Project Name: Space Shooter X

Group Name: lgd

Group Members: Yuxuan Bu, Shiyuan Duan, Yizhou Li.

Project introduction:

We propose an IOS based 3D shooting game. Player controls an aircraft and fires bullets to defeat various enemies. There will be a boss at each level and the player has to defeat it in order to complete a level. To win the game, player must complete all levels. The potential of this project is huge and we are planning on extending this project after this course but the plan in this proposal is solely delivering a demo,by the end of this semester, with all the core features that make up the game.

Minimal product goals:

- Using swift scene kit to construct a game like space shooter game.
- Design different game levels by each group member.
- Design AI of the Boss in each game level.
- Global scoreboard is stored in the backend database.
- Game UI (Health bar, items, score, settings & etc)

Stretch goals:

- Multiple players.
- Using a swift AR kit to display the 3D model of planes on the table before starting the game.
- Using swift Object capture API to capture the objects from the player's graph, and generate the customized 3D model.

Projected timeline:

- 9.30
  - Finish the final version of the project proposal.
  - Start project report.
- 11.24
  - Start project video.

- Finish the final version of the project (Basic app complete, stretch goals partially complete).
- 12.1
  - Finish the project video.
  - Finish project report.

Project Report:

Developing a swift project for the iOS platform is both unfamiliar and fascinating for our group. Although we have substantial knowledge in programming applications, we have not tried to develop in Apple's swift. However, with more corresponding knowledge we learned, and more progress we made throughout the semester, we could finally deliver a polished, interesting app in the end. Let me introduce the development process of our app SpaceShooterX.

In the beginning, our group mastered the foundation of machine learning, and game development, so we vaguely know we are going to develop an app in these directions. Thus, we made a schedule to brainstorm ideas about this project. Throughout this process, we first googled keywords such as "Swift projects for beginners", "Swift machine learning projects", "Swift game development". Then we searched the App store for inspiration. After we did the research, we decided to narrow down to two project ideas. One is to make a food planning app, and the other is to make a 3D game.

Our group members spend a few days discussing the pros and cons of doing which project. More specifically, which one is more meaningful, more technologically achievable, more appealing to the rest of the class. At first we seriously considered choosing the food app, because we are interested and good at machine learning, and this app has higher potential in this aspect. Namely, we gather the data of customer's eating habits, and apply our machine learning algorithm to schedule better and healthier food for our customers. Later we realized that to achieve our goal, we have to spend more time developing the algorithm. Additionally, since it's a start up app, we don't have enough data generated by the customers. Even Though, we could come up with a functional algorithm, we could not guarantee it is reliable and suits people's needs. Thus, we give up this idea, and think maybe we could develop a game by ourselves.

Later in the class, we saw some sample projects developed last semester. Some game projects like meteoroid have intrigued us, because the game itself is nice and easy. Additionally, it's definitely achievable and acceptable in less than

two months. Even better, our group has made the decision to make it 3D, because, why not? People love to feel the depth of an object, and want to see the game as photo realistically as possible.

Now, our group entered a detailed planning phase. Specifically, what kind of gameplay do we want our game to have? Do we want to have the same as the other “meteoroid”? How do we implement the enemies? Are they all flying objects such as meteoroids? Or some other kind of spaceship? If so, how do we design their moving trajectory, and abilities? What about the graphic? How to draw a background? How to do particle systems in swift? What 3D models should we choose? Should we buy some, or make our own? Those are the questions we need to consider first. Luckily we all have game development experience in Unity engine, so we know how to solve those concerns. Thus, the more important thing for us to consider is to find the game developing framework in swift, and learn and adapt to the new framework.

After doing some research. We have found that scenekit is used to develop 3D stuff in swift, and by reading tutorials from apple’s developer documentation, and third party websites such as raywenderlich and hackingwithswift. We get to understand the framework’s basic usage, such as what is SCNNode, how to place the camera node, how to manipulate the game’s render loop, etc. Now, we understand what tools we should use to build the basic functionality of our app. Then, we begin to wonder if the same tools such as scenekit can be applied in virtual reality or multiplayer games, if we want to extend the projects after we finish. Thus, we searched multiple Apple’s official forums, and posts, and we understand that AR kits and Reality kits are used for this purpose. Plus, we could use the game center module to incorporate multiplayer functionality. At this stage, we have understood the whole picture of our projects, and we start to implement the scripts.

Team member Yuxuan Bu has some substantial experiences in three dimensional game design, and he already has the demo space shooter game made in Unity engine. Naturally, he would do game’s infrastructures, such as scene establishment, scene transitions, gameobjects’ protocols, damage systems, etc.

Team members Shiyuan Duan and Yizhou Li are working on level design. Generally, we implement the moving and firing logic of the enemy AI, and also apply models and particle systems to them.

In Shiyuan’s work, there are 3 types of enemies and one boss designed. They are blinkers, tanks and sentinels. Blinkers teleport to a random position and

fire a bullet towards the player so the player will have a hard time locating it and avoiding the bullets. Tanks are not able to move but they have more health. They are equipped with burst shotguns which will make them more powerful. Sentinals roams randomly and they are able to fire tracking missiles. The boss designed is equipped with both shotgun and sniper. Also, the boss has a skill, the Warden's Eye. When the skill is active, player may not make any actions, otherwise player will be locked by hundreds of tracking missiles which are impossible to dodge.

In Yizhou's work, there are 5 types of enemies and one boss designed. They are meteoroid, enemy216, enemy351, enemy320, enemy330, and boss436. The meteoroids look just like their name represents "meteoroid". Their movement trajectory is straight, and they randomly spawn at the farside of the screen. If they hit the player, the player goes straight into death, and there are no other ways to remedy. Enemy216 is a small type of spacecraft. It usually comes with 4 to 5 within one group, and spawn either at the left side of screen, or right side of the screen. It has medium speed, low damage, higher shooting frequency characteristics. Sometimes, they come in the shape of interesting words, such as "I", "love", "swift", etc. Enemy351 is a heavy armor aircraft carrier, and it has low movement speed, and surprisingly no damage. However, it can spawn other two special high ability aircraft periodically. The first one's name is enemy320, the other's name is enemy330. Enemy320 is built for protecting its parent enemy351. It moves as their parents move, and it will circle around its parent to take bullet damage from the player. What great about the enemy320 is that it can absorb the damage and give it to its parent, so the enemy351 becomes more tanky. The moral of this ability is to tell the player to try to shoot more accurately to enemy351, and avoid shooting enemy320. On the other hand, we have enemy330, which is built for attacking the player, and limiting the player's move. Apparently, enemy330 will have higher damage, and movement speed, but with less armor. Now we introduce the boss436, which is an ultimate killing machine. It has high damage, and ability to summon all types of enemy. It has a higher shooting rate and split shoot attack pattern, which reminds the player to move carefully and predict which way is safer. Apart from the enemy AI we designed, we also need to take care of the orders that the enemies show up, and we also need to make sure the game is balanced, and the player won't be defeated by intense bullets.