Bilkent University
Department of Computer Engineering

CS 319 - Object-Oriented Software Engineering

Term Project - Analysis Report

Project Name: Walls & Warriors

Group No: 2B

Group Name: OOPs

Group Members: Ali Babayev
Tunar Mahmudov
Merve Sağyatanlar
Çağla Sözen
Emin Bahadır Tülüce

Table of Contents

1. Introduction

For Object Oriented Software Engineering (CS319) course, as Group 2B - OOPs, we've decided to design and implement a desktop application version of the game *Walls & Warriors*. A game based on "Troy" that challenges the cognitive skills, its first version, designed by Raf Peeters, was not widely played because of its complicated physical structure and redundant features [1]. For the second, widely successful version of the game, the design of the was simplified and its physical structure was improved to be more compact and more user-friendly. As OOPs, we've decided that for a game that faced such obstacles during its development, we could increase the functionality, user-friendliness and make it more entertainable by implementing it as a desktop application, as well as presenting a good design in terms of object oriented programming with its numerous distinct structures.

With already defined physical objects in the board game like Red Knights, Blue Knights, Walls and a High Tower this game, highly motivated us to come up with a good object oriented game design. In addition to the existing structure of the game we decided to implement additional features that will move with the natural flow of the game and utilize its object oriented design.

In our initial design plan, we are considering to include and add the following features:

- Campaign Mode
- Custom Mode
- Sound Options
- Knight Color Options
- How To Play Information
- Standard Challenges
- Wild Challenges
- Challenges with Increasing Difficulties
- Creating Custom Challenges
- Sharing and Playing Custom Challenges

The structure of the game is based on dragging and dropping with mouse movements. The maps will vary according to the challenge type.

For *Campaign Mode* the player will be given a challenge with a High Tower, a number of Red and Blue Knights located on the map and the player will be expected to place the given wall pieces on the map according to the rules of the game. In the beginning, some challenges will be locked in Campaign Mode and they will get

unlocked as the player proceeds on challenges, by this was the player will be motivated to solve all challenges while also increasing his/her ability to solve challenges more gradually.

For *Custom Mode* the player will again be given a similar challenge. But this time, the challenge will either be created by the player himself/herself or be a challenge created by another player who shared the challenge with the playing player. In custom mode, there are no locked challenges, the player is free to play any challenge he/she wants.

For *Challenge Editor*, the player will be given the chance to create his/her own challenge. Here, the player first decides if the challenge will be a *Standard Challenge*, meaning that the player will be given a limited number of resources, or a *Wild Challenge*, in which the player will be given an infinite number of resources to place on the map. Challenges will be allowed to be created only if the challenge is solvable, so the creator will be expected to place the walls for the solution as well as the other components. Challenges will be stored with the name of the creator and an optional short description of the challenge. If the player wants to add the challenge to its system, he/she can simply add the level by a button. If the creator wants to share his/her challenge, he/she will be given a copyable challenge code and will be free to distribute this code with the Walls & Warriors community. The same sharing option will also be available for loaded custom challenges in the *Custom Challenges* screen as well as the *Add Challenge* option in this screen to add the challenge using the challenge code.
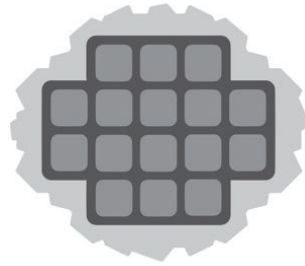
2. Overview

2.1. Game grid

Figure 1

The standard game grid consists of 14 blocks that are aligned as shown in Figure 1. However, the size and shape of the grid may vary in wild challenges (See Section 2.6 for all standard and wild differences). All of the other game elements will be placed on the grid. The knights and high towers stay on the blocks (square hollows) of the grid where the walls can be put on the borderlines of the blocks.

2.2. Knights

There are blue and red knights placed to particular blocks in each challenge. The player is not allowed to change their place or add/remove them from the grid. The knights are represented with circles that are colored accordingly.

2.3. High Tower

A high tower allocates two blocks on the grid and they act like two blue knights. The only difference is the two squares are connected by a solid rampart that does not allow the wall placements in between. (See Section 2.4 for all wall placement rules)

2.4. Walls

There are walls of different length and shapes to be placed by the player on the grid lines in each challenge. Some walls can have bastions on their specific places. The four standard walls are given in Figure 2. There are some rules apply when placing the walls on the grid, as listed below.
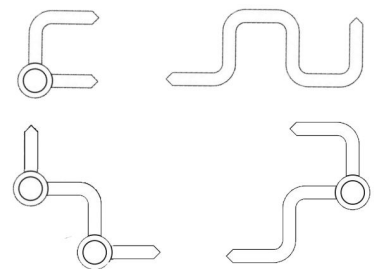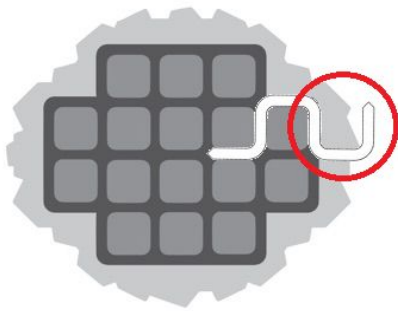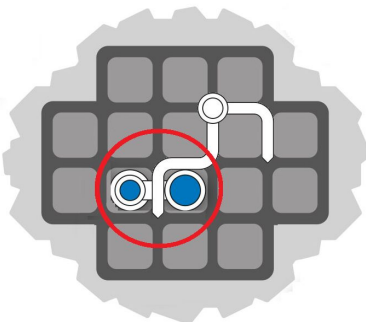

Figure 2: Standard walls

Figure 3

Rule 1
The walls cannot be placed outside of the grid.
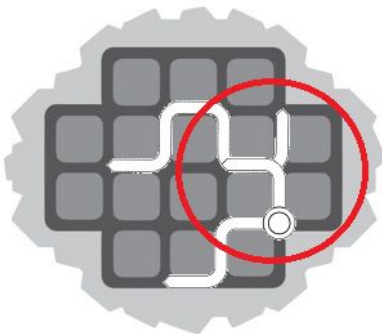

Figure 4

Rule 2
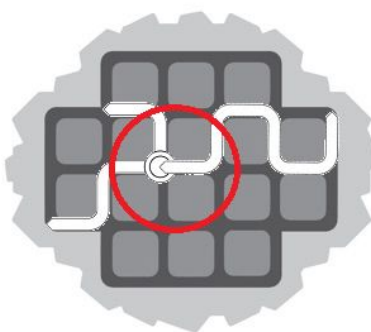The walls can't be placed where the high tower stands.

Figure 5

Rule 3
The walls cannot be placed on top of each other.

Figure 6

Rule 4
The walls cannot end on a bastion of another wall.

2.5. Win Conditions

In order to win, the player should place all the walls to the grid in such a way that the following conditions are all satisfied.

● Blue knights and high towers should be surrounded by the placed walls from all sides.

● Red knights should be kept outside of the walls.

2.6. Challenges

In each challenge, a number of knights and high towers are given on the grid. The player is expected to place walls according to the rules to satisfy the win conditions.

2.6.1 Standard Challenges

The game grid is in a standard shape and size. In each challenge, the same 4 walls are given to the player (See Figure 2). The number of knights and high towers is limited by a number in standard challenges as listed below.

Maximum number of blue knights: 4
Maximum number of red knights: 3
Maximum number of high towers: 1

Other than these, all standard challenges have exactly one solution.

2.6.2. Wild Challenges

There are 10 different types of walls instead of 4. The creator of the challenge can specify which and how many walls to provide for the challenge he/she is creating. Moreover, the number of blue knights, red knights and high towers in the challenge are not limited. There might be more than one solution to wild challenges. The only limitation is that the challenge should be solvable by the given walls.

2.7. Game Modes

There are two different modes of the game.

2.7.1. Campaign Mode

In the campaign mode, the player is given a list standard challenges on a basis of increasing difficulties. Harder challenges are being unlocked as the player progresses on easier challenges. All the challenges in campaign mode are standard.

2.7.2. Custom Mode

In custom mode, the player can play challenges created by the game community. These challenges can be both standard and wild. There is no concept of a locked challenge for the custom mode.

2.8. Challenge Editor

All players are able to create their own challenges. In challenge editor, the player will pick the grid size and specify the number of red knights, blue knights and high towers. Afterwards, they can share their challenge with the game community to be played in the custom mode.

2.9. Settings

The user is able to modify the settings about the game including the volume of the sound effects and the visual theme pack.

3. Functional Requirements

3.1. Challenges

3.1.1. Campaign Challenges
The player will access this screen from the home screen using the "Campaign Challenges Button". In this mode, the player is given a set of locked challenges that are default on the game. As the player succeeds in an easier challenge, harder ones will be unlocked. This way, the level of difficulty of challenges will increase gradually.

3.1.2. Custom Challenges
The player will access this screen from the home screen using the "Custom Challenges Button". The player is given a set of challenges that are either created by himself/herself or the other players. Challenges created by other players can be imported using the code given for each challenge during challenge export. In this mode, all the challenges are unlocked and the player is free to choose any. The difficulty levels of challenges vary randomly within the challenges list.

3.2. Playing a Challenge

3.2.1. Resetting Progress
In the game screen, the player is provided with a "Reset Button" that enables the player to easily start over in the challenge. This feature is added to increase the user-friendliness of the game.

3.2.2. Helping the Player for a Challenge
In the game screen, if the player clicks on the "Check Button" without successfully completing the challenge, he/she will be provided with a visual and textual hint that shows and describes the missing piece or the fault.

3.3. Challenge Editor
The player will access the Challenge Editor from the home screen using the "Challenge Editor Button". Challenge Editor allows the players to create their own challenges. There are two types of user-created challenges supported in the challenge editor, the player will be directed to each of the following screens according to his/her choice of challenge type using the buttons on the pop-up window:

3.3.1. Standard Challenge Editor
The player will access this screen from the pop-up screen on the home screen using the "Standard Challenges Button". Standard Challenge Editor allows the player to select a desirable grid size within a range and use a limited number of game objects

such as red/blue knights, walls and high towers. The player is expected to grab and drop the objects on the game grid appropriately according to the rules.

3.3.2. Wild Challenge Editor

The player will access this screen from the pop-up screen on the home screen using the "Wild Challenges Button". Wild Challenge Editor allows the player to select a desirable grid size within a range and provides a non-limited number of game objects. The player is expected to grab and drop the objects on the game grid appropriately according to the game rules.

Once the player chooses the challenge and decides upon the number of objects and locates them on the grid, he will be asked to write his/her name and an optional short description of the challenge. The challenge will be accepted if there exists at least one solution, provided by the creator himself/herself by also locating the walls pieces on the grid appropriately. Also, if the challenge does not satisfy the constraints of building a challenge (i.e. There must be at least one knight from each color). If so, a copyable challenge code will be generated and given to the player in order to share this challenge with the other community members. The creator of the challenge will also have the option to add the level directly to his/her custom challenges using the button "Add as level".

3.4. How To Play

The player will access this screen from the home screen using the "How To Play Button". This option presents the rules and tips of the game. Basic information about the following is provided in this screen on different tabs,

Rules: Wall placement rules, win conditions, definitions etc.
Game modes: Definitions of the game modes and their differences
Challenge editor: Description for usage of the challenge editor
Controls: Description of the input controls

3.5. Settings

The player will access this screen from the home screen using the "Settings Button". To make it user-friendly, game settings can be adjusted by the player. On this screen, the player is free to change the following features of the game: music volume, sound volume, color pairs for the knights.

3.6. Credits

The player will access this screen from the home screen using the "Credits Button". Developers of this game and their contact information are listed in this option in addition to the GitHub link of the game.

4. Non-Functional Requirements

4.1. User-Friendly Interface
Since Walls & Warriors is a game originally designed for children,so the user-friendliness of the game is highly significant.
- The actions to place a wall should be easy to perform. The player should not have to perform a pixel-perfect wall placement, instead the wall should align itself to the closest available location.
- The menus and challenge navigation should be easy to follow and the user should be able to feel in which screen they have been and in which screen they want to switch.

4.2. Performance
Performance is very important in Walls & Warriors because it is meant to be a simple and straightforward game.
- The algorithm that will check the win conditions will be performed after each wall placement so this algorithm should run in a reasonable time. That is, it should not affect the experience of the player during the placements.
- The procedure to import and export challenges should be fast as well, since the challenge should be provided to player at an instant.

4.3. Extendibility
We want Walls & Warriors to be extendible since the game structure is open for features and rules.
- The algorithms that run in the game should be able to run with any type of wall shape, with any number of grid components and with any grid size. This way, the new grid components and different sized grids can be easily added to the game.
- The grid components should be extendible as well. A new type of wall should be able to be added to the game easily without changing the other entities.

