Bilkent University

Department of Computer Engineering

**CS 319 Term Project**

*Section 1*

*Group 1A*

*Walls and Warriors*

**Final Report**

Project Group Members

1. Dawood Muzammil Malik
2. Deniz Yüksel
3. Faruk Oruç
4. Mert Soydinç
5. Muhammed Salih Altun

**TA: Gulden Olgun**

**Supervisor: Eray Tüzün**

**Contents**

1. [Introduction 3](#_Toc533367157)

[2. Design Changes 3](#_Toc533367158)

[3. Lessons Learned 6](#_Toc533367159)

[4. User’s Guide 7](#_Toc533367160)

[4.1 System Requirements & Installation 7](#_Toc533367161)

[4.2 User Manual — How to build 7](#_Toc533367162)

[4.3 User Manual — How to Play 11](#_Toc533367163)

[5 Work Distribution 14](#_Toc533367164)

[6 References 16](#_Toc533367165)

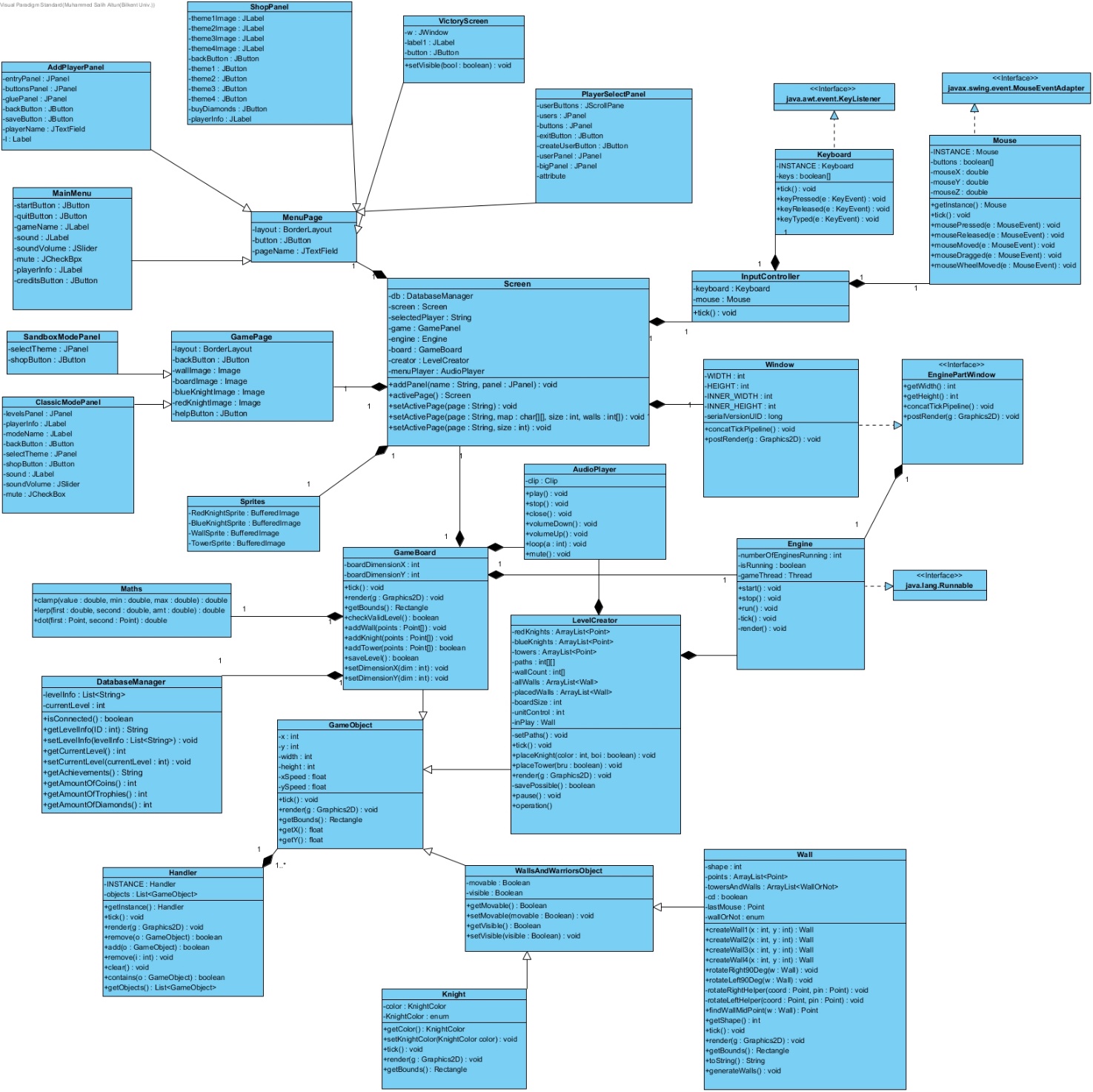
# Introduction

The implementation of the project is complete with a few changes in design due to some complications and different ideas during the implementation. The core logic of the game works correctly and in sync with the database. We changed the proposed themes which were a last moment thought before the analysis report. We believe the themes that we replaced them with are more creative and enhance the user experience more compared to what we proposed earlier. There are seemingly lots of changes between design and implementation and especially with iterations 1 and 2 and that is because we could not do well in iteration 1 and we had to change the whole way we were working together and communicating.

# Design Changes

Some changes were made to the design of the game compared to what was planned earlier. As the implementation progressed, some unexpected situations led to few changes in the class diagram made in iteration 2. Specifically, around the time of design report iteration 2, there were still some issues that had not been addressed in terms of the design of the game. We had to accept that the design we had was not going to be exactly reflected in the game’s code. Here we will provide the updated class diagram and sequence diagrams.

* The updated class diagram is as follows:



