**Software Engineering**

**Projectile Parry**

**Final Document**

**Table of contents**

1. **Introduction**
2. **Final Class Diagram**
3. **What We’ve learned**

**3.1 Randy**

**3.2 Audrey**

**3.3 Brian**

**3.4 Abaseen**

**3.5 William**

1. **User Guide**

**4.1 Installation Guide**

**4.1.1 Video Installation Guide**

**4.2 How to Play**

**4.3 Level Objective and Game Overview**

**4.4 Enemy**

**4.5 High Scores**

**4.6 Bonus Section**

1. **Introduction**

Our development of Projectile Parry was progressing as we used LWJGL3 as our graphical drivers library and IntelliJ Idea as the IDE for the project. Our project leader William created a github repository for the game and we worked on level designs for the title using handmade block layouts as shown in this image. In implementation Github copilot was used. Github copilot is an AI driven tool that assisted the project greatly with repetitive workloads like quickly and easily generating level layouts from diagrams in comments, along with helping to refactor code more easily.

**Figure 1.1 level design code.**

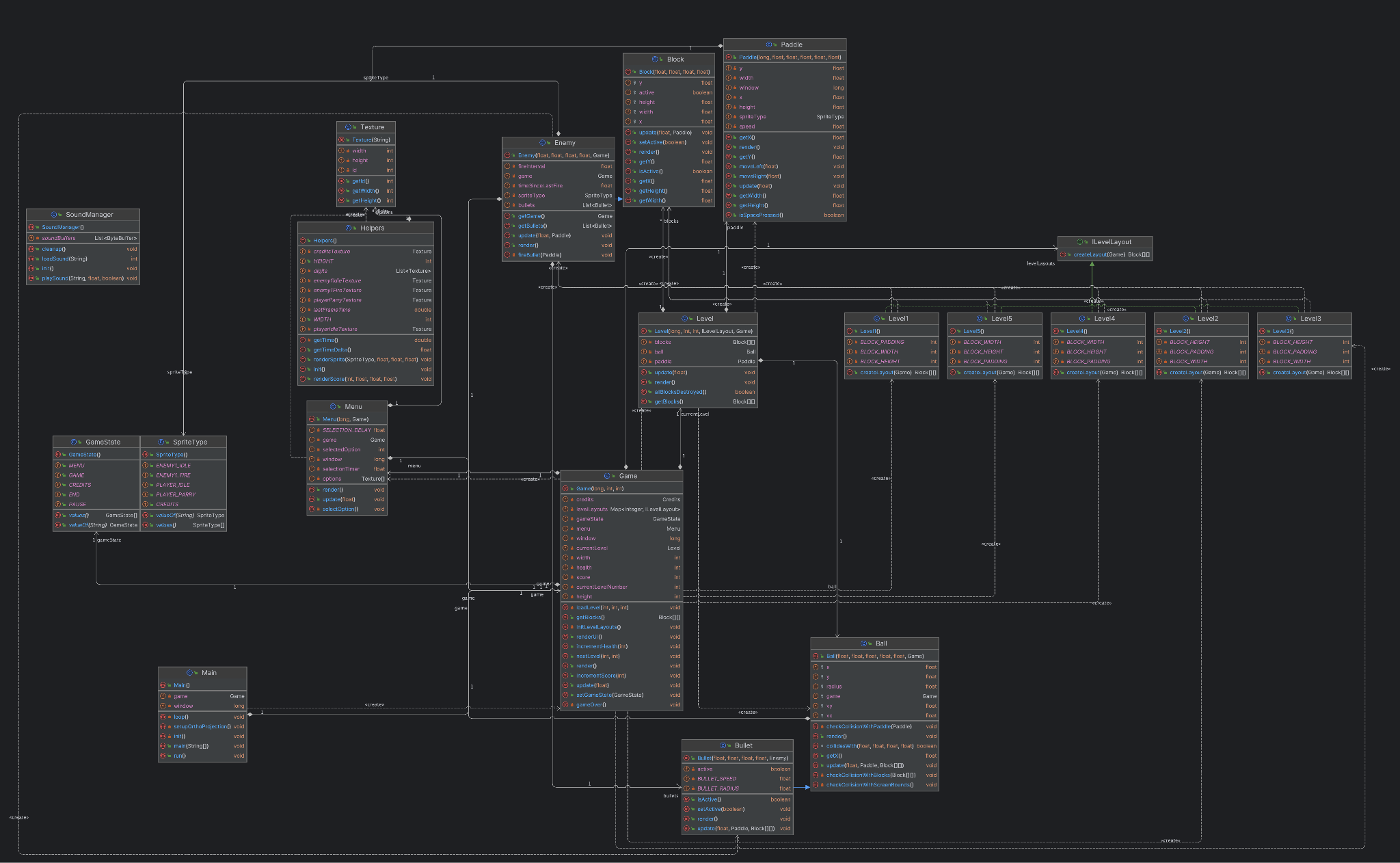
We also created custom art for the title’s enemies, player, and blocks. We had run into some issues adding our custom sprites and had considered using LWJGL 2 + Slick2D but after an attempt using it it was decided that the libraries in LWJGL 2 + Slick2D were too old and deprecated to feasibly use which meant hardcoding in LWJGL 3 was done instead. We were progressing well having implemented the basic player control, the custom player sprite, sound effects, one enemy type with its custom sprite, the ability for the player to parry enemy attacks, the player being able to interact with blocks and enemies, and the ability for the player to take damage and get a game over all being put into the game. We ended up with five unique level designs in the game along with scoring and credits.