4.4. Reliability
The player should not be able to import or export any unsolvable challenge. Thus, the algorithms that check the challenge validity should be consistent and reliable in terms of this. Also, the algorithm that checks the win conditions should be giving the expected results for all challenges.
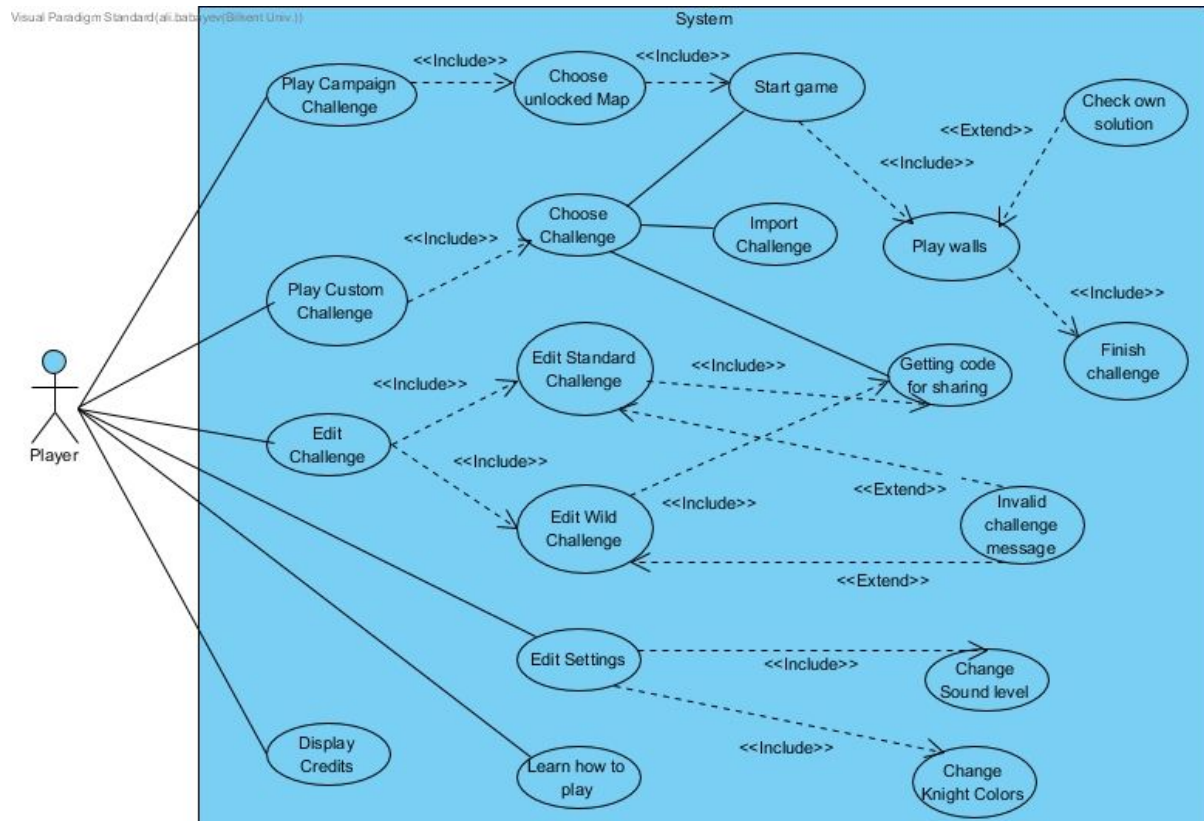
5. System Models

5.1. Use-Case Diagram

Figure 7

Use Case #1

1. ***Use Case***: Play Campaign Challenge

2. ***Participating actor***: Player

3. ***Entry Condition***: Player should be in the main menu

4. ***Exit Condition***: Player accomplishes the challenge

5. ***Success Scenario Event Flows***:

> 1. Player chooses one of the unlocked maps
>
> 2. Player chooses one of the given wall pieces
>
> 3. Player places it into the map
>
> 4. Player repeats the two previous process until placing all of the given walls correctly
>
> 5. Game displays success message
>
> 6. Game returns to choosing the map menu

7. Game unlocks the next locked map

6. *Alternative Event Flows*:

    1. If player places the walls incorrectly

        a. Game gives chance to check mistakes

        b. Game displays the mistaken place

    2. If player wants to return main menu

        a. The system asks for confirmation to exit

        b. Player confirm the choice

        c. The system leaves the game and returns to main menu

Use Case #2

1. *Use Case*: Play Custom Challenge

2. *Participating actor*: Player

3. *Entry Condition*: Player should be in the main menu

4. *Exit Condition*: Player accomplishes the challenge

5. *Success Scenario Event Flows*:

    1. Player chooses one of the challenges

    2. Player chooses one of the given walls

    3. Player places it into the map

    4. Player repeats the two previous steps until placing all of the given walls

correctly

    5. Game displays "success" message

    6. Game returns to the "choosing a map" menu

6. *Alternative Event Flows*:

    1. If player places the walls incorrectly

        a. Game gives chance to check mistakes

        b. Game displays the mistaken place

    2. If player wants to return main menu

        a. The system asks for confirmation to exit

        b. Player confirm the choice

        c. The system leaves the game and returns to main menu

Use Case #3

1. Use Case: Edit Challenge

2. Participating actor: Player

3. Entry Conditions:

    1. Player should be in the main menu

    2. Player should choose to edit either "Standard" or "Wild" challenge

4. Exit Condition: Player gets code for sharing his new challenge

5. Success Scenario Event Flows:

    1. Player chooses one of the challenge categories

    2. Player determines the number of red and blue knights, and high towers

    3. Player chooses a wall from the given available wall pieces

    4. Player places the chosen wall into the map

    5. Player repeats the previous 2 steps until completing the new map
successfully

    6. Player fills the description and author's name part and chooses level of the
new map.

    7.Player exports the new created challenge

    8. The system creates and displays a unique code to share

6. Alternative Event Flows:

    1. If player placed the walls incorrectly

        a. Game displays the "invalid" message

        b. Game returns to edit menu for player to correct mistakes

    2. If player wants to return main menu

        a. System shows "Exit" menu

        b. Player selects "Exit"

        c. The system asks for confirmation

        d. Player confirm the choice

        e. The system leaves the current menu and returns to main menu

Use Case #4

1. Use Case: Edit Settings

2. Participating actor: Player

*3. **Entry Condition:*** Player should be in the main menu

*4. **Exit Condition:*** The game applies the new setting changes

*5. **Success Scenario Event Flows:***

    1. The system display settings

    2. Player changes the level of the sounds

    3. Player changes the level of the music

    4. Player chooses knight colors

    5. Player chooses to return main menu

    6. The system asks confirmation for whether saving the changes

    7. The system saves and returns to main menu

Use Case #5

*1. **Use Case:*** How To Play

*2. **Participating actor:*** Player

*3. **Entry Conditions:*** Player should be in the main menu

*4. **Exit Condition:*** The system returns to the main menu

*5. **Success Scenario Event Flows:***

    1. The system displays rules

    2. Player reads the tabs

    3. Player chooses to return the main menu

    4. The system asks for confirmation

    5. Player confirm the choice

    6. The system returns to main menu

Use Case #6

*1. **Use Case:*** Credits

*2. **Participating actor:*** Player

*3. **Entry Conditions:*** Player should be in the main menu

*4. **Exit Condition:*** The system returns to the main menu

*5. **Success Scenario Event Flows:***

    1. The system displays credits

    2. Player reads the given information

3. Player chooses to return the main menu

4. The system returns to main menu

6. *Alternative Event Flows*:

    1. If player chooses to visit our GitHub page

        a. The system opens the web browser

        b. The system visits to the our GitHub page's URL.
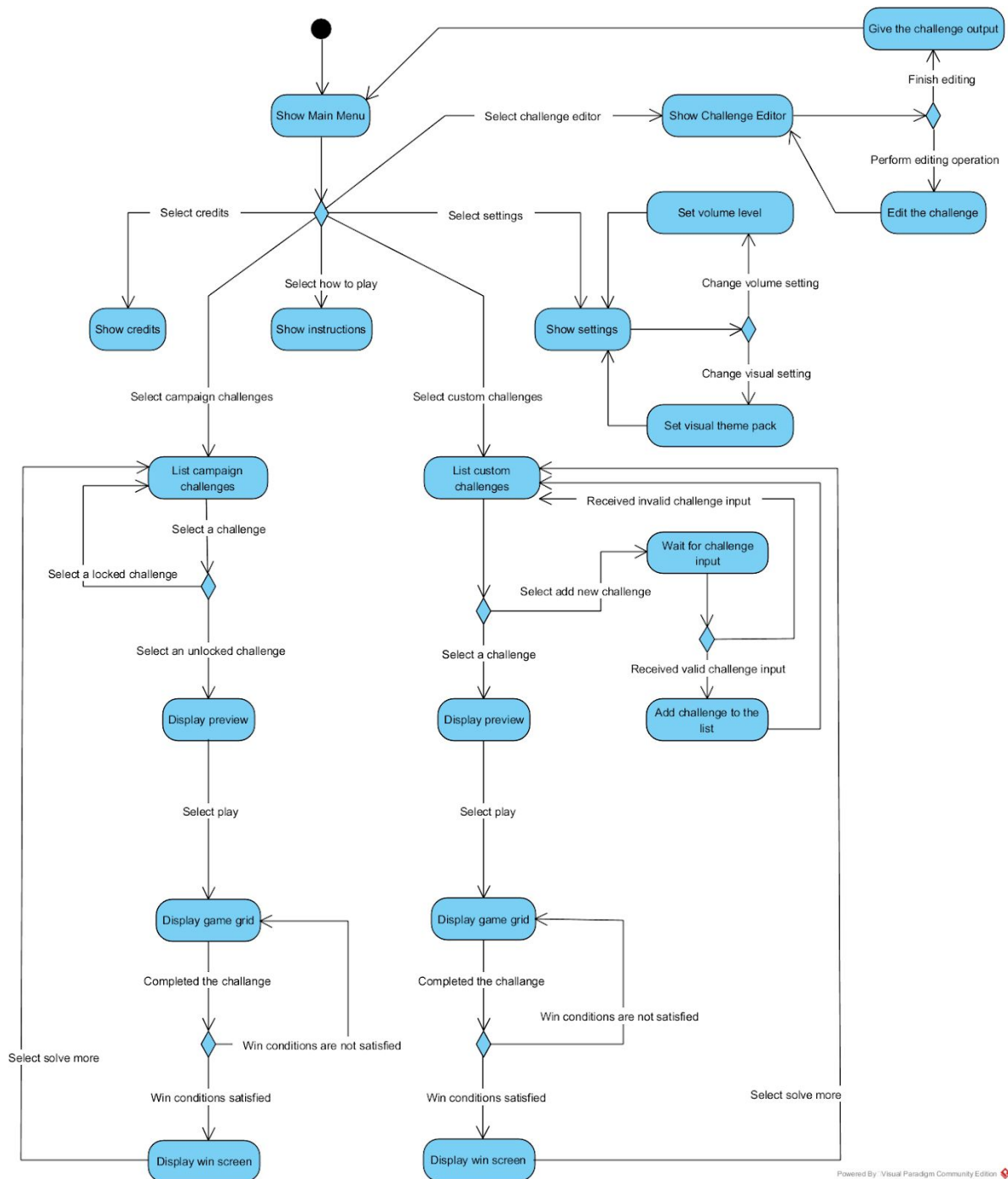
5.2. Dynamic Models
5.2.1. Activity Diagram

Figure 8

The main activities of the game are shown in Figure 8. This activity diagram excludes the actions of the buttons which provides going back to the previous screen or home screen, for simplicity. In the main menu, the player is able to play campaign or custom challenges and create a level. These selections take the player to further screens like the level selection and challenge design. In each screen, proper warnings are displayed with invalid inputs to provide a reliable experience.

5.2.2. Sequence Diagrams

5.2.2.1. Start Playing a Campaign Challenge
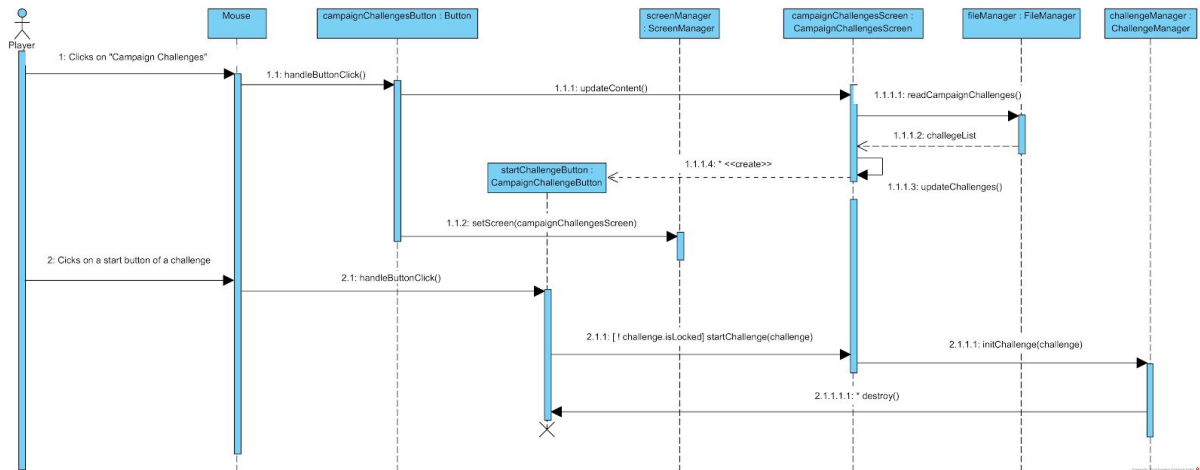
Figure 9

When the player wants to play a campaign challenge, he clicks on "Campaign Challenges". The list of campaign challenges is displayed to the player. If the player clicks on a locked challenge, selection won't open. If the player clicks on an unlocked challenge the game screen will be displayed with the chosen challenge's configuration.

