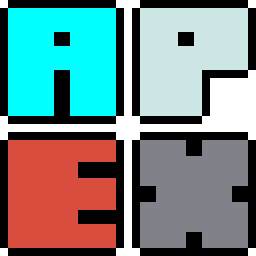
**APEX Postmortem**

**GAM150**



## **Introduction**

“APEX” is a 2D local-multiplayer platformer action game based on knock-back. Player should interrupt the opponent player and achieve the goal of the map, using knock-back action and several items.

## **Development Team**

Jookyung planned to do a team with Wonju at winter vacation. During winter break, Wonju and Haewon lived together and Haewon joined our team with Junseok whom was originally on Haewon’s team. Jookyung was interested in Technical Director, Wonju took a role of Lead Designer, Haewon did Test Manager, and Junseok was selected for Producer. We did conference often for our game and shared many ideas.

## **Tools**

**GAM150 Provided Engine**

In GAM150, Provided Engine supports the c/c++ languages, so we can use both of languages. WE can use the engine by using professor and TA.

**Visual Studio 2017**

We use to use the Visual Studio 2017. This support the ‘Debug’ and ‘Release’ mode, so we can control the mode when we want to check our feature work well or check the errors.

**piskelapp.com**

It is a tool for drawing pictures of our game. I drew all the game pictures on this site. This tool provides pictures in 8 bits and can also display animation effects by connecting pictures in different dot formats. The good thing about this tool is that it is easy to express dot-type pictures with a mouse and keyboard.

**opengameart.org**

It's a tool to find the sound of our game. There are lots of game effects and bgm. The important thing is that most of the sounds are not copyrighted.

**SourceTree**

It's a tool for sharing files in our game. I studied this tool a lot to share game files. The tool created a master path and branched files and shared them across multiple paths.

**What went Right:**

**Designed by Everyone**

* Since we communicated a lot, we also considered our game’s concept together. When all of team members met together first time, we decided what game to make. It was not a perfect idea, but eventually we polished our game design to what we really wanted.

**Communication**

* Our team almost met everyday while taking school courses. It naturally resulted team meeting and maintenance of our game. APEX kept in touch by using messenger even if we were not together. By these communications, we could know any problems from our game and fix them immediately.

**Open Playtest**

* Since we focused on just completing APEX’s functionalities during development, we didn’t know how it will be shown and what players will feel in real game. In addition, since APEX’s concept changed a week before open playtest, there was no information about real play. Thus, open playtest was very important opportunity for us.
* From open playtest, APEX got a lot of feedbacks from testers. Most of feedbacks were about APEX’s balancing, natural motions and minor things we could not catch. Especially, APEX got many feedbacks about the game’s tempo and visualized goal of the game. These feedbacks were helpful to making the final version of APEX.

**Professor Usage**

* APEX is a game that can win by achieving goal while battling the opponent. Due to characteristic of battle game, we struggled with physics like knock-back. Implementing players being knocked back was hard, but the real problem was how to make knock-back look natural. In order to solve this problem, APEX took a meeting with professor. Professor gave us many useful functions like easing out and in. These functions helped us to make our knock-back look cleaner.

**Better understanding of game development**

* We had to observe example from variety games, and this were very helpful to understanding game development. For example, APEX has 2 modes in game, domination and coin collection. To decide modes in the game, we referred a lot of games and most helpful game was Overwatch and Mario. By this sequence, we could understand well about what components can make game more fun or boring.

**What went wrong**

**Time management**

* There were CS230 and CS170 class in this semester, and both classes required lots of efforts to follow the class compared to last semester. However, at the beginning of the project, we couldn’t realized that and kept working slowly like last semester. However, stuffs and assignments from classes were stacked more and more, and because of that, we had to spend a lot of time to catch classes.
* Since that, we couldn’t spent enough time on our project, so we picked this as our problem and set mitigation in our report. However, it didn’t get improvement, and we couldn’t work enough for our project until final submission.

**No details in documents**

* While developing APEX, we didn’t put enough detail on our documents. It was not a problem at first. However, after developed the main functionality of our game, there were problems while adding sub functionalities, such as items. We had to add several items to APEX. However, since we didn’t put the details about these things, we argued about how they should work exactly several times, and spent more time than we expected.

**Sudden change of game concept during development**

* At the beginning of the project, our game’s concept was focused on pushing opponent and climbing. However, in alpha presentation, we got a feedback from our instructor that the concept of our game is not clear enough.
* Since we were busy with assignments from other classes, we decided to change the concept of our game the week before open playtest. This caused crunch mode until open playtest. Also, after playtest, there was no enough time to add more detail to our game, thus the result of project could not satisfied us.

**Not clean code**

* In code, if we needed redundant functionalities in different file, we just copy and pasted our code and changed that a little bit to make it work in that file. It made our code more longer and difficult to read. To solve this problem, we tried to arrange our code, but it failed since there was no enough time.

**Bad Usage source control**

* At the start of gam150 class, Professor David, recommended us to use Sourcetree which is source code controlling program. While using this program, we had a problem because of immature usage. For example, we uploaded the code, but on the program it did not update so we spent many time to find our latest version of our project. In order to avoid these kind of problem, we were to use the program a lot and look up for instructions.