papers
Complete front-end white paper
Complete back-end white paper
Complete database white paper
EPIC - Product Management Start Installation Document
Create installation outline
EPIC - Write SDD Document
Create SDD Sections 1-2
Create SDD Section 3
Create SDD Sections 4-5
Create SDD User Interface Design
Create SDD Main Menu Design
Cookie Management Design for SDD
Error Handling design for SDD
Continued error design research
Wrote error pseudocode and general flowchart
Simulation parameter design for SDD
Simulation controls design for SDD
Visualization windows design for SDD
Transfer 2024FallCSE397_Team3 to iridium_22f_m-type_sim-byu-i
EPIC - Product Management: Start Development Guide
EPIC - Front-End: Development Software
Determine IDE
Determine Front-End Libraries
Determine Interface with Back-End
Update SRS
Update SDD
EPIC - Create Simulator Front-end
EPIC - Create home page
Create Welcome component
Create Logo component
Create Activate Simulator button component
Create More Information button component
Create home page component
Create Psyche mission resource page
Create Information page component
Create Parameter Panel component
Create Constant Parameter panel for lower half of Parameter Panel

Create Variable Parameter panel for upper half of Parameter Panel
Create Configuration Menu component
Create Main Menu component
Build keyboard in method for user input
Import shared variables to main menu
Build error handling method for failed load or save ops
Build error handling method for failed landing
Build graphics generation using main simulation canvas and Three.js engine
Psyche Mission Resource page when user clicks on the home page button
Link Home Page to About Page
Link About Page to Home Page
Link Home page to More Info Page
Link Home Page to Main Menu Page
Create Main Simulation window with Placement For components
Create About page with NASA verbiage, authors
Create JSON Request code to send and retrieve data
Build simulation controls for user input
Create simulation controls component
Create a variable tracking system to access variables across components
Import the shared variables through app
User Manual
Revise SRS
Revise SDD
Tie parameter to orbital speed
Update Shared Context
Create Start Time parameter
EPIC - Back-End: Development Software
Initialize ASP.NET Core API with React
Physics logic
Build Project using Visual Studio Debugger
EPIC - Create Simulator Back-end
Design and implement model for parameters
Implement I/O calls for frontend API
Write functions for physics calculations
Write business logic to handle data exchange and requests
Use built-in Exception Handlers to manage possible errors
Use built-in Exception Handlers to manage possible errors
Build front-end API handler code

VI. Scenarios and Use Cases

You can find all the Scenarios we have within this file:

(You'll need to have been given access to the GitHub repository to see these.)

https://github.com/MissionToPsyche-Iridium/iridium_22f_m-type_sim-byu-i/tree/main/Documents/Elicitation/Scenarios

Also, use cases can be seen in the Software Requirement Specifications document, version 2.4, which can be found at:

https://github.com/MissionToPsyche-Iridium/iridium_22f_m-type_sim-byu-i/tree/main/Documents/SRS/