5.2.2.2. Start Playing a Custom Challenge
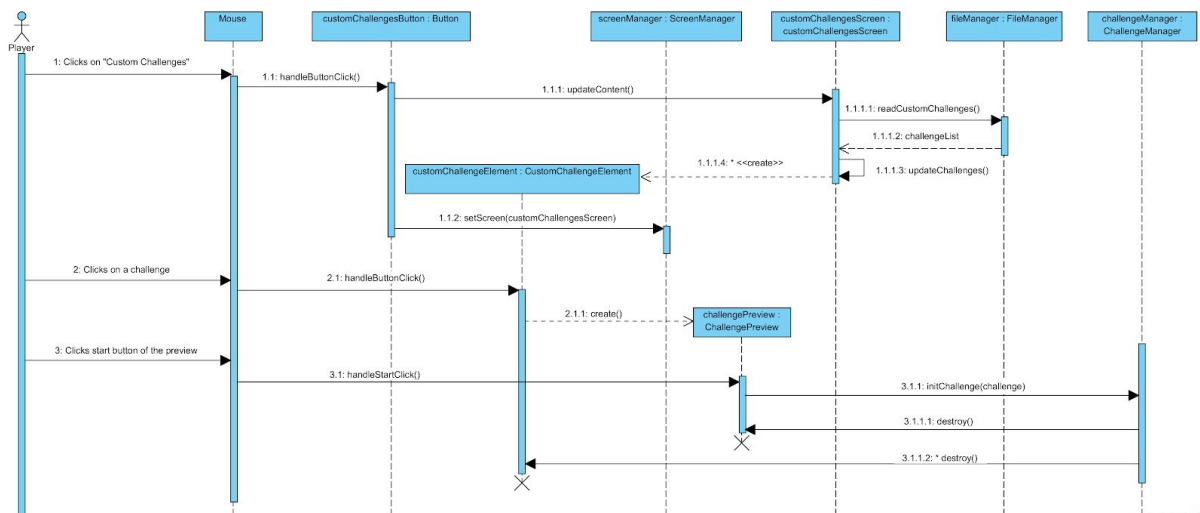
Figure 10

When the player wants to play a custom challenge, he clicks on the "Custom Challenges". The list of previously imported custom challenges is displayed to the player. If the player clicks on a challenge, the preview and information about the challenge will be shown. The player can click on "Play" to switch to the game screen with the selection, or there is also an option to get the challenge code to share the selected challenge.

5.2.2.3. Play a Challenge
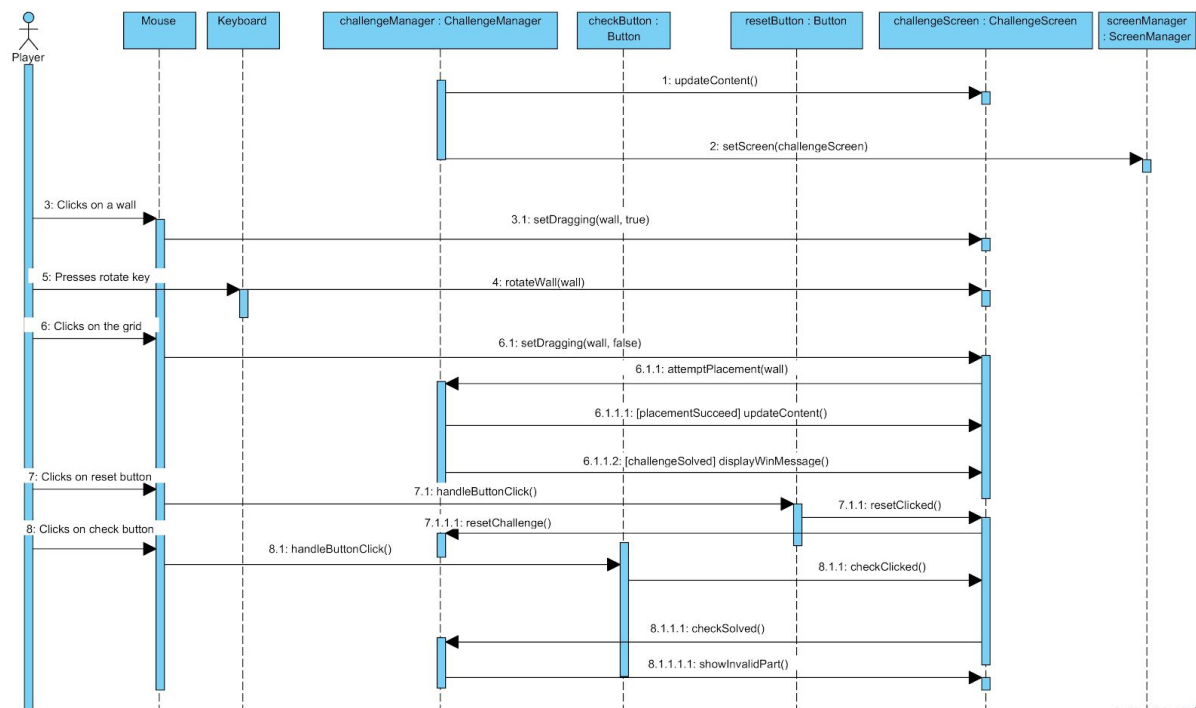
Figure 11

When the game screen is displayed, the user will see the configurations of placed elements (blue knights, red knights and high towers) along with the walls that are available for placing. The user drags a wall to the grid and the wall will be attempted to be placed to the nearest location. If the location satisfies wall placement rules (See Section 2.4) the placement will succeed, otherwise it will fail. After each placement the win conditions are checked and a wining message is displayed if they are satisfied. Also, the player can click on "Check" to see the non-satisfied win condition (i.e. invalid part).

5.2.2.4. Import a Custom Challenge
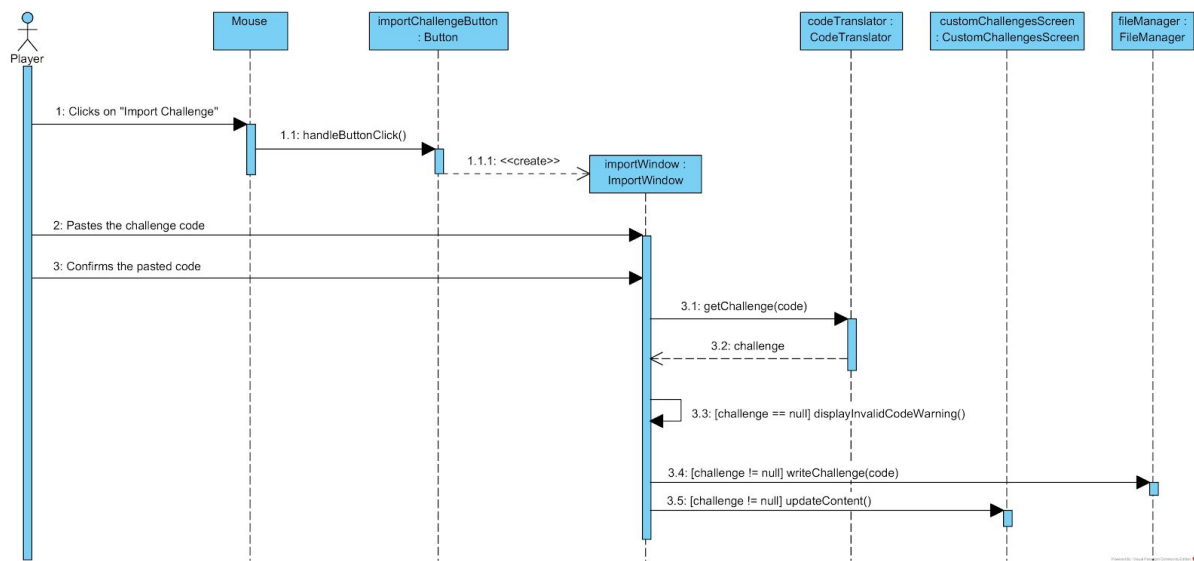
Figure 12

The player has the code of a challenge in their clipboard. They click on the "Import Challenge" button and paste the code. The code is processed and analyzed. If it corresponds to a valid challenge, the challenge is added to the custom challenges list, otherwise, a warning will be shown to the player.

5.2.2.5. Create a Challenge
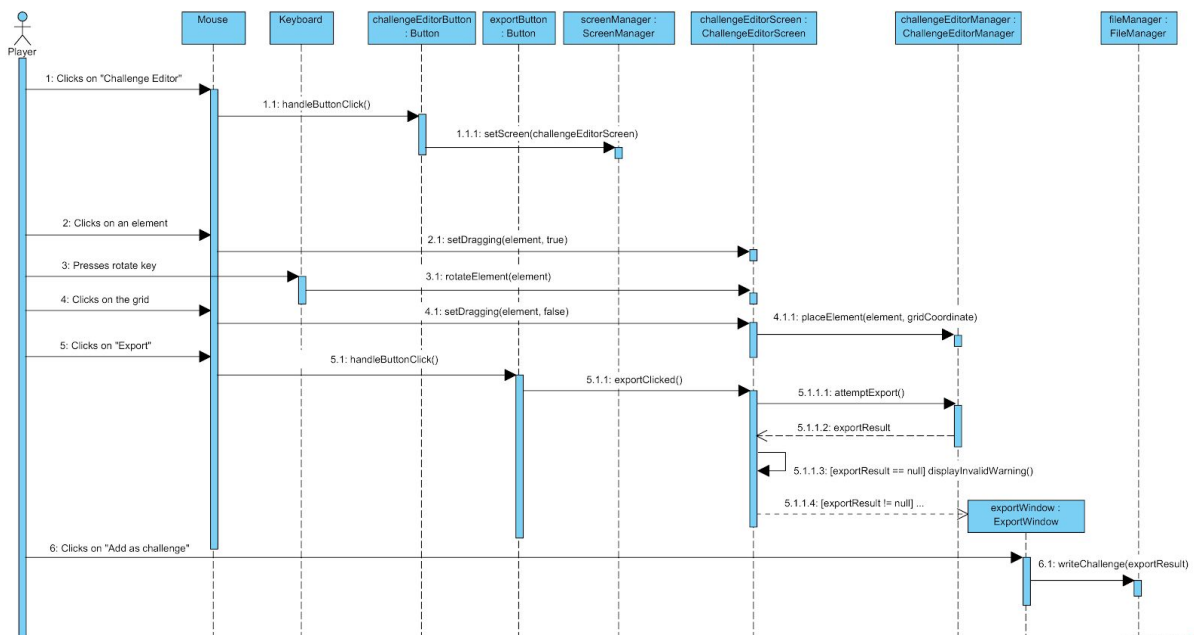
Figure 13

When the player wants to create a new challenge, they click on the "Challenge Editor" button that is in the main menu. The elements (knights, high towers, walls) of the challenges will appear on the editor screen for the player to drag them to the grid. When the player is done creating his challenge, he clicks "Export" button and the code of the created challenge will

be provided to the player. They can copy the code to share or they can directly add the challenge to their imported challenges list.

5.2.2.6. Modify Settings
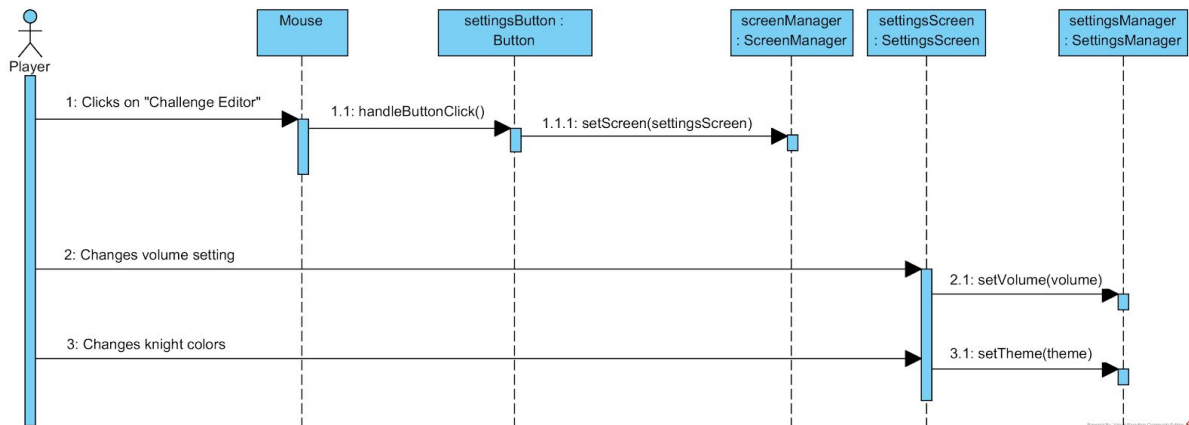
Figure 14

When the player wants to customize the game, he clicks the "Settings" button. Then the user will see the configuration options of sound, music and the theme of the game. So the player can set them as they desire.
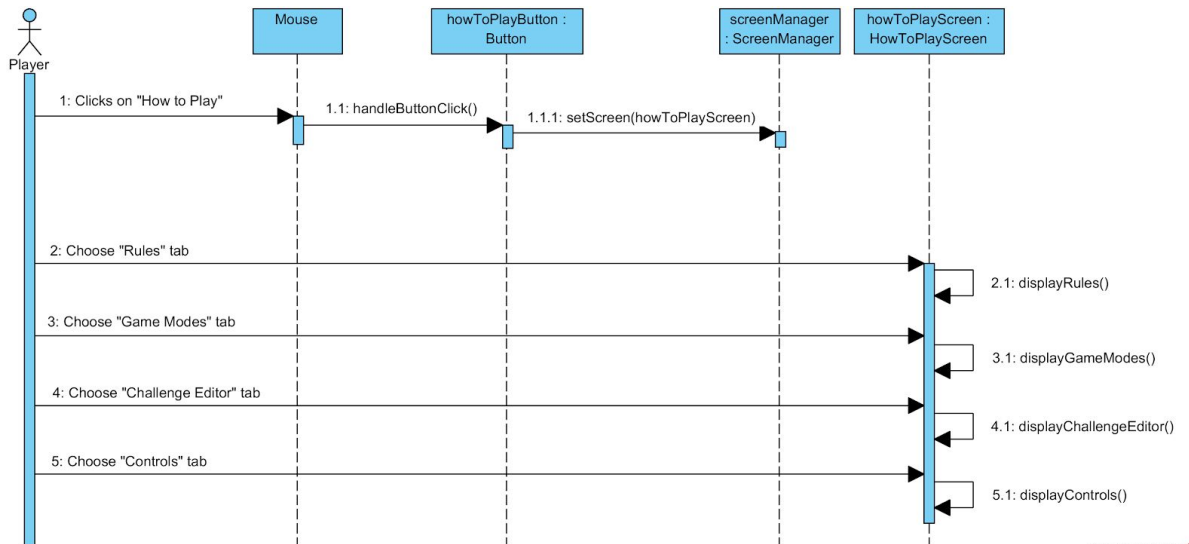
5.2.2.7. Learn How To Play

Figure 15

When the player wants to learn how to play the game, he clicks the "How to Play" button. Instructions are displayed to the player which consists of different categories: Rules, Game Modes, Challenge Editor and Controls.

