Project Name:

- “Asteroid Adventure”

Project Group (number + members): #04

- Meekit Patel – Created game menu/pages, soundtrack, page transitions
- Junkai Hu – Backend game play/controls, power ups, art, sound effects
- Chris Conover – High scores, page layout/constraints

Project Location:

- https://github.com/mpatel18/AsteriodGame

Description of Project:

For this term’s end of semester project, our group decided to design an asteroid shooter game that is loosely inspired by the 1979 classic, “Asteroids” by Atari. To make the game more playable on mobile devices however, we chose to replace the traditional omni-directional movement system of the original game with side-to-side movement and an infinitely scrolling background with random asteroid spawns. Furthermore, we chose to utilize the iPhone’s on-board accelerometer system to regulate the horizontal positioning of the user’s spaceship to avoid common gripes with on-screen controls. Shooting is accomplished by a touch action anywhere on the screen, which is how the player destroys the asteroids that continually “fall” towards the user’s ship.

In terms of the actual gameplay, users are placed in a space-like environment where they are tasked with staying alive for as long as possible while navigating an endless asteroid field. Players can choose to avoid these falling projectiles or destroy them by firing their spaceship’s on-board laser weapon; but if the user’s ship comes in contact with an asteroid, they will lose one of their three given lives. Upon crashing into asteroids three times, the game ends. This concept would be boring however without added content to make every play through unique. So, to account for this fact our team implemented a powerup system, various “debuff” objects and the possibility for layered asteroid objects. Powerups have a randomized chance of spawning any time a player destroys an asteroid and upon making contact with that powerup will provide the player with the buff related to the icon shown. Powerups include an extra life, the ability to slow down oncoming asteroids, a 100-point increase, the ability to shoot three projectiles at a time, double score, and the ability to clear the current screen of all asteroids. “Debuffs” on the other hand are undesirable and should be avoided by the player at all costs. These “debuffs” also have a random chance of spawning whenever an asteroid is destroyed and can increase the speed in which asteroids fall, instantly kill the player, or reduce the player’s number of projectiles per shot to one (assuming they were shooting three at a time previously). For added variability in each played game, asteroids have a chance of being layered when they spawn onto the playing field. Just as the name implies, layered asteroids cannot be destroyed by a single shot, but will

instead break into a smaller asteroid that requires another shot to fully disintegrate. Points are calculated based on the total number of asteroids destroyed over the course of a game, adding of course 100 points for every point boosting power up collected over the same time period.

In addition to the discussed gameplay, our team also chose to include several other options on the main interface of the application that include high scores, helper, and mute. The high scores menu option will take the user to a separate screen that shows them the value and date/time of their ten highest recorded scores. The "helper" menu option brings the user to a screen that displays all of the possible powerup and debuff options that could spawn upon destroying an asteroid and teaches the users what icons relate to each effect. Finally, the mute button is responsible for silencing all in-game sounds as well as the background soundtrack that would otherwise play over the course of a game.

Features of This Project:

Our project contains the following:

- Made use of new frameworks/technologies such as SpriteKit, GameplayKit, and CoreMotion
- Assigned different CollisionBitMask and CategoryBitMask objects (from PhysicsBody) to handle various collision instances between objects
- Custom sprites for app components such as the player's ship, power ups, asteroids, etc.
    - Custom artwork
    - Various visual animations
- Sound effects for various in game events as well as a looping background soundtrack
- Persistent storage of user high scores
- Multiple views for different user prompted requests from the main menu
- Constrained UI elements for optimal performance on a range of device models and sizes
- Dynamic and unique user gameplay for independent game instances
- Real time game play modifications through the use of power ups

Evaluation of Goals Met / Goals Unmet:

Our team was able to meet all of the "minimum product goals" that we set out for ourselves in the "first checkpoint" exercise. In addition to these basic goals, our team was able to meet half of the stretch goals as well, which include the implementation of a multi-layered/dynamic asteroid object, and full-screen animation/visual effects. One stretch goal that was not technically met was the possibility of implementing an omni-directional movement system similar to that of the original Atari game. We later decided however that this particular style of movement could not be conducive to seamless gameplay on a mobile device and instead opted for strictly horizontal movement of the player's ship. Finally, the only real stretch goal that was not met was the possible implementation of several unique game modes that were all in the spirit of this core game. Our team figured that it would be better however to have a more

integrated and seamless “central” gaming experience in a single game mode, then several underdeveloped yet unique auxiliary game modes.