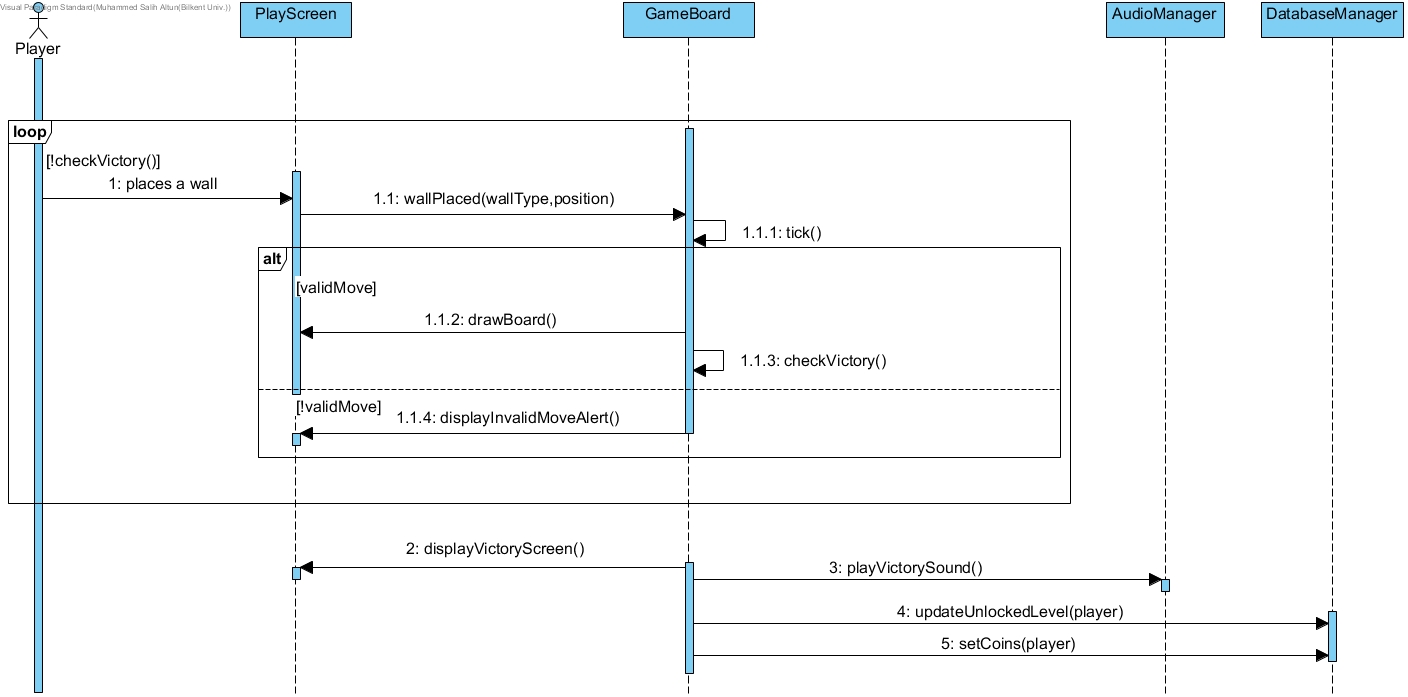
*Imgur Link for better quality:*

[*https://imgur.com/a/YpbWjC5*](https://imgur.com/a/YpbWjC5)

*Figure 1 Updated class diagram*

Since the implementation of user interface has changed from JavaFX to Swing, there has been some changes to accommodate it. The control of the whole game is in the hands of class Screen and GameBoard and LevelCreator classes are used in order to control the classic mode and sandbox