5.2.3. State Diagrams

5.2.3.1. Element Placement Process

In the challenge editor screen and in the challenge screen, the player will be able to place any element to any valid place he/she desires. The state diagram and state definitions of this placement process are given below.
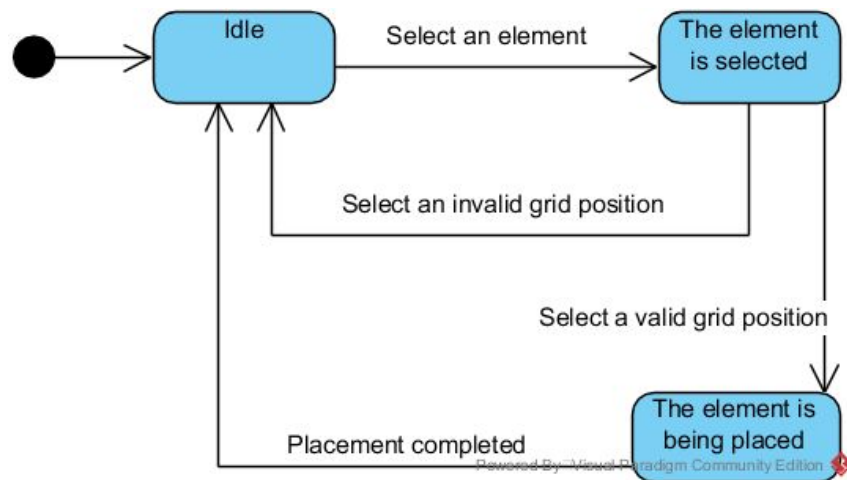
Figure 16

Idle State: The collection of placable elements and the grid is being displayed to the user. There is no element being placed or planned to be placed at this state.

Dragging State: One of the placeable elements is selected by the user and it is planned to be placed in a grid position soon.

Placing State: The user has selected a valid grid position so the program is checking if the position is satisfying the wall placement rules (See Section 2.4 for rules). When the placement is completed (either failed or succeed), the player should be able to make more placements so the state is changed back to Idle.

It should be noted that, the user interactions (element selection and grid position selection) which are defined in this process might differ with the future design choices. They could be controlled through mouse clicks, keyboard presses or screen touches.
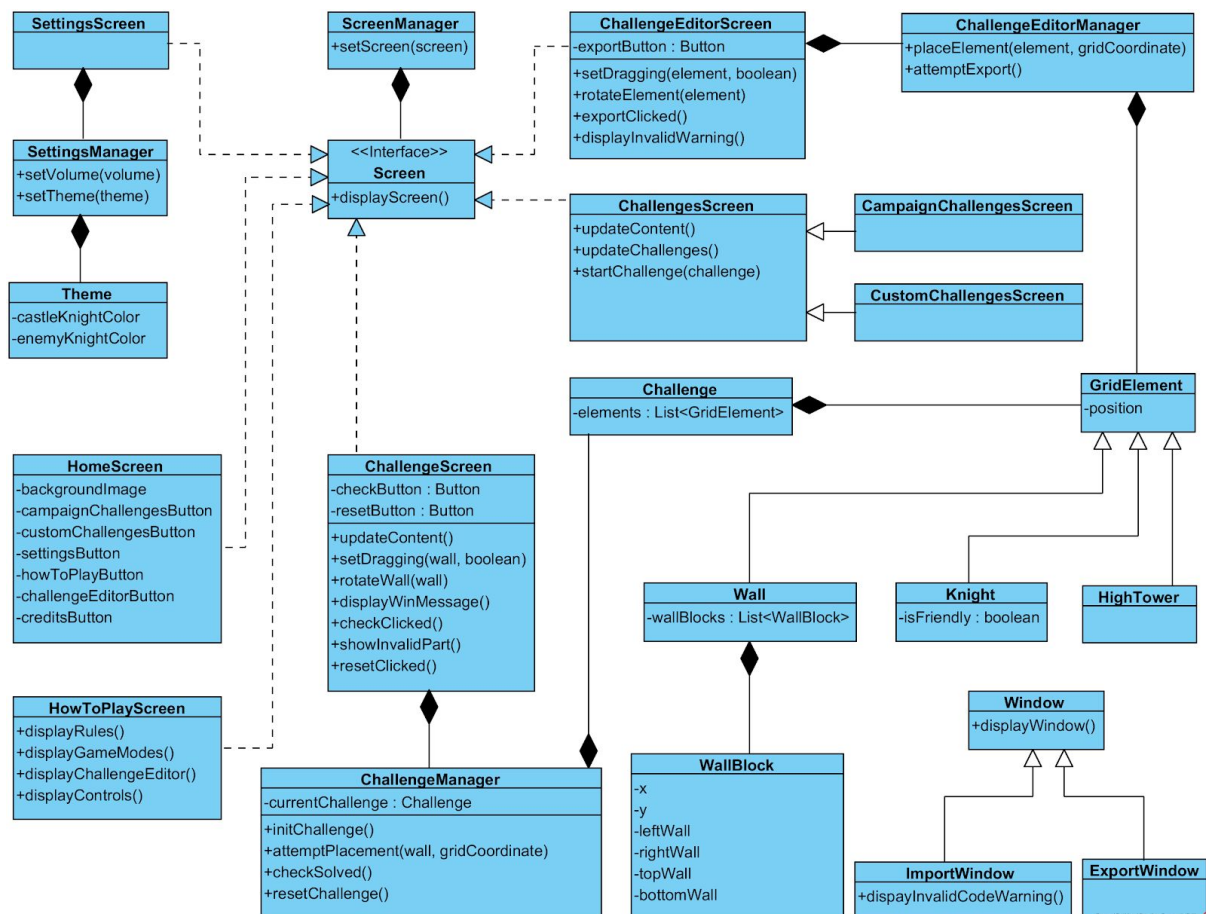
5.3. Object and Class Model

Figure 17: Class diagram for the game objects
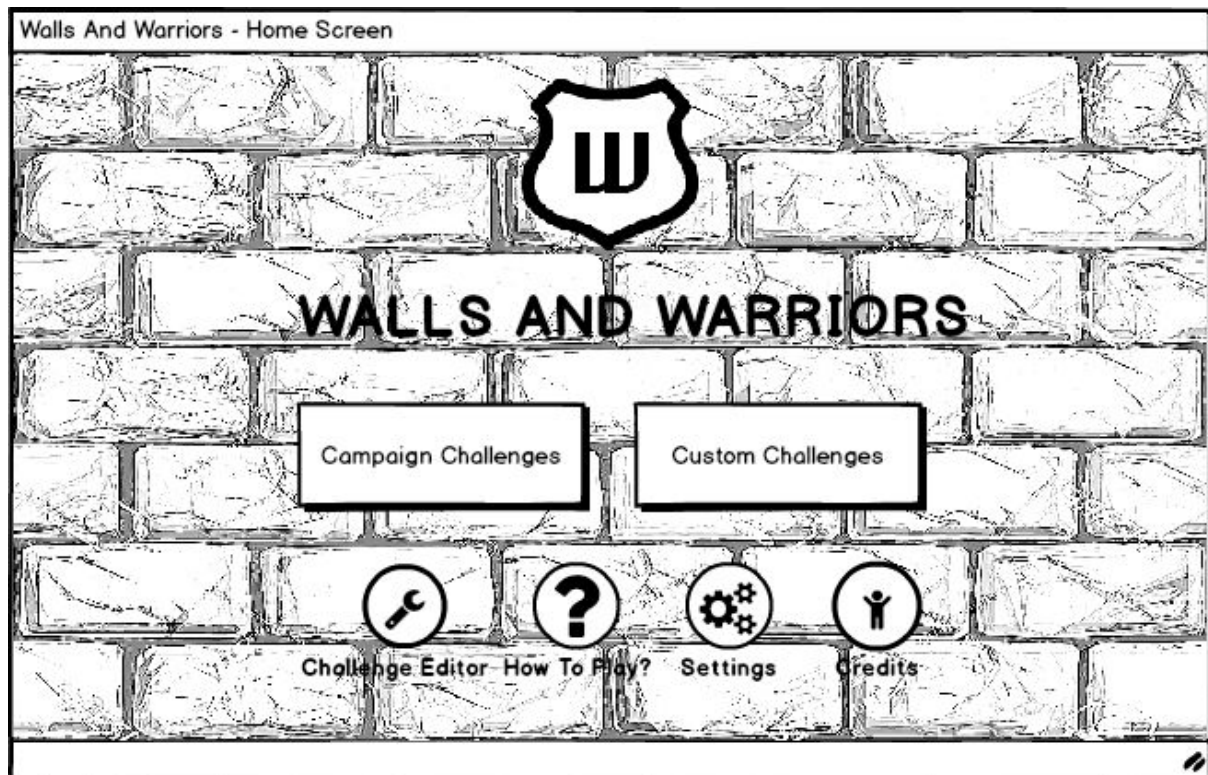
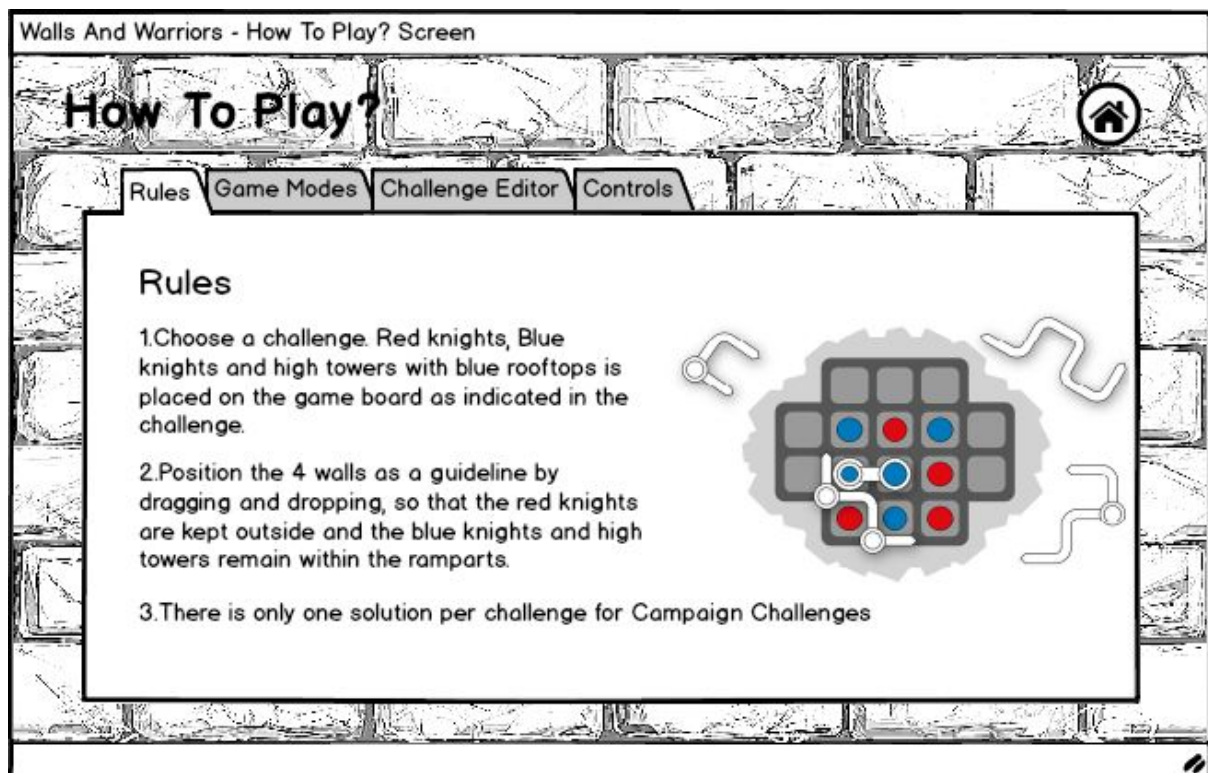## 5.4. User Interface

