# **Costa Rica Technology Institute**

# **School of Computer Engineering**

# **Second Project**

# **Bomberman**

# **Students:**

# **Daniel Montero**

# **Kevin Hernández**

# **Sede San Carlos**

# **19/11/15**

# **Introduction**

The next project showing different ways in which we can develop and take advantage of object-oriented programming. In this case we will apply this to develop a video game that was very popular in his time. To implement all the functions you use this project we can develop and implement the knowledge acquired in class, so we will learn in a deeper way that knowledge. The game is a hero who is responsible for exploiting blocks and also used bombs to destroy the enemies. In the original game the hero powers that can be found maximixar your skills, you should also find a hidden key that will take you to win the game.

It should take into account not only the benefits of object-oriented progrmacion but the multithreaded programming for logical operation of the game are used, this should be applied on various items such as hero, enemies, barrels, explosives and among other things such as the time of departure. This is because the hero and other item can be moved independently of each other if all the threads would move sequentially are not used so the game would be poor or very boring.

# **Analisys of the problem**

Must develop all kinds to be used in the project structuring their relationships well.They should investigate various things like the use of threads in Java and a way in which you can print graphics so the game items. You think in advance using a logical matrix and other graphics. these are moved simultaneously for reading and writing on the graph matrix. Must implement three types of game, so easy, medium and hard difficulty mode, the difficulty depends also on the number of blocks. For other side should look for ways to work with design patterns so we should look where implement this. All items must be objects so that the matrix will be comprised of the same objects. you should investigate about using audio in the game. the parent must support graphical images of the above items during the game should carry a timer and heading east end should create a file to store the names of the players and your score.

# **Problem solution**

For the solution of the problem, we first had to identify as much classes as possible to get a view of what we had to do. Next, we had to figure out a way to do the matrix. To do that we created an Element matrix, so every element in the matrix will be from that class. After we had that, we proceed to create the 3 different types of matrix with the Factory Design Pattern. We implemented that so when the user chooses the difficulty, the Factory creates the appropriate type of matrix depending on that.

For the characters in the game, we made the Element class an extension of Thread, so every element can also be one. This way, the hero, balloons and barrels can move on their own. The MP3 class was created importing a library called javazoom. That class got in charge of playing the music. Every new MP3 object played the music in a new Thread so it can work independently. That is also the case with the Chronometer class. It is also a thread so it can work independently as well.

For the movement of the character, what we did was use The KeyListener to capture which key was pressed. Depending on that, the hero moves in the matrix. That technique was also used to place the bombs.

The bombs where the real problem, because they couldn’t go over the hero when they were placed. They do explode, but they don’t kill the enemies.

To place the door and the powers, there are two Booleans placed randomly in some blocks. When a bomb explodes a block, if that block had the Boolean set in true for the door or the power, instead of transforming in an empty space, it transforms in either the door, or a power. If the player runs through the door, he wins.

After a player wins, a small window appears so he can submit a nickname. After that, the program returns to the main menu. In that moment you can go check the high scores. When the player entered its nickname, the program saved that and the time (which stopped when the player won) in a file. There are 3 different files to store high scores for the 3 different difficulties. When you try to see the scores, the program opens the 3 files and, if there is something, it will appear on screen.

If the player loses, there will be the option to either retry or go back to the main menu. If the user chooses to replay, the same level will start over, but not with the elements in the same places, because that is chosen randomly every time.

# **Analysis of results**

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Implemented | Bugs | Not implemented |
| Objects in the game | x | Barrells don’t move as expected sometimes. Bombs don’t kill enemies if they are moving |  |
| Interface and Singleton | x |  |  |
| Different maps | x |  |  |
| Images for objects and a difficulty choosing screen | x | The JPanel sometimes have some issues reloading |  |
| Time is captured | x |  |  |
| Access Modifiers and sets & gets used correctly | x |  |  |
| Abstract Class with at least 2 methods | x |  |  |
| Exceptions | x | No exception of our own |  |
| Unit Tests | x | The Test for the Chronometer failed |  |
| Extra Points | x |  |  |

# **Conclusion**

With the completion of this project I manage to get a clearer idea of object-oriented programming and also managed to get more than ample knowledge studied in class. The project had a good outcome in their development and are able to implement all its functions properly. More knowledge about programming with achieving in turn design the game with the characters moving independently threads are obtained. I was able to apply to the project design patterns, also use files to store the best games. They learned more about how the games and their dependence on the use of threads for optimum performance are implemented.

# **Recommended**

The project was really good, it was fun and interesting, and there are no reviews for this. Maybe you could explain a little about what methods or ways to implement the bombs correctly. Outside, this project was very good.