modes. Also, AudioManager class was added later than others and is shown here.

* The changed Play Game sequence diagram is as follows:

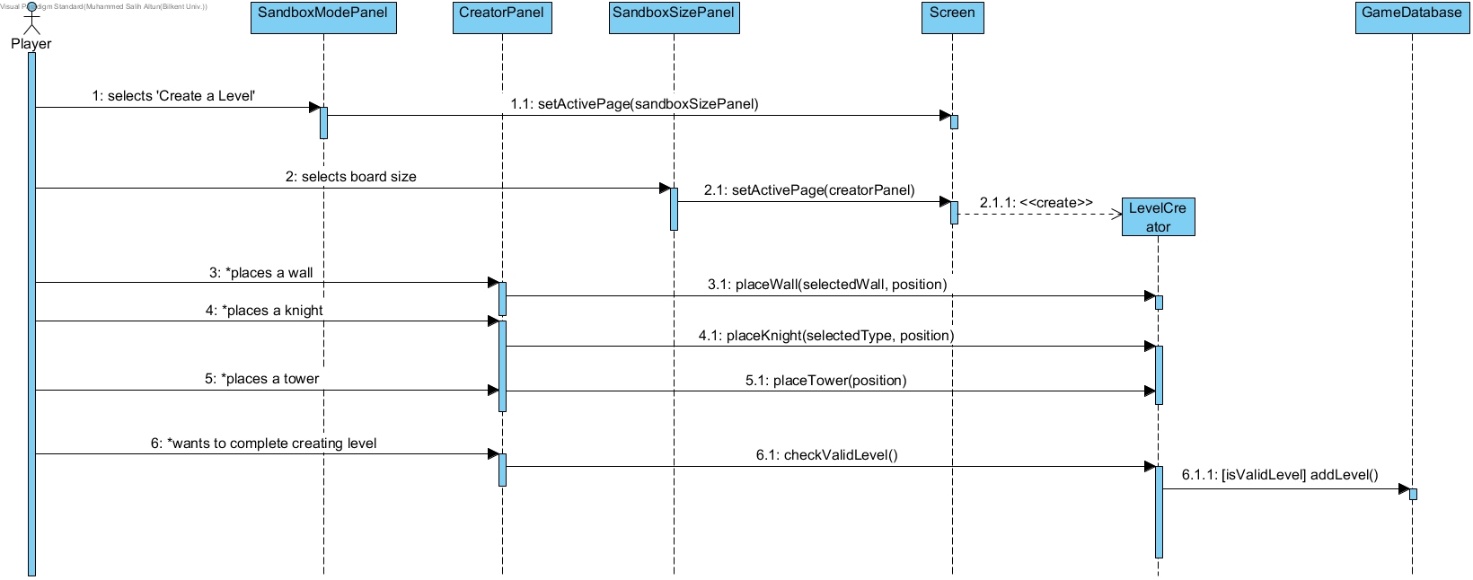


*Imgur Link for better quality:*

[*https://imgur.com/a/pN1zGnW*](https://imgur.com/a/pN1zGnW)

*Figure 2 Updated playGame sequence diagram*

The changes to this sequence diagram reflect the changes to the class diagram. The core logic has not changed but view classes have been changed.