Figure 18: Home screen

Figure 19: How to play - Rules

How To Play?
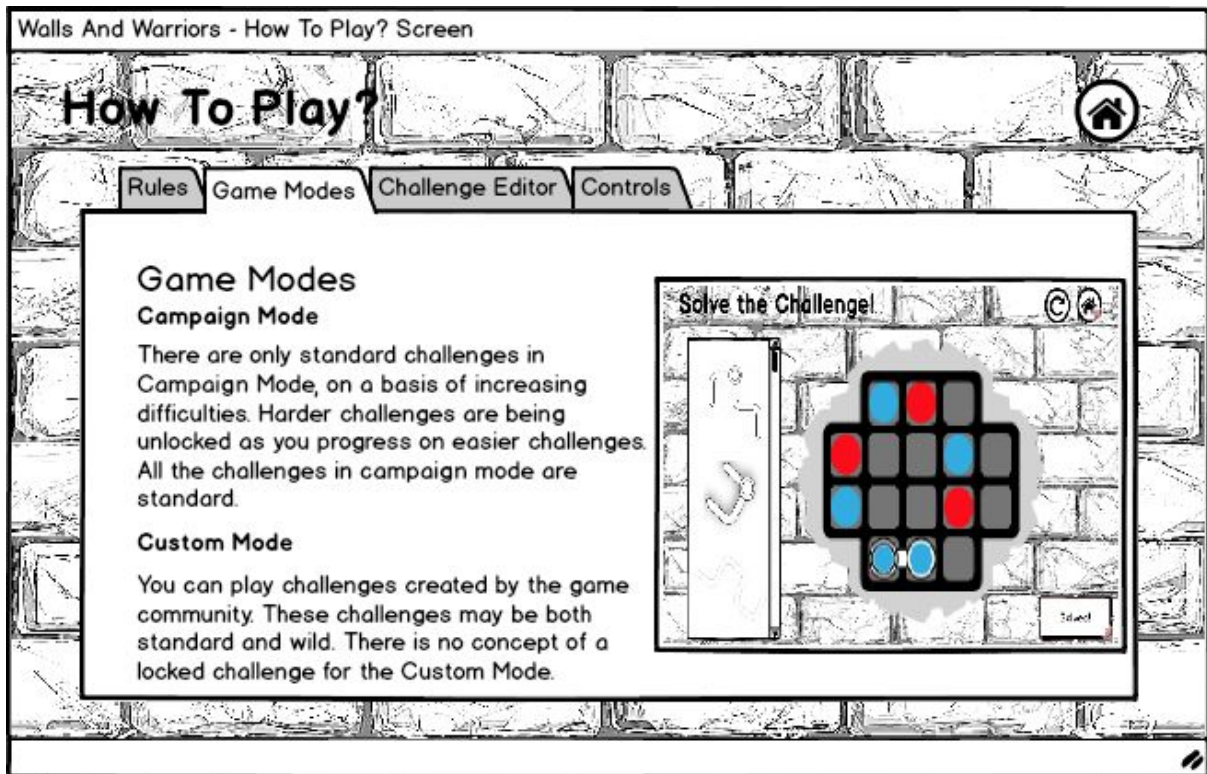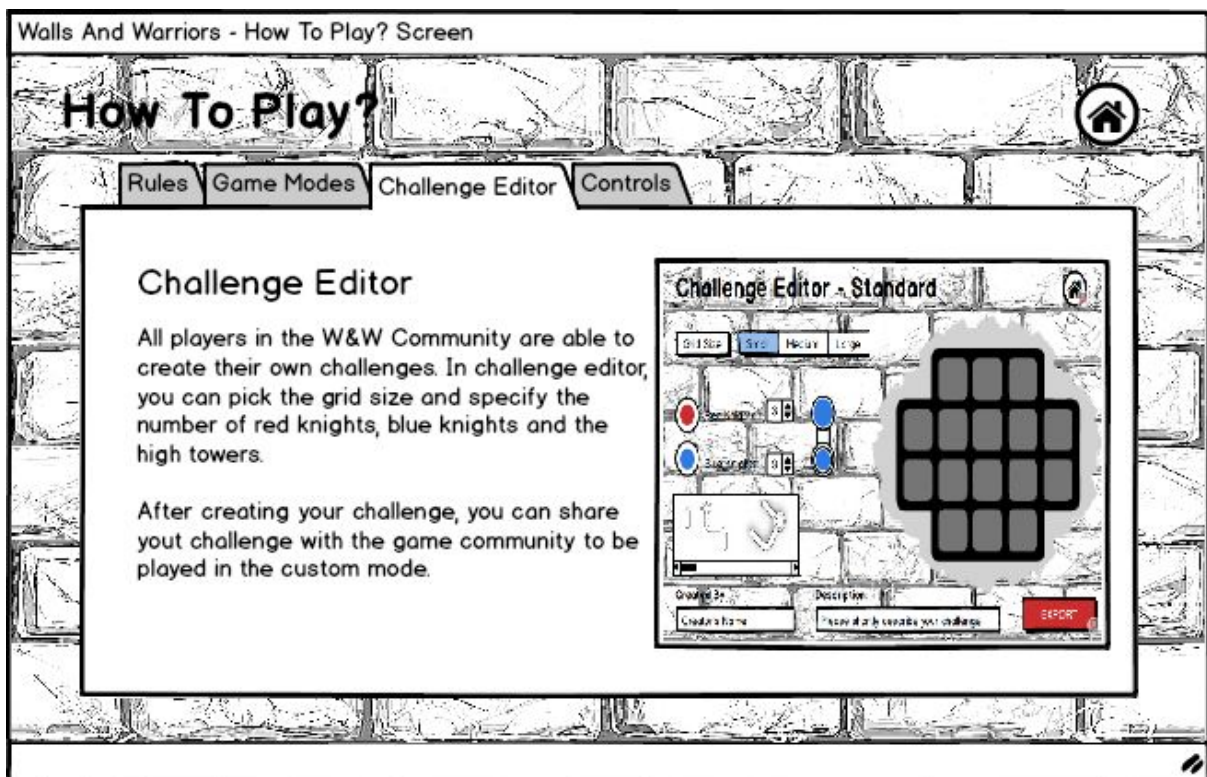
Rules | Game Modes | Challenge Editor | Controls

Game Modes

Campaign Mode

There are only standard challenges in Campaign Mode, on a basis of increasing difficulties. Harder challenges are being unlocked as you progress on easier challenges. All the challenges in campaign mode are standard.

Custom Mode

You can play challenges created by the game community. These challenges may be both standard and wild. There is no concept of a locked challenge for the Custom Mode.

Solve the Challenge!

Figure 20: How to play - Game modes

How To Play?

Rules | Game Modes | Challenge Editor | Controls

Challenge Editor

All players in the W&W Community are able to create their own challenges. In challenge editor, you can pick the grid size and specify the number of red knights, blue knights and the high towers.

After creating your challenge, you can share yout challenge with the game community to be played in the custom mode.

Challenge Editor - Standard

Figure 21: How to play - Challenge Editor

How To Play?

Rules | Game Modes | Challenge Editor | Controls

Controls

For moving the Knights, High Tower and Walls in the Challenge Editor you can grab the object by clicking, drag the mouse to the desired place and drop by clicking once again.

In the Challenge screens, you can only grab&drop the Walls. You can rotate the pieces 90° by grabbing and pressing space bar.
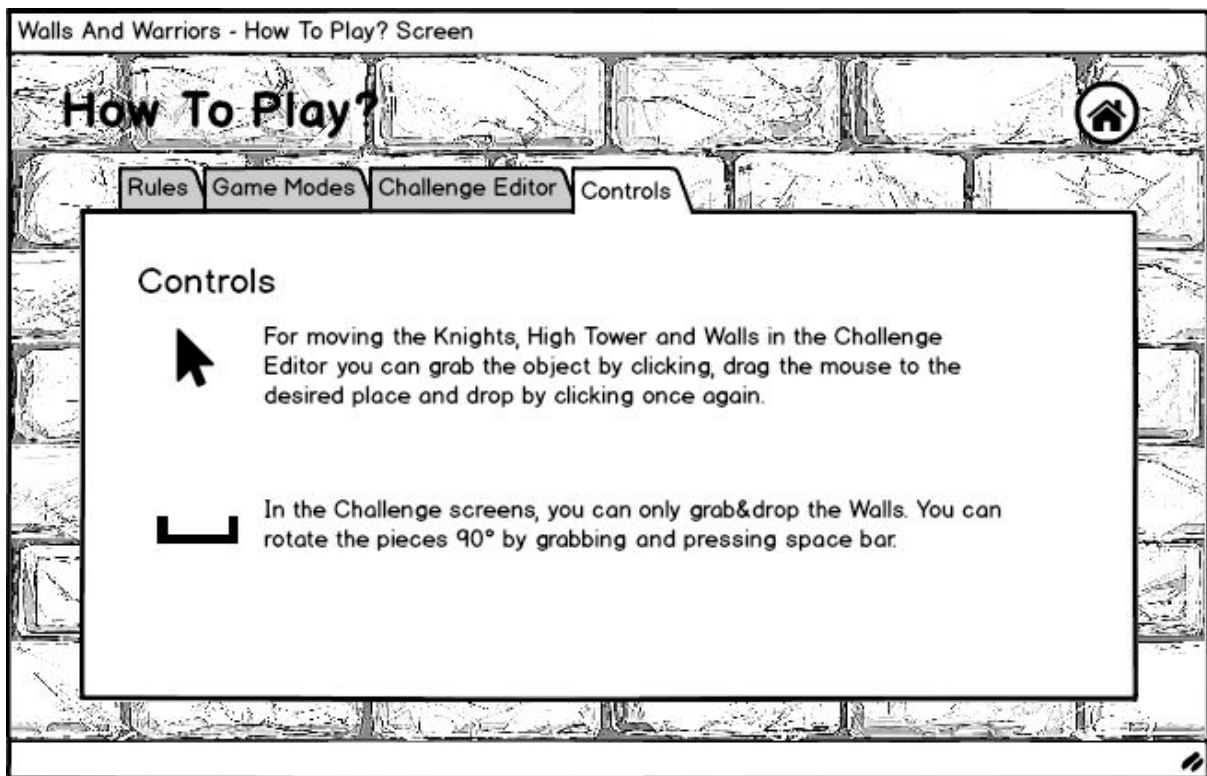
Figure 22: How to play - Controls

Settings

Sounds	◁)) ──○──────────
Music	◁)) ──○──────────
Knight Colors	Blue - Red ▼

Figure 23: Settings

Credits

Ali Babayev: ali.babayev@ug.bilkent.edu.tr

Tunar Mahmudov: tunar.mahmudov@ug.bilkent.edu.tr

Merve Sağyatanlar: merve.sagyatanlar@ug.bilkent.edu.tr

Bahadır Tülüce: bahadir.tuluce@ug.bilkent.edu.tr

Çağla Sözen: cagla.sozen@ug.bilkent.edu.tr

Visit out GitHub for more!

