315 Week 3

**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:

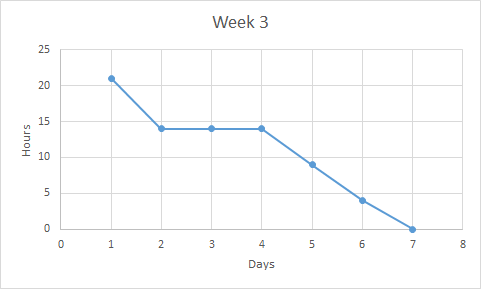
* 3D Environment
* Space background
* Earth model
* Basic camera movement (rotate, zoom)
* Izzy and the Arklets that orbit the earth
* Moon and its resulting fragmentation after it blows up
* Html intro screen
* Option for number of Arklets
* The ability to pause the simulation by pressing p
* Collision detection between Arklets, IZZY, Earth, and the asteroids
* A flocking system to make the Arklets follow IZZY around the earth.

Sprint 4 backlog:

* Finalize AI implementation by allowing the arklets and IZZY to dodge all obstacles coming their way without fail.
* Make an option for the number of arklets

**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 that a collision between an arklet and a meteor has a resulting action.”

User story: “As a user, I want to see where the sun is and for the moon exploding to be more dramatic.”

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

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

Status: We are on point for where we need to be at this point in development. Our next move will be to implement the remaining features and put final touches on the code so it can be turned in.

Plans: We need to continue work as planned. Our schedule seems to be working well and does not need adjustment. All that we have left to do is complete the implementation of the AI.

Issues: The A.I. will be a challenge because of the natural way that flocking is implemented.

Work from Wesley: I worked on making the orbits more realistic by changing their velocities and orientations. I added swarm flocking to the Arklets. They now return to their position (though slowly).

Work from Inaki: I made blender images for the half moon and learned how to use photoshop to include the sun’s and planet’s images into the simulation.

Work from Colin: I added a sun, and updated the project homepage. I also researched flocking algorithms, and their implementation.

Work from John: I did research on how collision is detected in CANNON.js. I was able to take what I learned and implement a collision detection system into our simulation using the utilities that CANNON.js has.

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

During this week’s sprint we focused on asteroid collision, and AI implementation. We were able to fully implement a collision detection system into our simulation. We also added a sun to the background in-order to make the background more interesting to look at. We have also changed the lighting system so that when the moon blows up the light changes to make the action more dramatic.

**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.