* The changed Create Level sequence diagram is as follows:

*Imgur Link for better quality:*

[*https://imgur.com/a/WWOpv3l*](https://imgur.com/a/WWOpv3l)

*Figure 3 Updated Create Level sequence diagram*

Again, the core logic has not changed but some classes have, especially in the GUI package.

**3. Lessons Learned**

In this project we all have learned many things. We ran into many obstacles.

One of these issues were communication. Working as a group in sync is not easy. We had to learn the importance of communicating the hard way in iteration 1. We could not put the project together because we didn’t talk to each other enough and we didn’t have a working communication structure. After the first demo (Iteration 1), we sat down and defined roles in the group. We made a communication structure and work share that everyone was comfortable with.

The other issue was trying to implement the project with the proposed design. We learned that while implementing the project, design sometimes gets overlooked in favor of what works. Also, not everything that was proposed as a feature can be implemented due to different types of constraints.

For example, with our game’s additional sandbox mode feature, we envisioned a mode in which the player can make a board of any desired size and place any amount of game objects inside it and make very complicated levels. But in the implementation phase we realized that it was not feasible. Once the board dimensions exceed a certain amount, the cells and edges of the board become so small that playing the game or placing objects was not possible. As we preferred usability over functionality, we had to make a change in terms of requirements. We decided to let the player choose between three different board sizes: small, medium and large. We also realized we were not able to let the player draw their own walls in the sandbox mode. This was causing many different problems, so we decided to add two additional walls and not let the player draw walls of their own.

Another issue was keeping up with a schedule. To solve this problem we set up a Google Calendar which everyone could follow.

**4. User’s Guide**

**4.1 System Requirements & Installation**

Walls & Warriors can be run on any computer that has the Java Runtime Environment and MongoDB installed. All one needs is to download the .jar file on their consoles and he/she will be able to play the game.

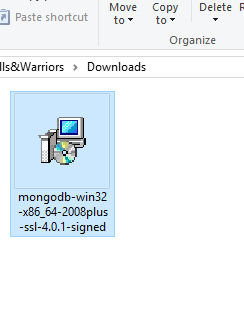
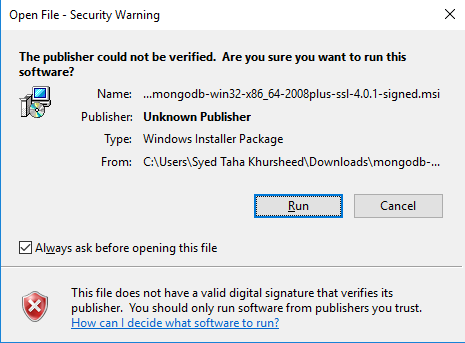
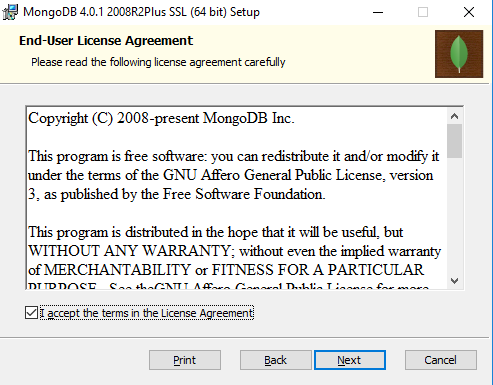
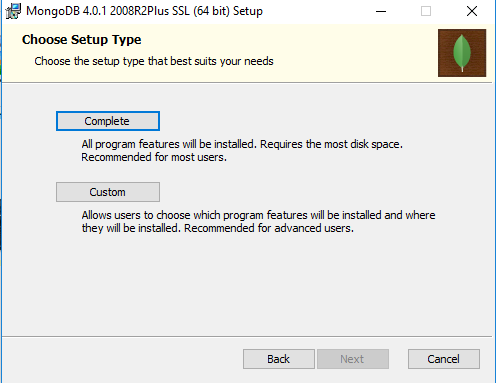
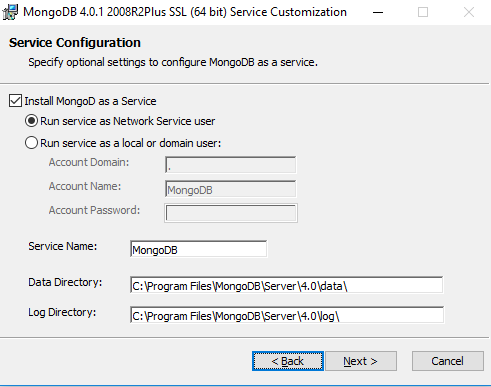
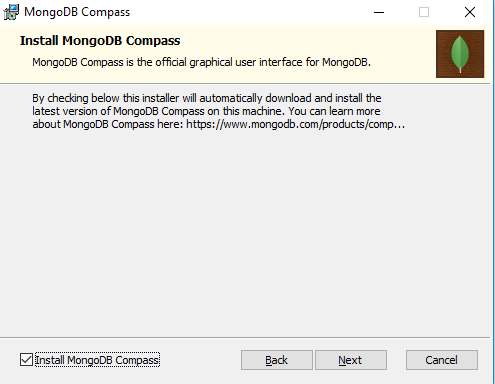
**4.2 User Manual — How to build**

