Project Proposal

Jeremiah M. Via

28 January 2011

1 Functionality

The game will be a multi-player game with simple mouse-based controls. Players move about the environment by expelling a proportion of their own mass in order to gain momentum. Gameplay consists of absorbing smaller objects and the avoidance of absorption from larger objects. These two facts together force players to create a balance between their speed and their mass.

Levels will vary in main goals and level design. Some may be situated more in deep space while others are situated around a star system with all the effects of an orbit.

2 GUI design

For the GUI design, we are going to focus primarily on using the swing graphics. If there is time at the end, we may extract an interface from our graphics code and allow either OpenGL (using JOGL or LWJGL) or swing to plug in to it.

We plan on making the game full-screen since that creates a more immersive experience. The escape key will be used to bring up a menu so that players can access various functions. We may make the GUI have the ability to exists in windowed states as well. That would be a nice feature for debugging because we could get back to our code much more easily.

To prevent those with much larger screens from having an unfair advantage, we are going to scale the graphics of the game to the monitor. In this respect, a person playing on a netbook won't be at too much of a disadvantage against someone playing with a much lager display.

Our initial prototypes will be made with simple Java shapes. As time goes on, we will use sprites to create nice looking graphics. We will use techniques like mixing a base picture with various rotating, semi-transparent pictures on top to create effects within the player's character.

3 Release Plan

We our going to adopt the eXtreme Programming methodology; just tailored slightly to fit our needs and constraints. We plan on having weekly iterations of our code. At the beginning of each week, we will have a few features that we really want to add. That entire week will be spent on coding them up and making them robust and well documented. At the beginning of the next week, we will review what we did good and bad over the last week and see how we can improve our processes.

Every week we will aim to have a stable, minimal working product. We will tag what we have as a release so it will sit as reference in the repository whenever we need it.

4 Software Engineering and Testing

As mentioned earlier, we will be using the eXtreme Programming methodology for this project. We all feel it will be a good fit for this project and our team. We hope to code at a consistent pace so at the end we are not forced to work feverishly just to meet the deadline.

We plan on doing a lot of pair programming for the project. This should help everyone have an understanding of the code base. Currently, we meet up three times a week for our pair programming sessions. That way, everyone spends time coding with everyone else.

We're using Apache maven to build our product. It frees from some of the more manual aspects of using Ant and has a very simplified release cycle. It will also handle the dependency resolutions for us. This means no one has to worry about installing jUnit because Maven takes care of it.

The other benefit is that Maven will generate a website with information regarding our project. One of the more useful pages is the 'Checkstyle' page. It ensure that our code is correctly formatted to Sun's Java standards.

We plan on creating a lot of unit tests to ensure our code is always working. This will free us from worrying if any new changes have broke any exiting features. As long as we have good, specific tests we can rest assure that our product is functioning as intended.

5 Networking

The networking aspect of our code is less certain. We want to implement a P2P system because we feel it would be the most fair in terms of network latency but it will likely be much harder to implement.

The client-server approach is probably easier but one player is guaranteed to have latency while the other is not. For that reason, we prefer the P2P approach.

This is why we have a sandbox area in repository. We plan on creating many prototypes that we can merge into the main branch of code when we feel they work well enough.