Group: Filipino Laotian Object Oriented Programmers (FLOOP)
Members: Rolly Padua, Christian Martin, Marcus Lopez, Nicholas Souchitto
Game Title: BOMBER BOIZ

The goal of our project was to recreate the classic video game *Bomberman*. In this single player game, the player's task is to try to eliminate the enemy artificial intelligence (AI) with bombs dropped and detonated by the player. The pre-made map was a 9x9 matrix-like grid that was created by an object class called "Wall" which was inherited from an object class called "Rect".

In terms of implementation of object oriented programming concepts, as stated, we used inheritance to create the "wall" object which was the foundation of our project. Below is a snippet the derived class's constructor and the bass class's declaration.

```
class Rect {

        float x, y, width, height, center_x, center_y;
        bool player;
        int holds = 0;

        public:

                // Constructor and Destructor
                Rect(float x, float y, float width, float height);
                ~Rect();
Wall(float r, float g, float b, bool empty, bool breakable, int pos_index, float x, float y,
float width, float height) : Rect(x, y, width, height) {
                this->r = r;
                this->g = g;
                this->b = b;
                this->empty = empty;
                this->breakable = breakable;
                this->pos_index = pos_index
        }
```

Data Abstraction was heavily used in creating the user interface. Drawing functions were made in order to easily redraw and change the outlook of the game in real time. Below are some examples of functions used to show this data abstraction.

```
drawHero();
drawEnemy();
drawGridMap();
```

Much of the project process was put into creating the game's mechanics. For example, the code below illustrates an small interaction between the bomb dropped by the player and the walls around it.

```
switch (key) {
        case 'w':
                if (playerOne->getPos_Index() < 9) {
        // this means that the player is at the top row of the gridMap. Cannot move further up.
        std::cout << "Cannot move Up. Out of Bounds." << std::endl;
                        return false;
                }
```

In conclusion, the production of BOMBER BOIZ was a challenge yet entertaining project. Even though we attempted to utilize the object oriented programming concepts that we were taught by this semester, we understand that we still have much improvement can be made.

Time planning and division of labor

Marcus took the task of doing the textures for the entire game. Unfortunately, the end game implementation did not work together very well, so textures could not be implemented into the game very well. In addition to his work with the textures, Marcus also worked on the matrix grid, which was wonderfully done. Nicholas dedicated a majority of his time to implementing power ups. Christian did the implementation of wall.h, the implementation of the bomb and the AI implementation. I, Rolly, took the responsibility of creating the Hero.h and Hero.cpp files from scratch.

Collectively, all of us worked with game mechanics, debugging, and collaborating with each other through Skype. Some members helped more with debugging, and others helped more with the creation of the game mechanics. Of course, the reason why I say "some" of us because we always switched around. For example, I worked mostly with debugging code, but there were times where I would go into making game mechanics because I would be more productive at that section. Overall, I believe the division of labor was fair, especially when taking our schedules into consideration. Some of us had some finals Saturday and even earlier, excluding our OOP Final. Others had other projects to do.

Lessons Learned

During this project, I learned that creating a video game from scratch is significantly difficult to produce when working with a team. This was exacerbated due to a very close deadline. The reason this was so difficult was due to other group mates

having to study for exams or complete other projects. This conflicted with my schedule. Despite these difficulties, working in a group gave me real life experience on how fast games could be developed when cooperating with a competent team.

My ideas for approaching the project seemed similar to my group mates. There were many times that my methods were less efficient, and other times when my implementation was better. I believe that the most important lessons I have learned during this project was how to correctly implement code I learned in Object Oriented Programming and that I need to think of more innovative ideas on how to implement things in video games. This project was also a test to our time management skills, especially with various deadlines and test dates all in such a close time interval. As a group, we agreed that everyone did what they were able to do in the given time that we had left. Although we know we have a lot of room to improve, I believe that we did the best we could despite our circumstances.

Additional Comments

I surprisingly had fun during the production of this project, despite the time limitation. Despite the game being incomplete, the experience I received was most invaluable.