|  |  |  |
| --- | --- | --- |
| CSC 436  Due 9/14/17 | **Project Proposal** |  |

Project Title: OpenBurn

* Team #12: Team Rocket
* Team Members:

Abhishek Rane, Daniel Tranfaglia, Isaac Plunkett, Andrew Tarr, Vicente Figueroa

1. Executive Summary

* OpenBurn is an open source, multi platform solid rocket motor internal ballistics simulator that takes a fuel model, and grain geometry and produces a pressure vs time plot.

1. Customer Value

* University of Arizona IREC Team, rocket enthusiasts, Aerospace Engineers.

Unmet Customer Needs

* Better simulation results than those that are currently available that are also open-source, and accounts for erosive burning. Improve upon graphical user interface.
* Helps design better motors and save on expenses through simulation over physical testing.
* User Stories

1. Problem Definition

* Multi-platform, open-source software, introduces multiple unit availability, accounts for erosive burning, improved Graphical User Interface.
* Java implementation, open-source equations from NASA.

Proposed Benefit

* This proposed software addresses erosive burning during simulations.
* Users will be able to use simulation results to help build motors more efficiently.

Measures of Success

* Dr. Mark Langhenry – Current Raytheon propulsion, Former Lockheed Martin Propulsion Engineer, Former AIAA Solids Chair.
* We will test results against actual rocket motor data.
* We will produce a program with an easy to use interface that outputs accurate Thrust vs. Time data for a range of rocket motor designs.

1. Technology

Proposed System

Our team will follow the common OOP design pattern: MVC (Model-View-Controller).

* ***Model***:

RocketMath – Contains static methods which contain the equations, methods take Grain and Motor objects.

Motor – Contains the dimensions of the motor and the Grain objects.

Grain – Contains data for a motor grain, including shape, type, length, inner/outer diameter, and number of burning faces.

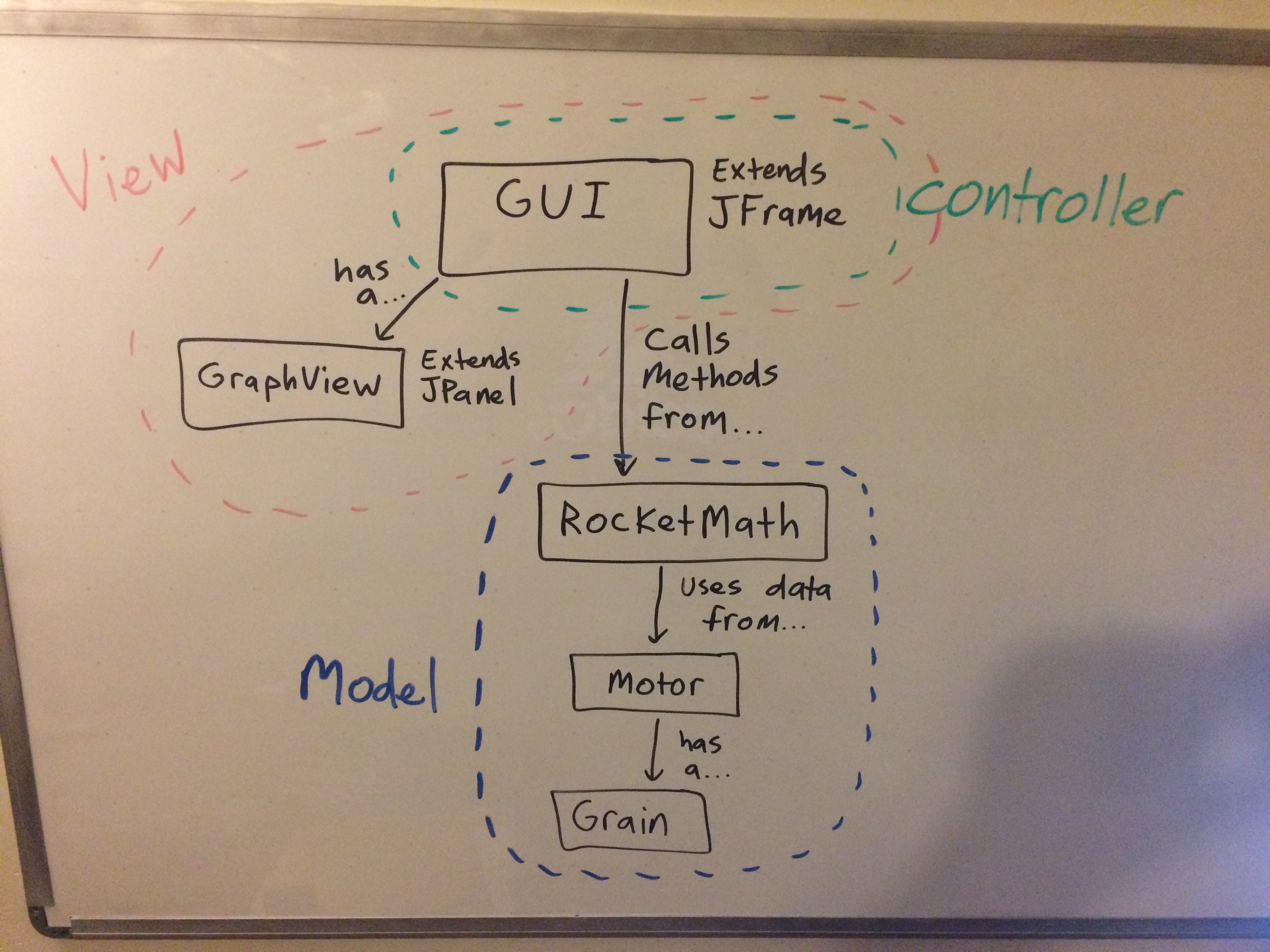
* ***View***:

