315 Week 4

**1. Initial list of overall app features, and the subsequent week’s project backlog**

This will be a web app using Three.js and Cannon.js. It will have 3D Models of the Earth, Arklets (small spacecraft with survivors), Izzy (the International Space Station), and fragments of the destroyed moon.

All ship motion will be controlled by the “Parambulator” which is basically a command center.

Physics simulation of the Moon fragments flying towards earth, as well as the Arklets performing flocking maneuvers to avoid them. The physics engine will also need to detect collision between all objects.

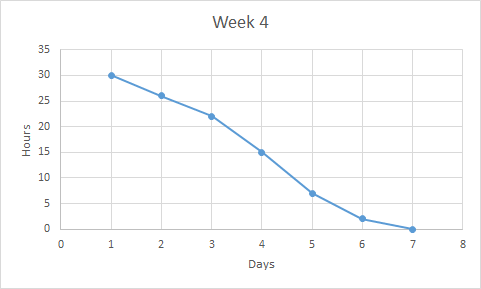
Each of the ships will also keep track of the number of people aboard, as well as the amount of fuel remaining.

App overall features:

* 3D Environment
* Space background
* Earth model
* Basic camera movement (rotate, zoom)
* Izzy and the Arklets that orbit the earth
* Moon and its resulting fragmentation after it blows up
* Html intro screen
* Option for number of Arklets
* The ability to pause the simulation by pressing p
* Collision detection between Arklets, IZZY, Earth, and the asteroids
* A flocking system to make the Arklets follow IZZY around the earth.
* AI implemented that gives the arklets the ability to dodge meteors
* Clickable meshes via raycaster
* There are a 2 different game modes

**2. User stories in that week’s scrum backlog and weekly burndown chart, SCRUM dev plan,**

Burndown Chart:



User Stories:

User story: “As a user, I want to see the arklets dodge meteors.”

User story: “As a user, I want to be able to play a third person survival game that is based in this .”

User story: “As a user, I want to see the artificial intelligence aspect of the simulation.”

User story: “As a user, I want a version of the game that works on a slow computer and has a lot of arklets”

**3. Notes from the SCRUM Master of your scrum’s daily 15-minute stand-up (status, plans, issues)**

Status: We have finished the code and now all we have to do is put some finishing touches on our presentation.

Plans: After finishing the code, we simply need to practice and prepare for the final presentation.

Issues: We noticed some lag towards the end of development. Reducing the number of arklets was a potential fix. We could also change the arklets to sprites. We decided to do both and create one simulation with a small number of high quality arklets, and one simulation with a large number of sprites for arklets and meteors.

Work from Wesley: I added game music, worked on the front page, worked on adding sprites. Added various touch ups to the game and simulation.

Work from Inaki: I implemented the half moon this time around. I made two game modes, ones that allow for the player to fly around and interact with the environment. One of the game modes is intended to be a survival game, with the goal being to not crash into the environment. The other is to destroy a set amount of objects in the map. I also improved the physics and meteor functions to allow for the map to be filled with meteors.

Work from Colin: Start screens for both the simulation and the game. Updated home page. Implemented 2D-sprite version of the simulation for performance.Added speed controls to the simulation.

Work from John: I implemented an AI that is simple in nature. It moves the arklets up and down based on the position of the incoming meteor. I also compartmentalized some of the functions that the AI uses so that it can also move the arklets back to IZZY if they move too far from IZZY.

**4. Summary report of End-of-week SCRUM review meeting and demo status**

During this week’s sprint we focused on AI implementation, making the arklets and meteors into sprites for another version of the simulation, and completely fleshing out the game. We were able to fully implement a AI early detection system into our simulation. We also made another version of the simulation that has sprites instead of full 3D models. This enabled us to add more aklets to the screen since it is not as memory intensive. We were also able to completely flush out the game so that it is a better experience for the user. We then updated the intro html screen to reflect the changes to our games and simulations.

**5. The demo code, which should be usable on a browser (specify browser and version)**

The demo code is included in the directory. It runs on a MacBook with Safari 9.1 or on Windows with Microsoft Edge . It requires that a computer is capable of running Shader algorithms on its GPU. If you wish to run the simulation on google chrome you need to enable -allow-file-access-from-files flag. You do this by entering "C:\the path to chrome\chrome.exe" -allow-file-access-from-files in the command prompt in windows.

Be aware that the 3d version of the simulation, and both of the games, are very CPU and RAM intensive. They require good machines to operate at the desires frames per second.