



**CS 319 - Object-Oriented Software Engineering  
Analysis Report**

**Instructor: Eray Tüzün**

**Project Name: The Wall**

**Group 3E**

**Erdem Ege Maraşlı - 21602156**

**Ayça Begüm Taşcıoğlu - 21600907**

**Alperen Koca - 21502810**

**Hammad Malik - 21600468**

**Ensar Kaya - 21502089**

## **Contents**

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Overview</b>	<b>4</b>
2.1	Game Play	5
2.2	Controls	5
2.3	Maps	5
2.4	Objects	6
<b>3</b>	<b>Functional Requirements</b>	<b>7</b>
3.1	Start-Menu	7
3.2	Play Game-Menu	7
3.3	Classical Mode	8
3.4	Campaign Mode	8
3.5	Challenge Mode	9
3.6	Developer Mode	9
3.7	Settings	9
3.8	How To Play	10
3.9	Credits	10
<b>4</b>	<b>Non-functional Requirements</b>	<b>10</b>
4.1	User-Friendly Interface	10
4.2	Game Performance	10
4.3	Extensibility	11
4.4	Pseudo Requirements	11
<b>5</b>	<b>System Models</b>	<b>11</b>
5.1	Use Case Model	11

5.1.1	Use Case Descriptions	13
5.2	Dynamic Models	21
5.2.1	Sequence Diagrams	21
5.2.1.1	Classic Mode	21
5.2.1.2	Campaign Mode	22
5.2.1.3	Challenge Mode	23
5.3	Activity Diagram	24
5.4	State Diagram	27
5.5	Object and Class Models	29
6	User Interface	31
6.1	Start-Menu	31
6.2	Settings Screen	32
6.3	Credits Screen	33
6.4	Play Screen	33
6.5	Classical Mode	34
6.5.1	Classical Mode Game-Play	35
6.6	Campaign Mode	35
6.6.1	Campaign Mode Game-Play	36
6.7	Developer Mode	37
6.7.1	Create Map	37
6.7.2	Play Map	38
7	References	38

## **1. Introduction**

Our aim is to develop a 2D strategy game named “The Wall”, which is inspired of the board game “Walls & Warriors”. The main objective of the game is splitting friendly and enemy units by using the designated walls. The units will mostly be regular knights. However, there will also be some additional units with additional features. The designated walls are same as the walls in “Walls & Warriors” and players are expected to use mouse to drag these walls to the game board and split the enemy and friendly units. The game will be a desktop application and can be played by using mouse merely.

“The Wall” has lots of additional features comparing classic Walls&Warriors. The Wall has 4 different games modes such as Classical mode, Challenge mode, Campaign mode, Developer Mode. It also offers user to take hints by spending coins. Players can win coins from completing a map, watching ads or exchange of real money. The Wall has also lots of additional units besides red and blue knights such as spy, horse, green knight, The Night King, Lord Commander. Players can also buy icons and use them instead of regular blue and red knight.

This report consists of an overview of “The Wall”, description of gameplay and game objects. Then, the report specifies functional and non-functional requirements. The system models namely use case, dynamic, object class and sequential diagrams will also be presented. In addition, there will be some example screen mock-ups.

## **2. Overview:**

The Wall will be a single player 2-D game, with its own login database for developer mode and with many different challenges and modes for the player. It is primarily based on the board-game as mentioned above; however, there will be many new additions which will be specified later on in this report. After launching the game, players encounters the login page where they will be required to enter their user-name and password. An option for “Forgot Password.” will also be available to the user. After completing the login process, the

Start-Menu will be displayed to the user, which has the following 5 options: “Play Game”, “Credits”, “Settings”, “How to Play” and “Exit”. Upon choosing the “Play Game” option, another menu is displayed which allows users to select the mode they want to play. The game consists of 4 modes: “Classical Mode”, “Campaign Mode”, “Challenge Mode” and “Developer Mode”. Apart from these modes, a “Back” button will also be present on the same screen which will help to return to the previous menu i.e the “Start Menu”. The “How to Play” option will guide the user about the differences between the modes and give a brief description about the different modes and how to play them.

## **2.1 Game-play**

As the game is based on a board-game, the game-play will essentially be the same in classical and challenge modes; the only difference between the two is that in challenge mode, the user will be given a specific time to complete the task. The goal of the player will be to place all blue knights (or icons chosen to replace them) inside the walls and all the red knights (or icons chosen to replace them) outside the walls and to make sure that the walls are closed on all sides. Thus, the task of the player is to move these objects (walls and knights) to their defined places.

## **2.2 Controls**

The controls for the game are very simple as only the mouse/touchpad will be used. The user will select an object such as a knight or a wall piece with the left-click and drag it to place on the map. Right-click will be used to tilt the objects inside the map by 90 degrees.

## **2.3 Maps**

There will be predefined maps for the classic, campaign and challenge modes.