GUI – Displays the program window, and text fields for the simulation inputs, and the output graph and data.

GraphView – Takes a list of points and graphs them. Possibly implemented with JasperReports.

* ***Controller***:

GUI – when the calculate button is pressed, gets data from input text fields and constructs motor and grain objects.  Passes them to static functions from the RocketMath class.  Passes the output into the GraphView class.



* What is a minimal system that would have some value to the customer?
* What are some additional features on your wish list?
* How will you test your system?

Tools and External Technology

* Eclipse, JUnit, JasperReports, CSV/Excel files, RSE files
* GitHub, JRE

1. Team

Backgrounds

* **Abhishek:** Aerospace engineering and computer science experience. I was a project lead for the intercollegiate rocket team and I have experience designing and building solid rocket motors. I was responsible for the simulations related to the motor. I also worked at an autonomous drone technology startup in Tucson. I have used Java, C, MATLAB and have participated in several hackathons.
* **Vicente:** I have majored in computer science my whole time in college and have some experience with GUIs and simulations. I am experienced with C, Java, and Javascript. I currently work at an internship with HealthTrio so I have a good amount of experience with working as a team and experience with Agile, Git and JIRA.
* **Isaac:** Has participated in multiple game jams and hackathons. Experience with C, Java, C#, and Unity3D.
* **Daniel:** Experience in C, Python, and Java, along with 10 months of internship experience with Metropia Inc. (6 months) and currently at HealthTrio LLC (4 months). Has worked with Agile teams that utilize software such as Git and JIRA. An aspiring Full-Stack developer who can design systems and databases. Has also participated in two hackathons, and has some experience with drones.
* **Andrew:** Industry experience with testing and releasing an application, experience with Java, C, HTML, CSS, and JavaScript.

Skills

Has anyone on the team built something like this before?

Are the tools known or new to the team?

* Vicente: I do not have any prior knowledge related to the project other than how to build an interface. I hope to learn many skills while working on this.
* Daniel: I have never worked on projects that involve aerospace concepts, however I constantly work on projects that involve OOP and Test-Driven Development, as well as GUI’s. I have also worked on Agile teams and have gained valuable knowledge that I believe will benefit this team. I am planning to do research on JasperReports as I believe they will help us with plotting data.

Roles

* **Product Manager: Abhishek** – Knowledgeable of the product requirements and constraints.
* **Team Coordinator: Vicente** – Tracks progress for the team. Somewhat of a Scrum Master.
* **Front End Development: Isaac** – Main developer on the GUI and view side.
* **Back End Development: Andrew** – Main developer on the logic/math side.
* **QA Engineer: Daniel** – Implements Test-Driven Development and ensures quality over all systems. Ensures good OOP practices are followed.
* **All members:** Development of certain components, and monitoring the work and progress of the entire team.

1. Risks

Constraints

* Rocket technology falls under Export Controls.  We must ensure all equations originate from unclassified sources.

Resources

* We have access to motor firings from last year. However erosive burning was not a factor for these burning.
* Access to motor firing data where erosive burning was a factor.
* We also to previous data from physicals to compare results.

1. Project Management

Process

* Our team will use Agile development. We will use Trello and GitHub to drive our Agile process.

Schedule

* **Viable -** CMD program that receives inputs of fuel shape, type and length and outputs a CSV of Pressure vs. Time.  Supports Imperial units.  Calculation assume steady state.
* **Feature -** GUI that produces a visual graph of Pressure vs Time.  Can save data to a CSV.  Imperial and metric units.
* **Beta -**GUI that produces a visual graph of Thrust vs Time.  Can save data to a RSE file.  Calculations account for Erosive burning.  Can display multiple plots.
* **Final -** Calculations support transient state calculations, can display side-by-side graphs for comparison.

Team Meetings

* Our team will meet 2-3 times a week in person. Also we will use the GroupMe app to stay in contact, along with FaceTime and google hangouts to stay on the same page.
* Additionally we will use Trello to keep track of tasks and their completion.
* We will discuss the status of our individual parts.  How close we are to implementing the next iteration, and if we need help to implement our individual part.
* Have all team members give updates on the progress of the current tasks. Also plan what tasks need to be done next.