Computer Science Department

Capstone Project Proposal

**Kerbal Space Program Engine Optimizer**

12/13/2020

Matt Jones

Course Number and Course Title: GPH CSC 520

Project Advisor

Dr. Bo Hatfield

**Contents**

Student Objectives x

Problem Specification x

Solution Processes and/or Design x

Requirement Collection and Analysis x

Identifying Needs of Customers

Identifying Target Specifications

Use Case Diagram/Sequence Diagram/Activity Diagram

Generating Design Concepts x

Descriptions and Diagrams

Detailed Design x

Descriptions and Diagrams

Preliminary ER-Diagram (SQL) x

Or show visual design of hierarchical structures (NoSQL)

Benchmark Specification x

Tool List x

Time Schedule

Grading Scheme for CSC521 x

List of Deliverables x

References (any useful support materials) x

**Student Objectives**

* Increase experience using Java
* Increase experience creating GUIs
* Learn how to use databases such as Amazon Webservices using \_\_\_\_\_ (MongoDB/DynamoDB, will choose before final draft)
* Experience with the complete life cycle of a project, from initial idea to problem specification, then creation and verification.
* Increase user’s(and my) enjoyment of KSP by streamlining rocket design

**Problem Specification**

Kerbal Engine Optimizer is a program to support the game Kerbal Space Program(KSP). The point of the game is very similar to that of NASA(just a much simpler) you make rockets from in game parts and explore the solar system. The apps main function is to assist the user in planning and building of a rocket.

The largest issue when building a rocket is knowing what you need in terms of fuel and engine thrust to get from point a to point b. Too large of an engine means more fuel and more parts to keep the rocket size proportioned, (like stacking dimes and quarters together, its too unstable without supports) Too small of an engine means you cant generate the thrust required to move your rocket. Then comes fuel, too much fuel means you need a bigger engine but not enough means you fall short and get stuck without fuel in the middle of space, which can be just a little problematic. So, you need to find a happy middle right engine for the right job and just enough fuel to get there and back again.

In order to pick the best engine, there are many calculations that need to be done, (even for a simple game that ignores many things that real rockets would need to take in to account.) Say you want to put a satellite in orbit, you build your satellite with all the gadgets and gizmos you want and the mass of it will be your payload mass. In order to put that satellite in orbit at 100k meters around Kerbal(the main planet) it will take about 3400 DV (plus 10-20% to be safe) and a minimum of 1 TWR when on the launch pad to lift off. Now you can just pick an engine add fuel and hope for the best or do the math yourself. But in order to that you need to figure how fuel efficient it is in the atmosphere at different atmosphere pressures, then the equation to find out how much fuel it will take to lift the payload mass + engine mass + fuel mass while still giving you your required DV and surface TWR. If you’re lucky your first pick will work, if not you have to select another engine and do it all over again, if you are super unlucky none of the engines will work and now you need to do all of those calculation again this time for multiple engines. This where the app comes in, you find your payload mass, your desired DV and TWR range, where this maneuver takes place, (each planet has its own gravity and atmospheric pressure to take into account) and maximum number of engines you would be willing to use. The app will do the thousands tedious calculations, graph the results and allow you to select the engine that gives you exactly what fits your needs. You will be able to select which engines you want tested in case you just want to compare a few and will also be able to create custom engines if you downloaded a mod that had more engines or if you changed an engine in the games files.

**Requirement Collection and Analysis**

Stakeholder Identification

* Standard user - use the program
* Admin – monitor backend, updates, and database

Requirement Lists for Requirement Specifications

* User can input desired trip values (payload, max/min TWR & DV, gravity, atmosphere, and max number of engines, and select specific engine/s to test for)
  + In the test section, could have a form output to verify the inputs have been saved correctly
* Create and save custom engines
  + After creation of custom engine to database, verify on database values are saved
* Have results graphed and selectable exact TWR & DV points on the graph
  + Check selected results against expected results that the game would have for similar requirements
* Login and general user functionalities
  + Create test user and test all functionalities
* Custom engines saved on database for each user
  + Create test users and test engines and check that created engines stay with its created user on the database
* User info saved on database
  + Create test user and check database for correctness

Use cases