The game also satisfies several non-functional requirements we had set out to hit including high performance due to the use of our graphics library, optimized resource management, fast and responsive gameplay, and quick load times.

We came across limits in our development process though. Our main limiting factor was time. A lot of time had to be spent hardcoding elements of the game’s graphics and due to that we were only able to get one enemy type into the game, the fireball enemy. We also we unable to use custom art for the blocks and block types. With more time we could have made a more complete project but due to constraints we had to cut features and details we had planned.

Our biggest loss due to the deadline was two other enemy types: the fast enemy and the standard enemy. We had created custom art for them which you can see in the bonus section of the user guide along with other cut elements and art. Menu and interfaces also had to be cut such as the pause menu and the variable settings for the player to decide things like language, difficulty, and controls.

**2. Final Class Diagram**

**Figure 2.1 Final Class Diagram**

The final class diagram for our title. You can see things like the individual levels having their own classes, there being less enemy and menu classes then we had before, and new classes such as “Helper” “Bullet” and “SpriteType”

**3. What We Learned**

**3.1 Randy**

Creation of clear diagrams was integral to the process of designing projectile parry. As without a clear understanding of what needed to go into the game, the code and the design would’ve been a lot harder. The diagrams also help show how the program works at a full scale level, so anyone who wants to add on to the project can do so. One day I want to be making larger scale video games, and in those games I want players to be able to create their own content (or mods) and the skills I’ve learned from this project will help with that.

**3.2 Audrey**

I learned a lot about creating diagrams and also about writing descriptions and notation of diagrams in order to explain their purposes to a team. Things like textual use cases to describe how a system is used and describing the attributes found in class diagrams was something I did not have experience in before now and I think is going to come in handy for future projects trying to convey what I want a user to experience and how a system is going to accomplish that. I also learned about how time consuming program development is and how even making something that seems simple can be very complex

Another thing I learned a lot about was game art. A lot of my art didn’t make it into the final game due to time constraints but it was a very interesting challenge to try and create artwork that fits the game's mechanics and conveys the game to the player. I will create more games in the future and getting experience with art now will be a great help.

**3.3 Brian**

The process of creating projectile parry provide a lot of insight into what game development may be. I learned alot about the steps required in order to reach a final project, or game prototype. Whether it was creating diagrams and descriptions for them to outline and explaining cases, or analyzing the code my colleagues developed to put this game onto our computers, there’s so many steps that I learned from. I learned alot about substem services and the scopes of that. I also became proficient at using lucid chart while developing the diagrams present in our project submissions.

**3.4 Abaseen**

While working on this project, I gained a lot of knowledge about how a full software application is created. The requirements document and making charts for each part of the system was completely new to me in a practical case. I have been learning about them for years but never had a reason to use them. Sequence diagrams and class diagrams were very helpful to learn to create and break down a program into smaller parts, and I'm sure it will help a lot in the future. I also learned how to import third-party packages into Java and how to use them. The code written by William made me realize how many moving parts are in a program, and how many pieces are needed.

**3.5 William**

This project taught me so much in terms of how to utilize a game library, how complexities of handling updates and graphical rendering really add up, and how to structure and work with a larger scale project in general. In addition I learned many of the nuances about efficiently using resources in such a way that is compatible with any machine in the Jar format. Overall, creating and writing the codebase for the project really opened my eyes to how even a simple idea can be extremely complex to piece together well, especially when factoring in external graphical and sound libraries.

**4. User Guide**

**System Requirements.**

Any system that can install Java 20 will be capable of running the game. You can download the JDK here.

