315 Week 2

**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.

Sprint 3 backlog:

* Improve orbits and collision detection.
* Initial AI setup, with basic flocking algorithm implemented.

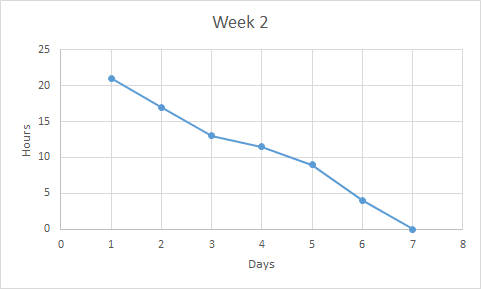
Sprint 4 backlog:

* Finalize AI implementation by allowing the arklets to dodge all obstacles coming their way without fail as well as the space station

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

User story: “As a user, I want to pause the simulation at any time..”

User story: “As a user, I want to choose the number of Arklets involved in the simulation.”



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

Status: Most of the development for the week is going smoothly. Blender is being used to create the Izzy and Arklets. Cannon.js is being utilized. We are implementing physics equations from our friends who are currently taking Physics 218. We are using centripetal force equations to simulate gravitational orbit.

Plans: We need to continue work as planned. Our schedule seems to be working well and does not need adjustment.

Issues: The main issue that we are facing right now deals with collision detection in CANNON.js. The exact protocols that CANNON.js has and the implementations are still unclear at this point in time.

Work from Wesley: I collaborated with physics students to develop ideas that helped simulate gravitational orbit. Due to the constraints of cannon.js, we are simulating gravity using centripetal force equations. I implemented an instance of this which set the groundwork for my teammates to set their objects (Izzy, arklets, etc.) into orbit. Later on, as new asteroids were being generated, they would fly in non-orbital paths. I generated orthogonal vectors to use as initial velocities for them which allows them to fly in orbit (with a random path rather than pre-determined path).

Work from Inaki: I developed the arklet and meteor designs this week. For that I had to learn how to use the software blender. As such I also worked on getting a lot of arklets to appear and each have their own physics and be on their own, which is an improvement over the way we had it before where all the arklets were all contained in a container. I also implemented the asteroids spawning from the moon. The way my implementation worked is that at several time intervals the asteroid would split up at the same position as another asteroid. My implementation caused the asteroids to fly out since they immediately collided, but that was fixed. I also modularized the functions to where each object had it’s own initialization function, to where before they were all under the same.

Work from Colin: I made the project homepage, index.html. I added some useful features to our main app, such as a start screen, pause function, mesh click detection, camera lock, and debug mode. I also created the Izzy model.

Work from John: I helped with some of the conceptualization during the sprint and added some minor features.

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

This week’s sprint was very productive. We now have a project homepage, many more 3D models, and a working physics system. The code is much cleaner, and is much more thoroughly commented.

We have a 3D model for Izzy which includes an asteroid that Izzy is mining. We have a 3D model for the asteroids, and for the arklets. These were all generated in Blender. They are synchronized with a cannon.js “physics body” of similar size that controls their movements.

Our demo status is where we want it to be. The asteroids, arklets, and Izzy orbit. We need an elegant way to handle collision detection and flocking for next week.

**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.