## Out&In Logo2.pngdizzypen.png

## **Introduction**

It is a 2D puzzle projectile game where player has to shoot a cannonball in certain angle to reach a goal through obstacles and hindering effects. The player has to shoot the ball to the final area that looks like a black hole. In addition, this game is very simple.

**Development Team**

HyunJun, BeomGeun and SeongHyun have lived in dormitory until previous semester. Living together, we shared our knowledge and friendship. We decided to work together while we were having lunch together at sushi restaurant. We talked individual’s exceeding skills and abilities and decided to work as a team. To elaborate individual’s abilities: HyunJun is good at leading the team and he has a good writing skill so he took part in this project as a producer. Beomgeun is exceptional at observing and analyzing so he took role as a test manager. SeongHyun who is good at math and physics took role of a technical director because he can manage physics part better than other members can. Finally, HyeonMin who is very creative and can make very motivational environment team took role of a lead designer.

**Tools**

1. Audacity was used to shorten the length of audio files. The site <http://audio.online-convert.com> was used to optimize the size for audio file.
2. We used paint.net, power point and paint to print some graphic material such as User Interface, characters.
3. Visual Studio and Warp Engine were used as coding, making the actual game.

**What went Right:**

1. ***Communication***
   1. Through communication of sharing game designs and concepts, we decided to make 2D projectile game like AngryBird. Many other ideas were discarded because of our abilities of making the game were not sure. We wanted to make sure that we could make game that we designed.
   2. At the beginning of the project, we had communication problems. One member was busy with his own personal problems. We were supposed to talk and communicate more often so that the team could have found the problems much earlier. It was hard to assign roles to members because we did not know where he was. However, after we had communications, the team understood his situation much better and team members were able to talk each other and assign roles to each other.
   3. After, the problem was solved, we communicated with each other more often than usual. We often discussed what was going right and wrong, decide whether we are going to keep the baby (ideas) or not, and how we can design the whole game for the future.
   4. We believe that the team had more discussions than other teams, so our teammates can get feedbacks from each other immediately and faster.
2. ***Plans and Schedules***
   1. Since not all team members are experienced with projects and codes, we were not sure what we have to plan for the project.
   2. So, we had to make all potential possibilities that the team can make for the project and we eliminated what we cannot do.
   3. With the help of documents that we made, the team were able to look what we have planned and predicted what we could do together for the project.
   4. The more team focused on project, we got better coding and fixing bugs as well.
3. ***Simplicity***
   1. Since the game design is very simple, when main game mechanism was implemented, speed of developing game was rapidly increased.
   2. All features that we needed were collision, reflection and physics between cannonball and walls. Once, basic features were completed, making levels and adding other objects are all we needed.
   3. Also, the map editor was made earlier than what we planned, making game levels and objects became very easy.
   4. Through open playtest, we were able to delete features that were too complicated for users. We made it very easy for players to understand so that players will not have hard time understanding the concept of ‘Out and In’.
4. ***Better understanding of game development***
5. In CS230 course, we learned how game engine works and states are managed. At the beginning, we did not understand fully why we need to learn them, however as we were actually developing the game, we had better understanding of how game was constructed and how important CS230 materials were.
6. It was hard to notice bugs and know how to fix them. However, as time goes by, ability to debug and find bugs was rapidly increased; we could tell we were improving.
7. ***PlayTesting***
8. The content of game was able to convey in a short amount of time. Through open playtest, the team was able to get many feedbacks from play testers.
9. The information from the playtest was so valuable and this was a big help of our game. We changed features that were mentioned from playtest survey such as a graphical issue and physics problems.

**What went Wrong:**

1. ***Lack of foreseeing the project.***
   1. This was the first time that we actually made the whole game project. Last semester, the team was only making prototypes of games. The whole team followed the release plan so that process of developing game will not go off from what we planned.
   2. Since the game was developed faster than we expected, the team members were very motivated to develop game at the first. However, as time goes by, motivation for developing game was slightly decreased after everything was implemented.
   3. Before the team submitted the final project, we had problems with file IO. We had to set up directories of reading and writing text files from project. We did not know until the last minute. If the team have thought about this more deeply and fundamentally, we probably have been able to notice this problem right away.
2. ***Lack of coding skills***
3. Our team did not have excellent programmers, so we started with hard coding. When we started, the format and how we coded were quite messy. Moreover, it was hard to understand other member’s codes and keep up with them.
4. Using comments, file headers, and function headers were not implemented through development, so other all team members had to count on each team members.
5. Also, in CS170 class, we learned concept of c++: inheritance and STL. We believe that it would have been much easier if we have utilized STL and inheritance to have better control objects that we created.
6. We were afraid to change codes because once it was established; it was very scared to change it.
7. ***Lack of Design***
8. Since team did not have design artists, we had lack of art assets for the game. Even we tried to make better art assets; they were comparatively looked lower than other team’s.
9. We mainly made images by using paint.net, the quality of images were not very high and resolution of images were low. So, when images were enlarged, it often got teared and sheared.
10. ***Communication***
11. Even though we had a good communication at first, because of busy individual schedules, we sometime had a hard time with the communication between team members.
12. We hesitated to point out the flaw that each member has. It caused our game slow progress.
13. ***Time management***
    1. The schedule and plan for developing games were not specifically and evenly divided to team members. Since we are thriving to become professionals in gaming industry, this was the most important lesson that we learned through this project.
    2. Even though we had good plans at the beginning, the team got quite loose at the end because all features were completed. We believe that if team members were more punctual for project, that we might have made more features and polished better for the project.
    3. Making installer for our game was different from others because we had to read and write text files for the game. We had to assign location for levels and saved data.

**How we managed to solve problems:**

There were many problems how we can code our project, with the help of TAs, we were able to solve problems that we faced such as developing physics, collision, file io and so on. Also, they gave us useful tips for coding and how to enhance our coding skills. It was very helpful.

In aspects of the communicating problem, we tried to meet each other and make a regular meeting so that team can make schedules for the future. Frequent contacts and regular meetings have definitely contributed to game development.

**Future**

We believe that there are many lessons that we learned from this project. Communicating is very important among team members to check status of game development. Being brave enough to delete whole code and start all over again for improving quality of codes.

With the experience that we learned from this project, hopefully each individual overcome their mistakes and become more professional in the future.