Figure 24: Credits

Campaign Mode - Choose a Challenge
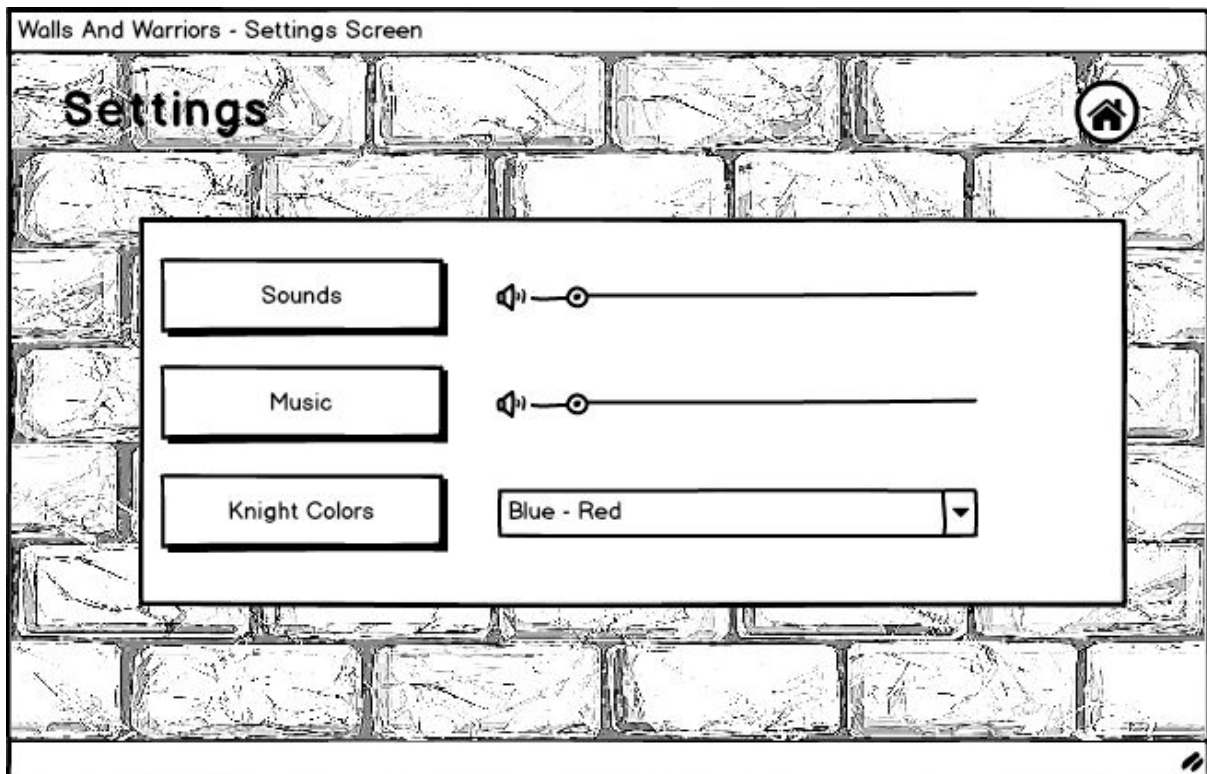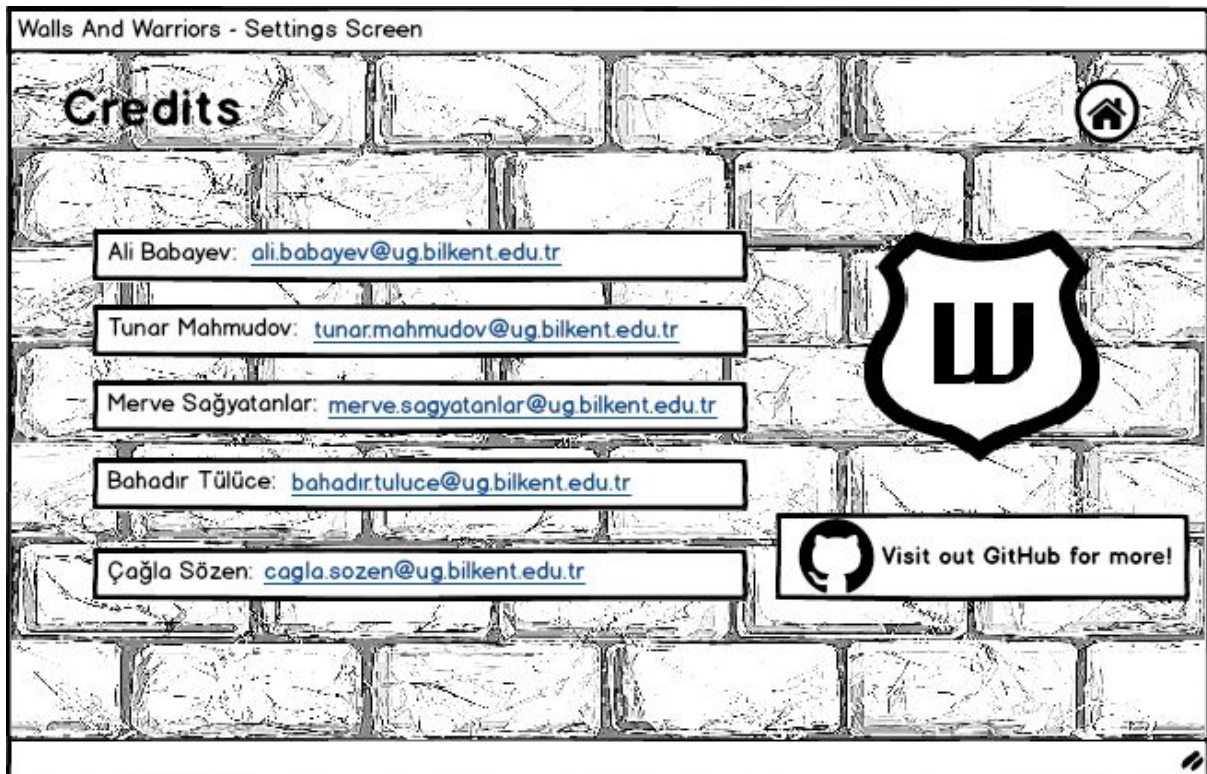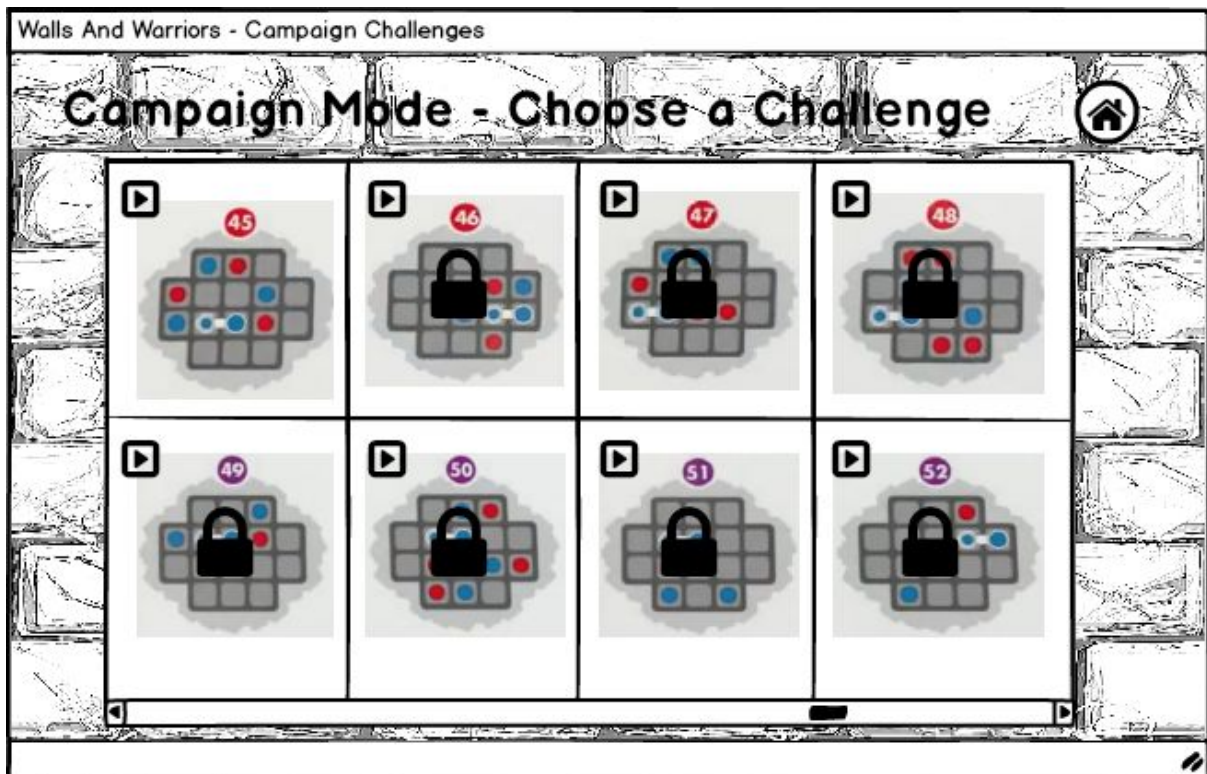
Figure 25: Campaign challenges
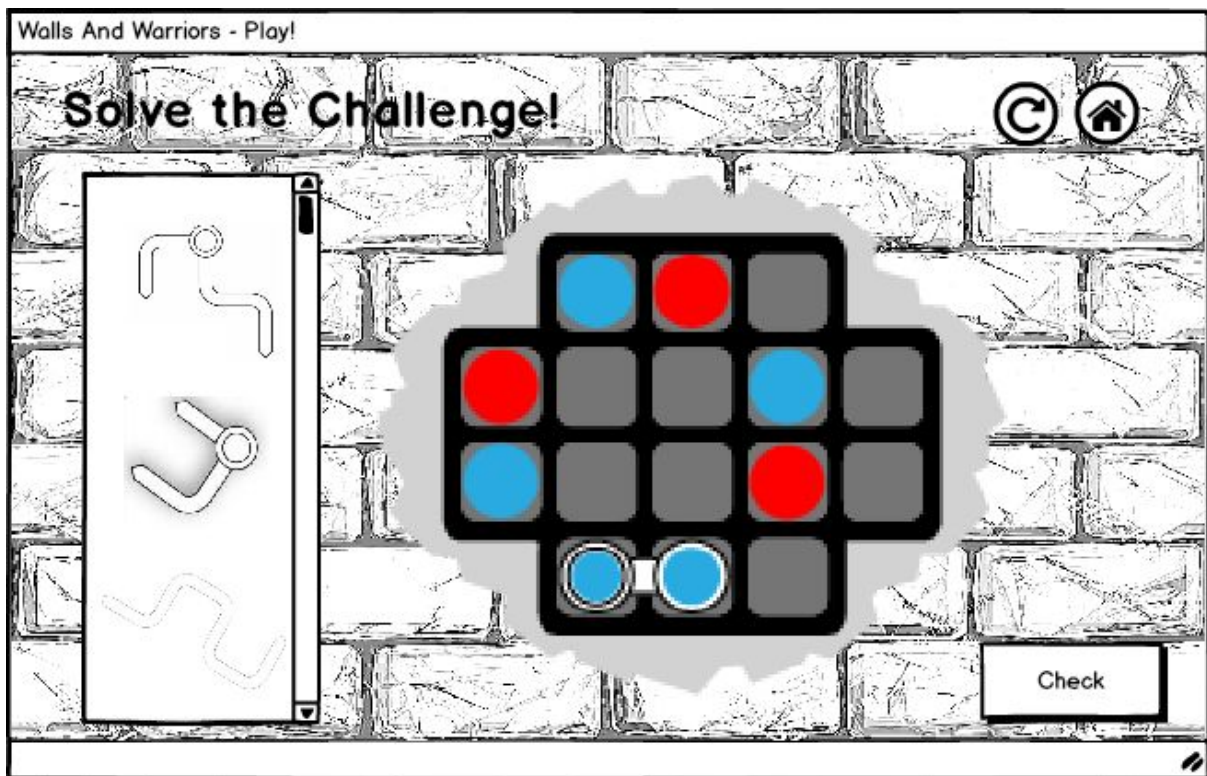
Figure 26: Game screen
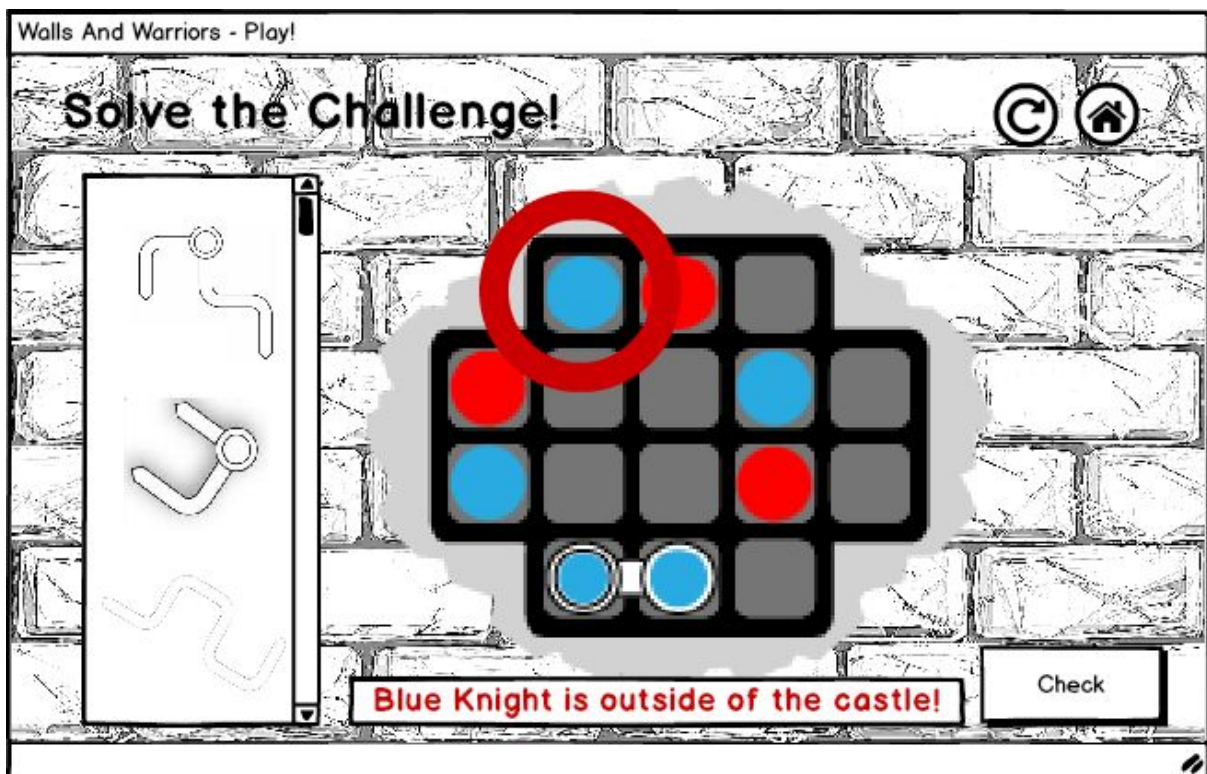
Figure 27: Game screen's warning message after clicking Check button
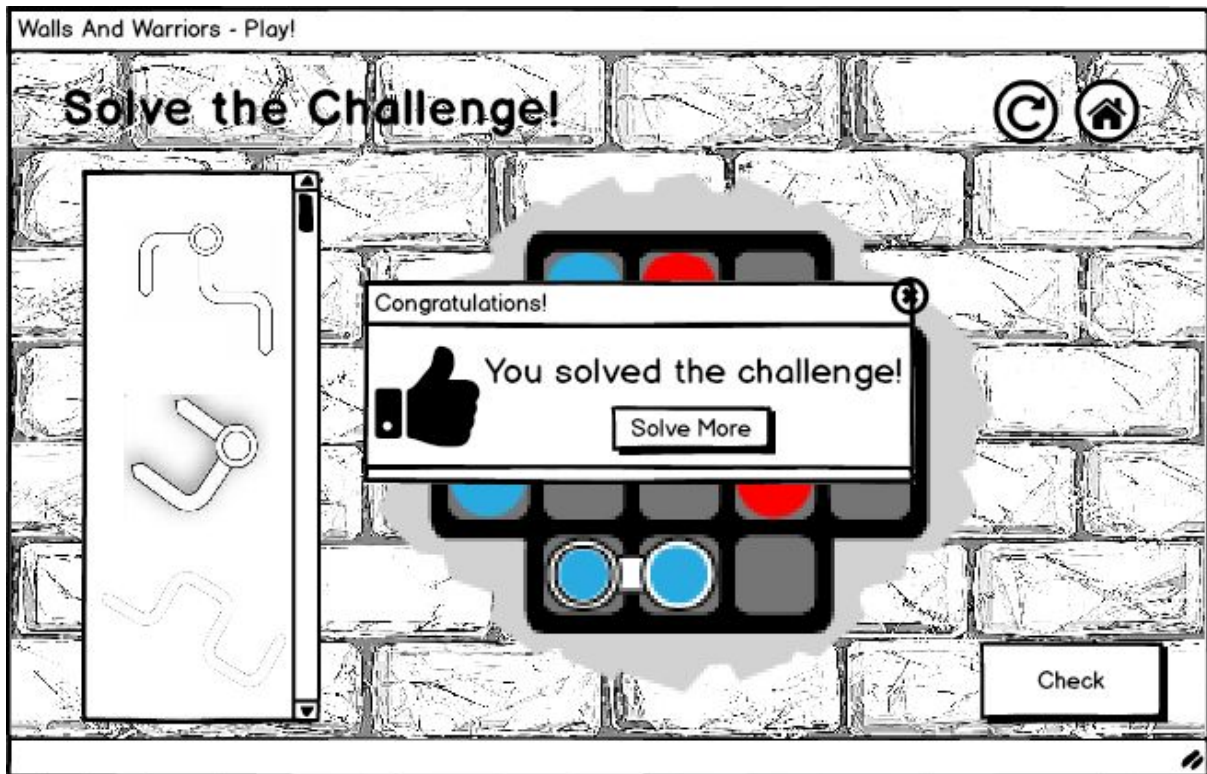
Figure 28: Win screen
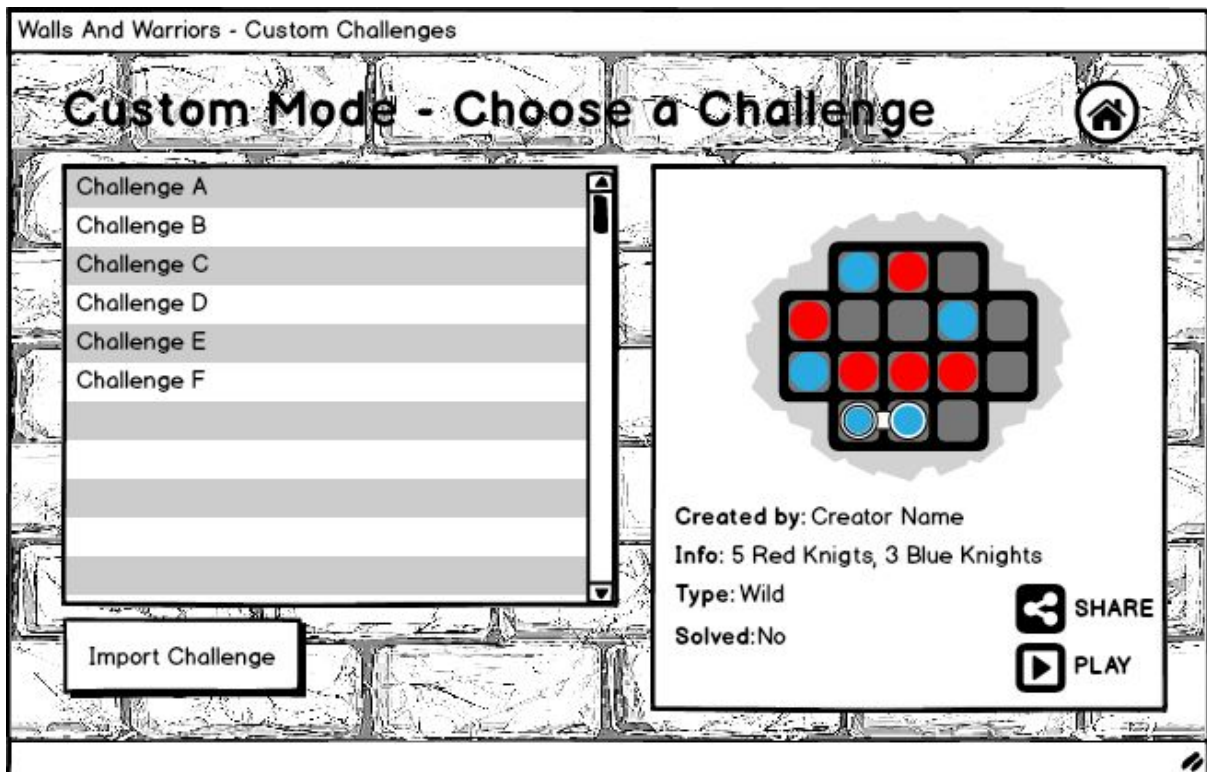