**4.2.1 Running through Source Code**

1. Download MongoDB. Instructions to install MongoDB on your computer system can be found [here](#Mongo).
2. Open the project’s source code in an IDE.
3. Make sure folder named *res* is selected as the resource folder in the project.
4. Add reference to the mongo-java-driver-3.9.0.jar (included in the project folder) from the IDE
5. Compile the program.
6. The game screen should appear, click on *Play Game*.
7. Play the game.

**4.2.2** **Installing MongoDB**

Download the latest production release of MongoDB from [here](https://l.facebook.com/l.php?u=https%3A%2F%2Fdrive.google.com%2Fdrive%2Ffolders%2F1QSoURgBFEpA87ijblXink85CYciHL28t%3Fusp%3Dsharing%26fbclid%3DIwAR3-OVSADRXjpAyagf9fLGSiBARh979F_ZAcCr_yZgHWq07fgx2IwgToCi0&h=AT3RhpFPvJWrrFYY2WjgA3hWCXWjQaINRYdQ7hOyeUCa7A1M3YGsyG0iOxGm4NdXI_y8NA_8pMQSpoIfHA056PzOQJAeZfFU6Ua-Yfh-XarnmBR-jH_jGzkyUR32SMUpU1BTmw). Once downloaded, follow these steps to install MongoDB on your system.

1. Run the installer from the location where you downloaded the setup.
2. Click *Run.*
3. Click *Next*.
4. Agree *Terms and Conditions* and click *Next*.
5. Choose ***Complete*** setup type.
6. Without any changes, click *Next*.
7. Click *Next*.
8. Press *Install*.
9. MongoDB is now successfully installed on your system.

**Note:** For detailed explanation regarding the installation, click [here](https://www.coursera.org/lecture/introduction-mongodb/installing-mongodb-on-windows-Hadhu) for a video tutorial available on *Coursera*.

[ this space is left blank intentionally ]