<https://www.oracle.com/java/technologies/downloads/>

**4.1 Installation guide**

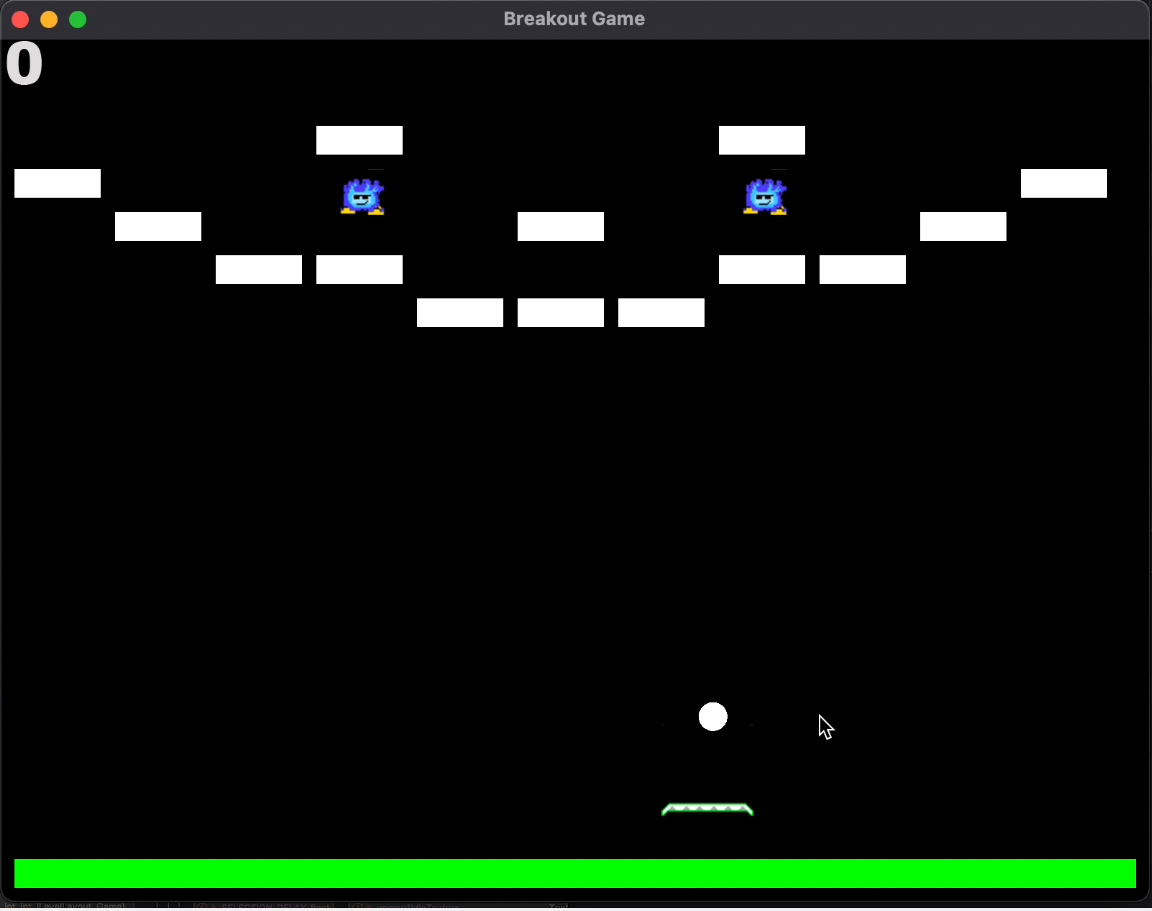
To install the game download the Jar file and on windows double click to run the title. On Mac after downloading the file run the command “java -XstartOnFirstThread -jar breakout.jar” in the MacOS Terminal.

**4.1.1 Video Installation Guide**

https://www.youtube.com/watch?v=aKnD6-tVUiI&list=LL&index=5

**4.2 How to play.**

To play the game you will need three buttons. You will need two buttons to move your paddle left and right and one button for the parry mechanic. The default buttons for this are the left and right arrow keys for left and right and space bar to parry. To quit the program you can use the escape key at any time. There are no current ways to remap the buttons used for the game.

**4.3 Level Objective and Game Overview**

The objective of each level is to destroy every enemy in a level by hitting them with a ball. You will be given your own “main ball” at the start of a level and can also get more balls by successfully parrying enemy attacks. There will be other obstacles in your way that will be destroyed by the balls similar to how you defeat enemies.

You will lose the game if you take enough damage that your health drops to 0. You take damage in two main ways. One is if your paddle gets hit by enemy balls and another is if your main ball goes off screen

**4.4 Enemy**

| The fireball enemy is the primary enemy in the game. They do not move quite as fast as the fast enemy but are plenty speedy and their fireball attack packs a punch. Be very careful around them.  The fireballs when parried will be a strong projectile for the player as it can burn through blocks with a single touch. |  |
| --- | --- |

**4.5 High Scores**

If you get a high score you will be asked to input your initials. The top ten scores will be saved locally to your game. There are no online leaderboards for this game.

**4.6 Bonus Section**

Here are elements we had ideas and art for but were unable to finish in time.

| The planned standard enemy of the game. Their fire and movement speed was meant to be low along with their damage.  When their attack is parried they provide the player with a standard ball. These were to be similar to the main ball the player has but more firepower would have been a benefit for you. |  |
| --- | --- |
| The unused fast enemy for the game. They were planned to move left and right across the screen quickly and shoot balls that movie quickly |  |