Figure 29: Custom challenges
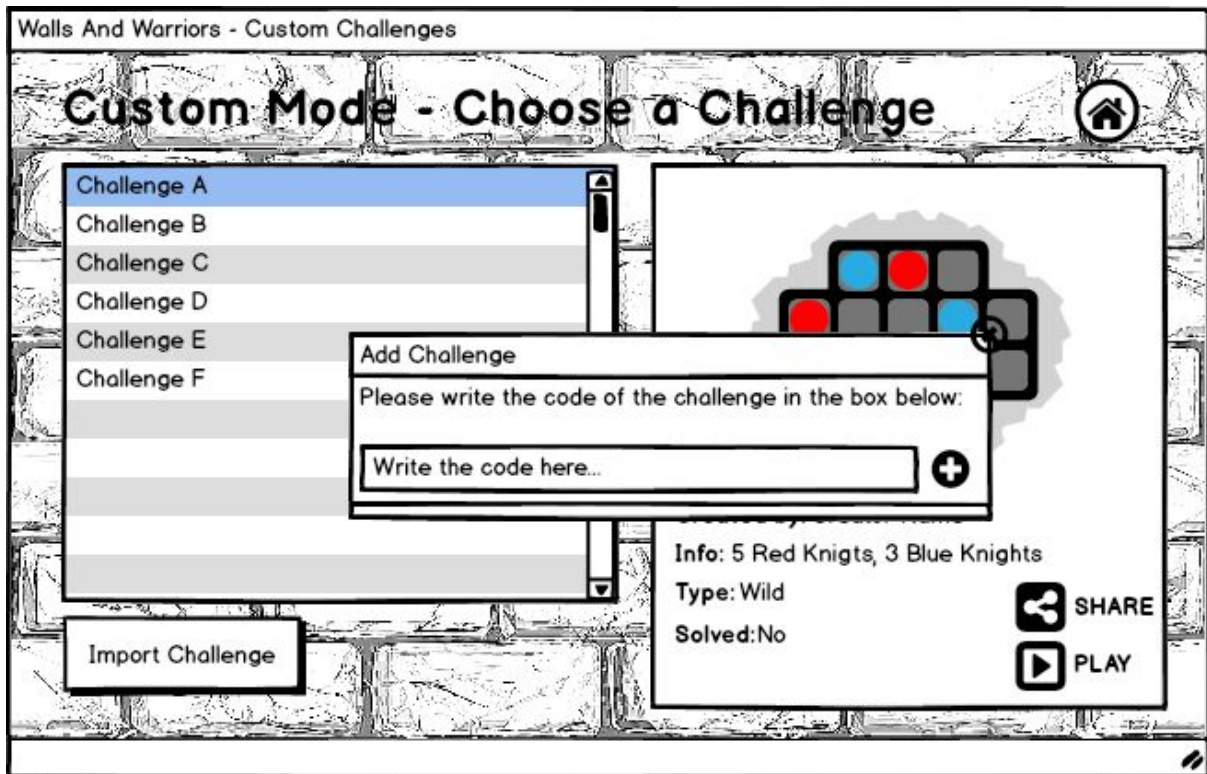
Figure 30: Import a challenge
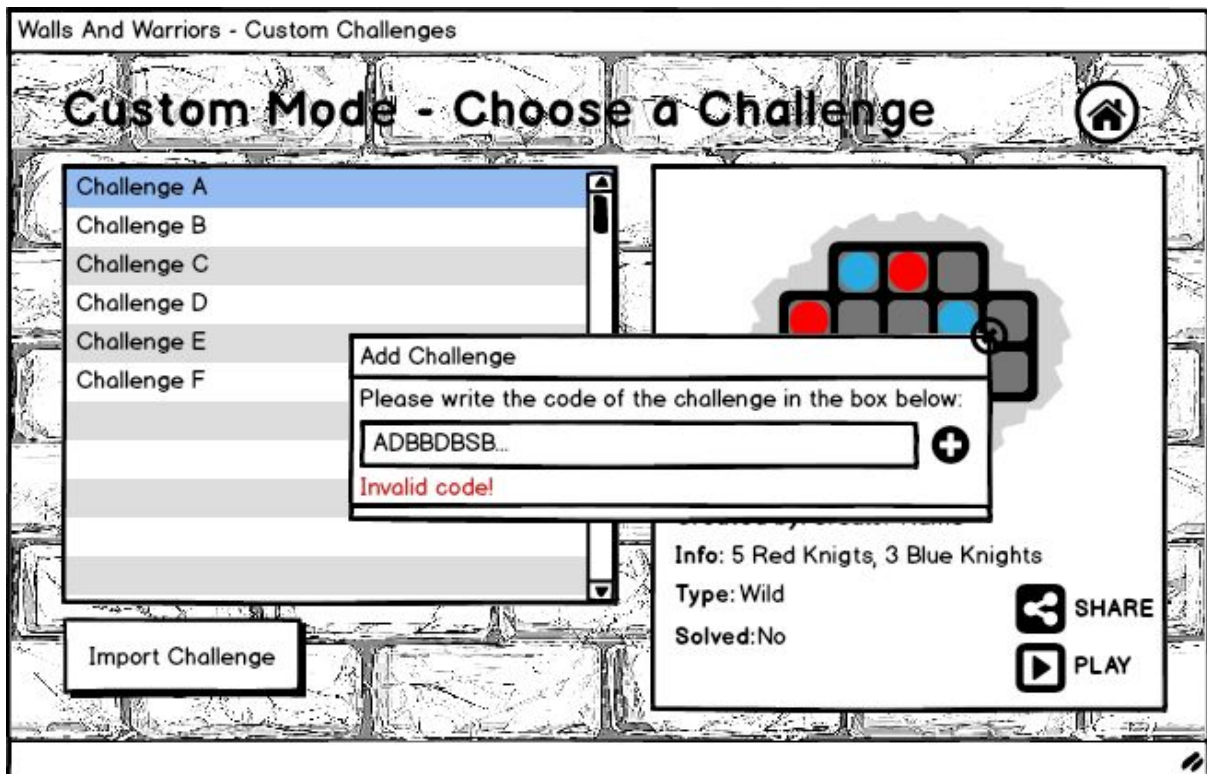
Figure 31: Invalid code warning
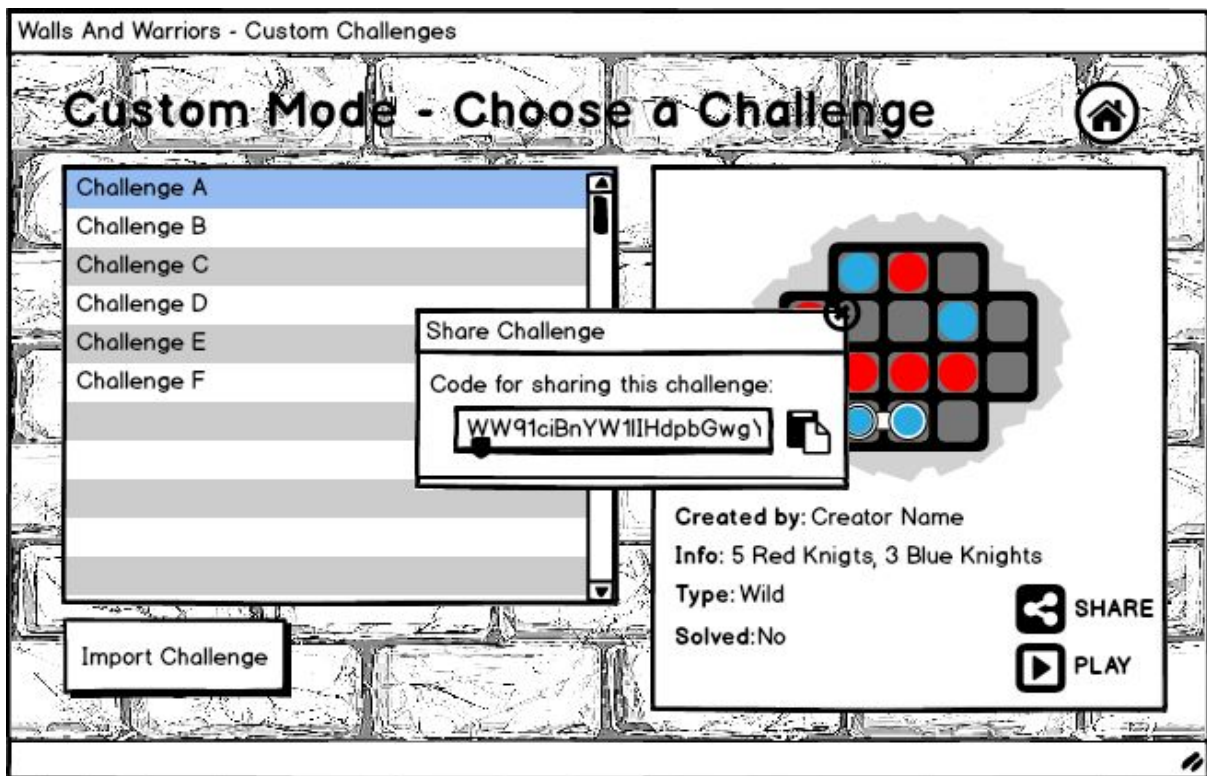
Figure 32: Share challenge
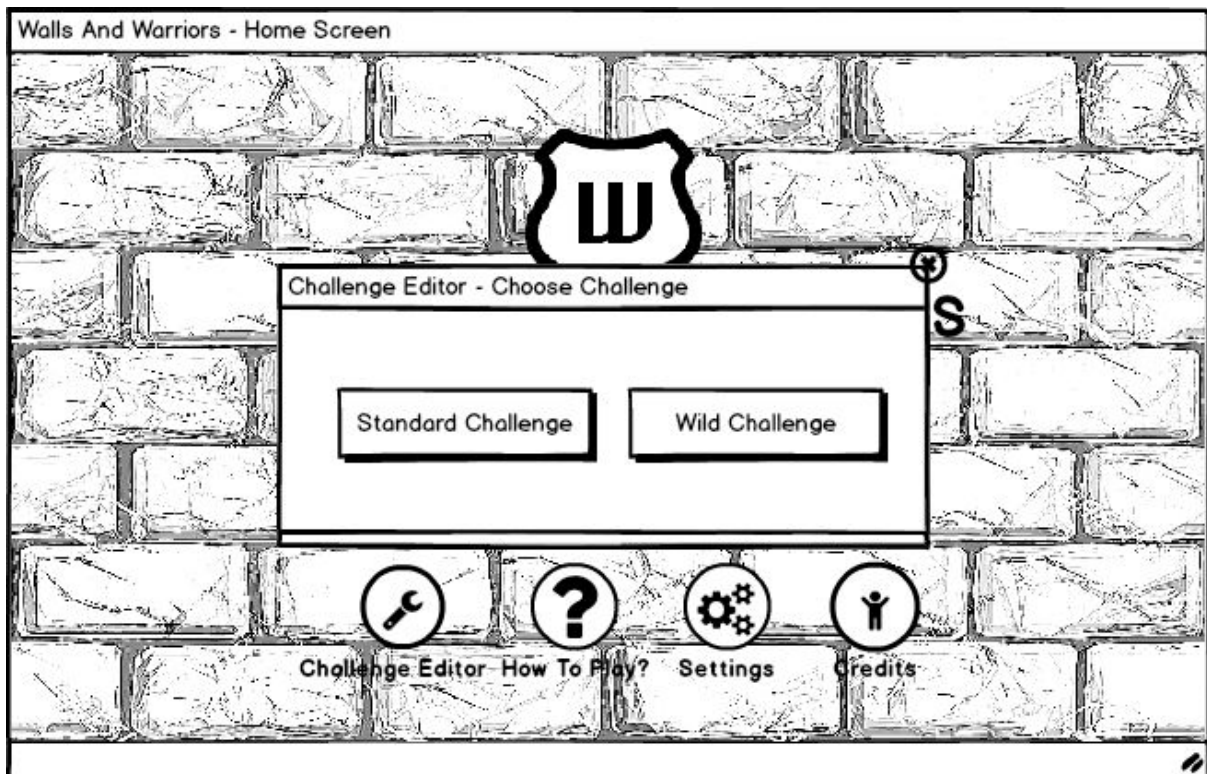
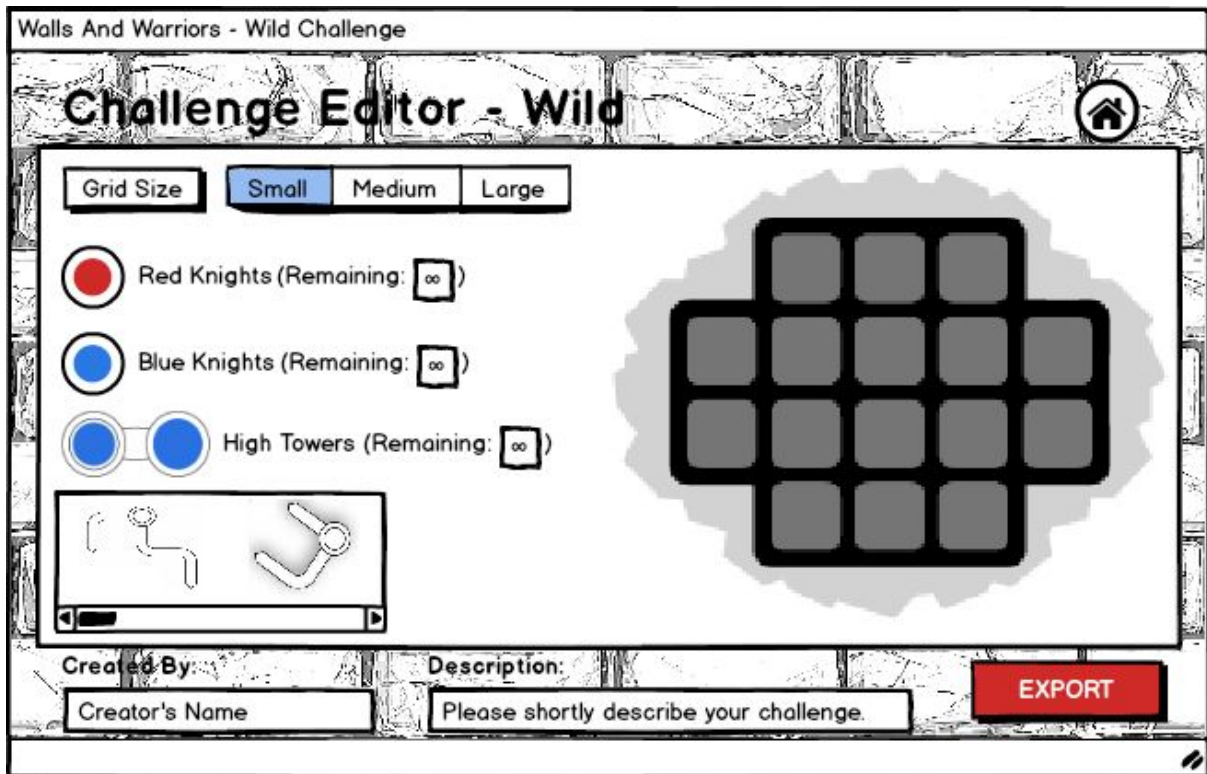Figure 33: Mode selection for challenge editor
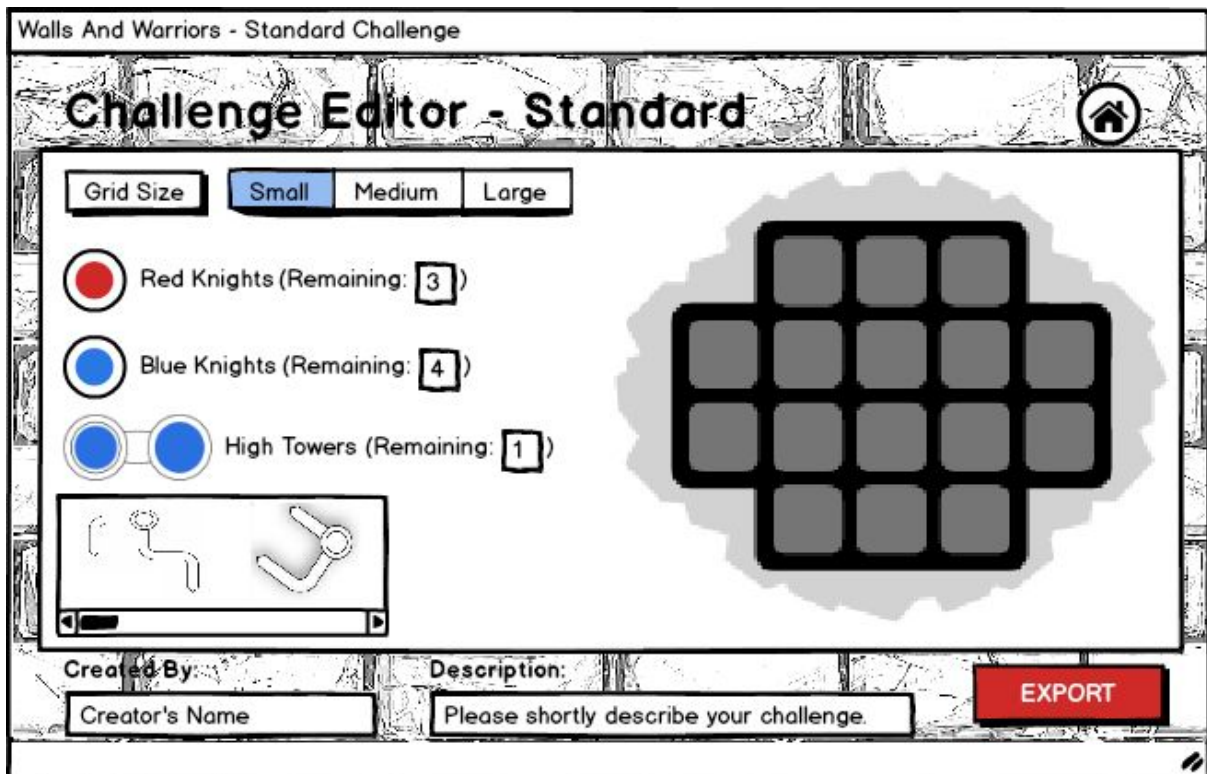
Figure 34: Wild challenge editor
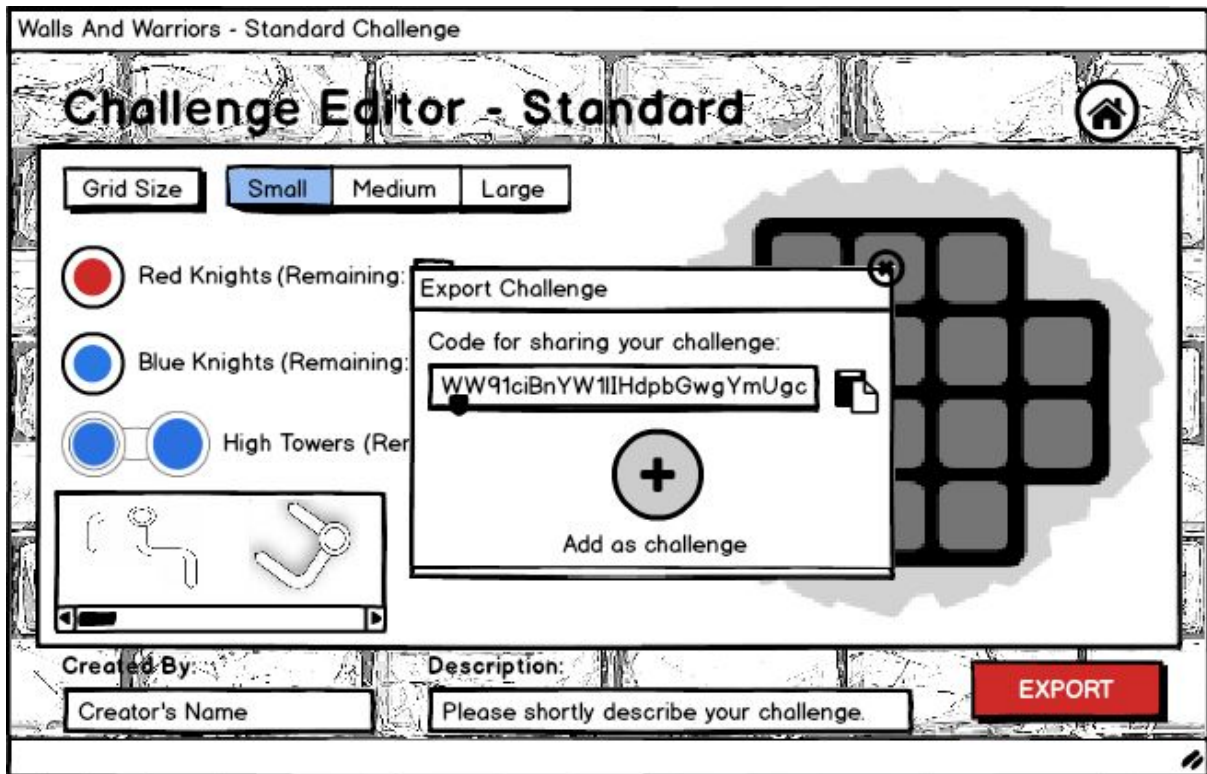