* 1. **User Manual — How to Play**

1. Run the game.
2. Select the user account you wish to play with. You can also create a new user account.



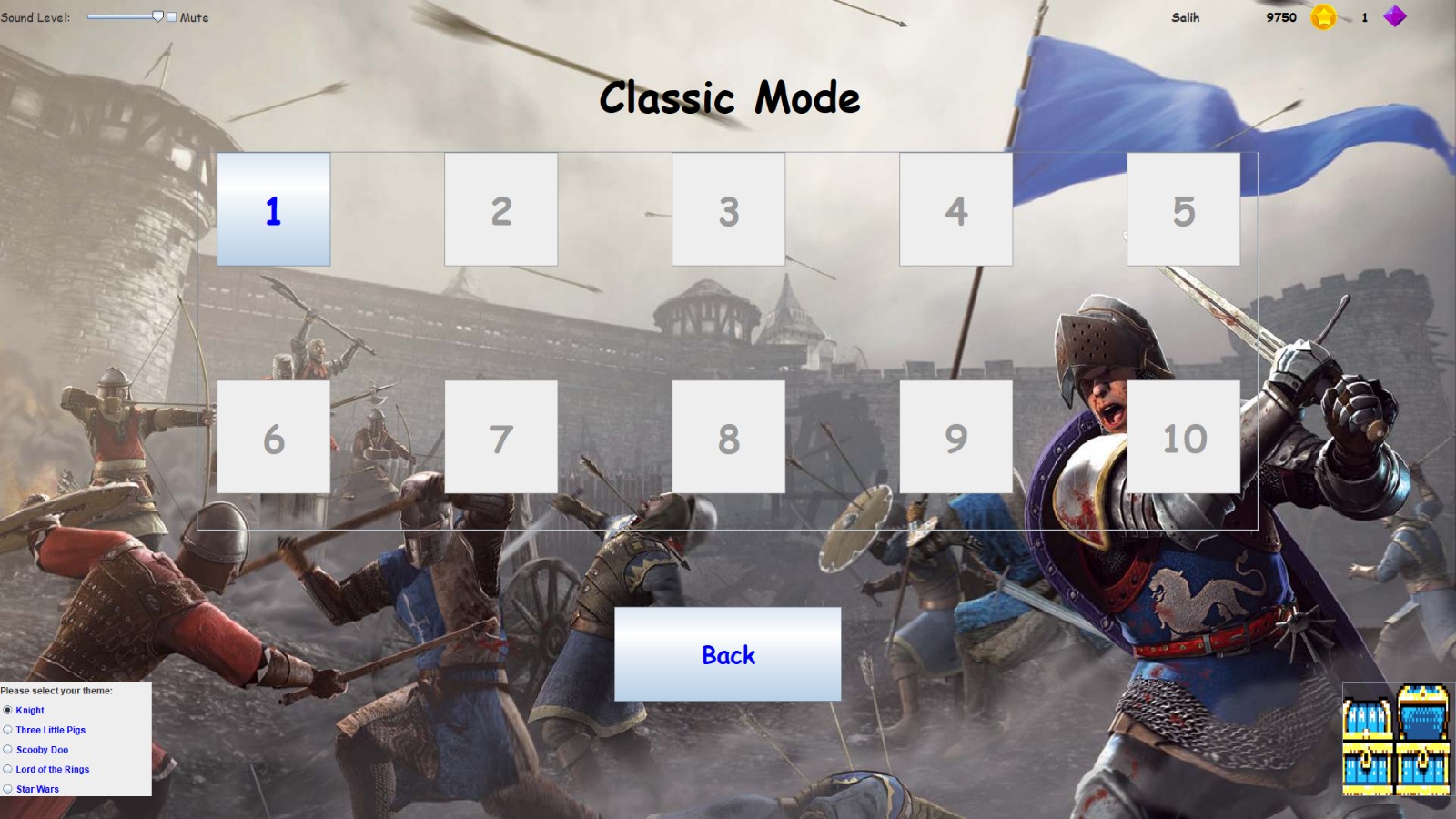
1. You will now be directed to the *Main Menu*



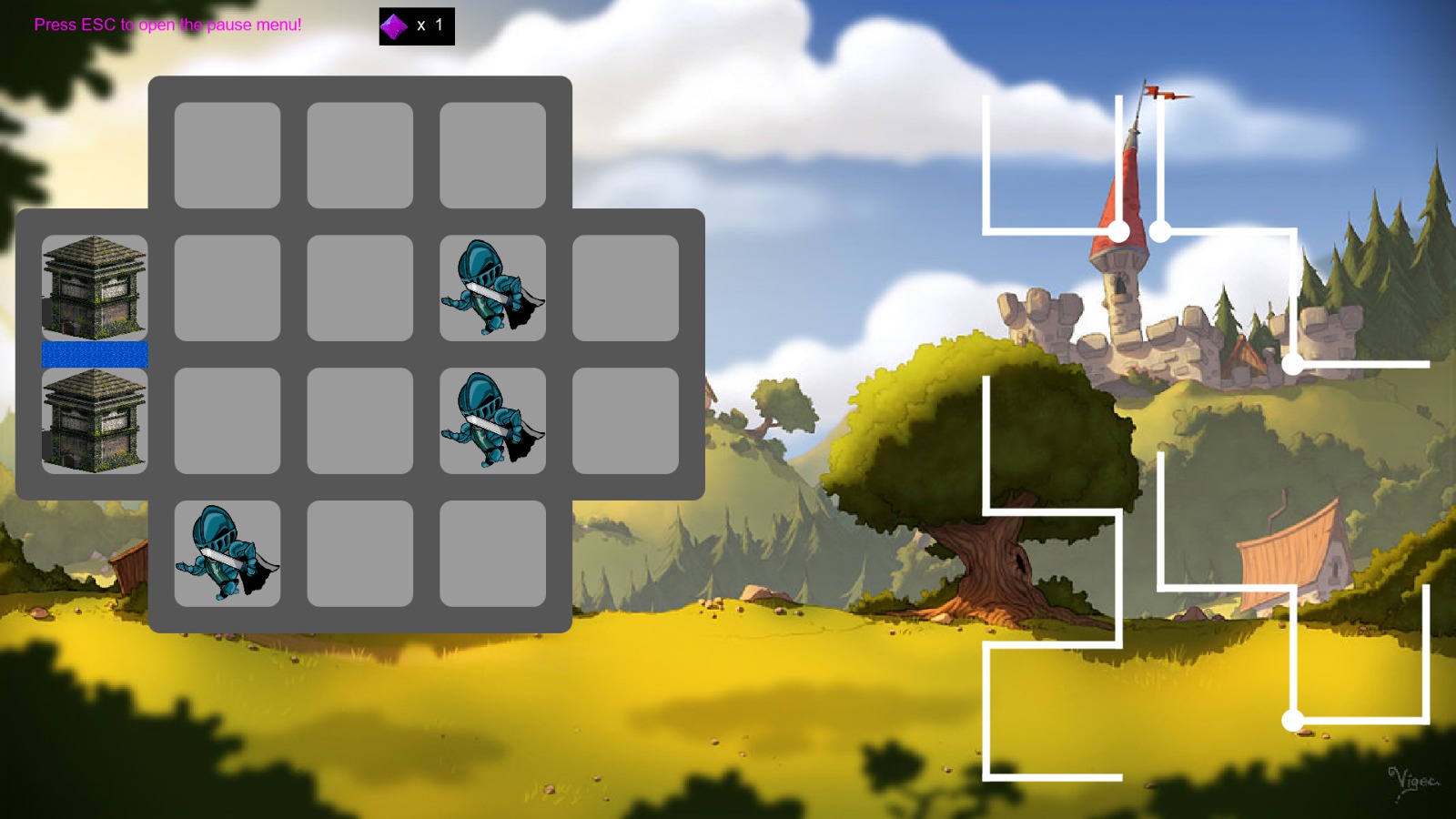
1. Selecting the play option will take the user to the following screen:



* 1. If the user wish to play *Classic Mode*, he should click on the *Classic Mode* button. He will then be taken to the following screen.



* + 1. When the user selects a level, he will be presented with a game board. Here the user will place the Walls and try to complete the challenge.



* 1. If the user wishes to create his own level, he should select the button labeled *Create a Level*. He will then be presented with the following screen where he can design his own puzzle.



# Work Distribution

Our group’s work distribution is as follows:

* **Salih Altun**
* Writing reports
* Drawing sequence and class diagrams
* Testing the game
* Subsystem decomposition
* Designing GUI of the game
* **Deniz Yüksel**
* Project Manager
* Maintaining meeting logs
* Reviewing all reports before uploading
* Implementation of asset classes
* Designing GUI of the game
* Finding sprites
* Testing the game
* **Dawood Muzammil**
* In-charge of all the reports
* Drawing activity and state diagrams
* Programming and Implementation of entire *MongoDB* database in the game
* Design trade-offs and decisions
* Testing the game
* **Mert Soydinç**
* Implementation of the game
* Worked on the GUI of the game
* Programmed the Classic and Sandbox mode of the game
* First iteration class diagram and subsystem decomposition
* Came up with the algorithm to mathematically check if the puzzle is solved
* Testing the game
* **Faruk Oruç**
* Implementation of the game
* Worked on the GUI of the game
* Programmed the Sandbox mode of the game
* Use case diagram
* Testing the game

# References

* + - 1. **Installing MongoDB on Windows** — <https://www.coursera.org/lecture/introduction-mongodb/installing-mongodb-on-windows-Hadhu>
      2. **MongoDB** — <https://www.mongodb.com/>
      3. **Java Swing** — <https://docs.oracle.com/javase/tutorial/uiswing/index.html>