Figure 35: Standard challenge editor

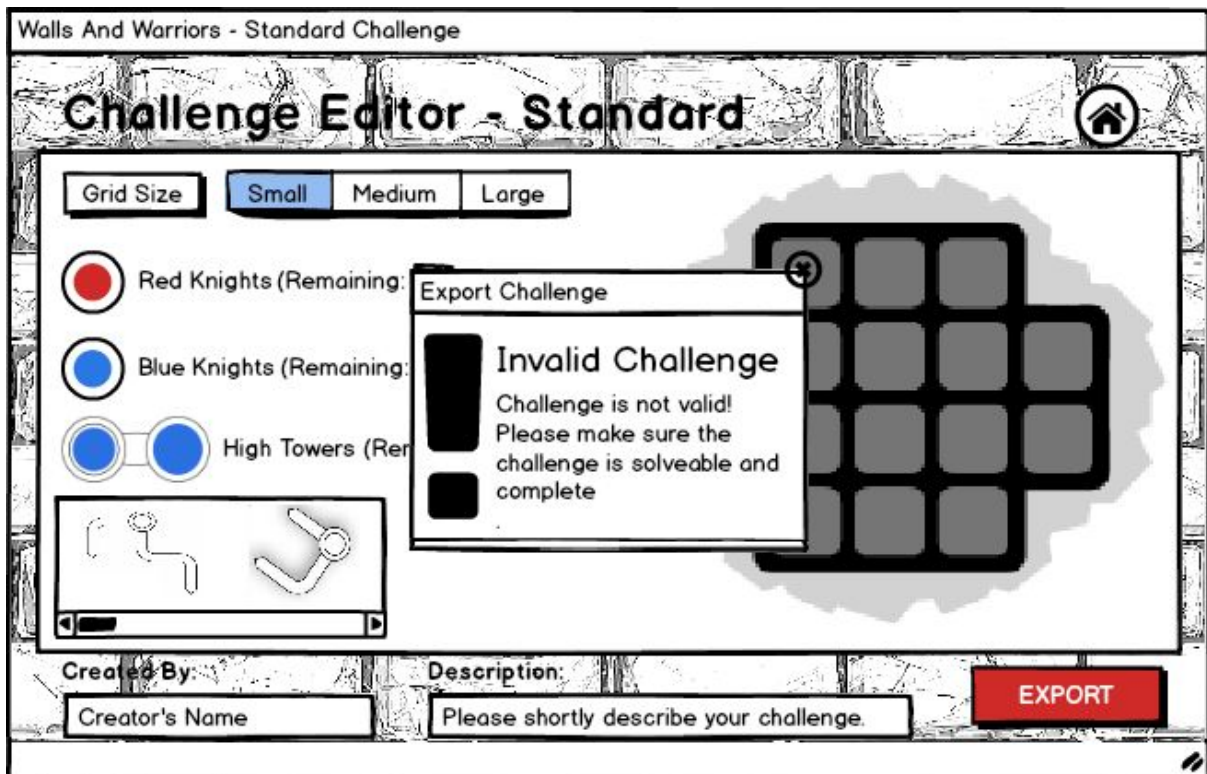Figure 36: Export created challenge

Figure 37: Invalid challenge warning

6. References

[1] Peeters, R. (2016, January). "The Story Behind the Creation of Walls & Warriors". SmartGames.http://www.smartgamesandpuzzles.com. [Accessed: Oct 13, 2018]