However, each user will have the option to create and edit his/her own map. These maps will be saved on the database for reusability. The user will also have the option to play maps created by other users in the “Play Designed Maps” option. Upon playing a particular map, the user will be asked to rate the map using the 5-stars system. The maps will be categorized by their ratings and number of times they have been rated by different users. Each map will contain a board-like structure at the middle with visible tiles to place the knights on and places between the tiles to place the wall pieces on. The most basic component of the map will be the tiles or blocks which will be used to create a board-like structure inside the map such as the one in the board game “Walls and Warriors.” On each side of this in-game “board” will be a decorated area such as grass, forests, lakes, mountains etc, depending on the mode selected by the user. In the user-designed maps, this area can be essentially anything the user wants it to look like.

## 2.4 Objects

There will be 4 main types of objects for each mode other than the decorations inside the map. These objects are already defined in the board-game “Walls and Warriors” and are as follows:

- High-Tower piece: There is only one of this, which can be either pre-placed in a given challenge or the user has to place it. This piece has to be placed on the tiles.
- Walls: There are 4 wall pieces, each with its own orientation, length and design. These pieces have to be placed in the space between the tiles.
- Blue Knights: These are the friendly knights of the castle, which must be kept inside the walls, in order to complete any challenge. These pieces have to be placed on the tiles. In total, there will be 4 of these but not all of them might be needed for a particular level/challenge.
- Red Knights: They are the enemies, the ones which must be kept outside the walls of the castle in order to complete any challenge. These pieces have to be placed on the

tiles. In total, there will be 3 of these but not all of them might be needed for a particular level/challenge.

- Spy: They are enemy like red knights with one different aspect which is, player have to keep a spy at least one unit far away from walls to pass a level.
- Horse: Blue knights can ride horses and move their places in the game board.
- Green Knight: Green knights can kill red knights.
- The Night King: They are enemies. The Night King changes the closest blue knight into a red knight. Green knights are useless against them.
- Lord Commander: The Lord Commander can kill The Night King.

However, it is important to understand that these objects can be made to look like something else using the “Change Icons” option in the Start-Menu. So, for example, the Red Knights can be changed to Barbarians. The design and outlook of the high-tower piece and the wall pieces can also be changed; however, the orientation cannot be changed since they are designed in such a way so as to fit at specific parts of the board in order to complete a particular challenge.

### **3) Functional Requirements**

#### **3.1 Start-Menu**

This is basic user screen which will be displayed to each user upon successful completion of the login process. This menu will contain the following 6 buttons: “Play Game”, “Change Icons”, “Settings”, “How to Play”, “Credits” and “Exit”. The exit button will help the user to quit and close the game.

#### **3.2 Play Game Menu**

This menu is displayed when the user clicks on the “Play Game” button in the “Start-Menu”. This menu essentially contains the different modes of the game. As mentioned above the game consists of 4 modes – “Classical Mode”, “Campaign Mode”, “Challenge Mode”, “Developer Mode”.

### **3.3 Classical Mode**

The Wall is a strategy game that is inspired by the board game “Walls & Warriors”. The game offers the classical concept of the board game with and without changes. In the classical mode, players will choose the difficulty level that they desire. Players will be given a map according to the difficulty level and they will start playing “The Wall”. The players can drag the pre-shaped walls to the game board using the mouse. In order to win the game, the player has to create a closed structure using the predefined walls such that all the friendly knight objects will be placed inside while none of the enemy knight objects are inside of the structure. By holding the left click, the walls will be selected and moving the cursor while holding the left-click will mean moving the wall. The players can rotate the wall by using the right click. Every right click will rotate the wall by 90 degrees. Releasing the left-click will place the wall at the location of cursor. In order to put the wall on the board, the tiles coincide with the desired location of the wall should be empty. If the player struggles to find the appropriate position for a wall, he/she can use a hint, which will reveal an accurate location of one wall. If the players thinks that the solution is reached, check button will be pressed and the game will inform whether it is true or not. When the victory condition is reached, player will be asked whether he/she wants to continue playing or not. In the classical mode, there are no additional units. Therefore, the victory condition for this mode is the same as in the board game “Walls & Warriors”.

### **3.4 Campaign Mode**



In the campaign mode, the game play will be nearly similar to classic mode. The player will be given a story that consists of several maps. These maps will be different from the ones in the classic mode and challenge mode. There will be pop up messages to inform the story to the players and there will be additional units with additional features. The player will be able to save/load their game in this mode.

### **3.5 Challenge Mode**

The only difference between this mode and the Classical Mode is that the map will be set randomly and there will be a limited amount of time. Thus, there will be an additional time counter on the screen and the player has to complete the task within this time limit.

### **3.6 Developer Mode**

In this mode, players will be able to develop their own maps by using the same walls as in the classic mode. Similar to the classic mode, there will be a catalog on the right of the screen consisting of units (knights and additional units). The players will drag these units onto the board and will create a pattern according to the rules of the game. The players are also expected to give the accurate solution of the map and save this solution. The players can play their own created maps while they can also play maps created by other players. There will be a database of maps, which will include only the created maps. Players can rate maps created by other users and in order to play these maps, they will be able to download them.

### **3.7 Settings**

This option will give opportunity to player to change the volume, background of the game, and will contain a button named “Change Icons” which can be used to change the icons/pictures of the game-objects. The icons will be locked. When user

pays enough coins player can buy these icons such as barbarians, country flags, animals etc.

### **3.8 How To Play**

This option will give information about game and teach the basic elements of the game to the player.

### **3.9 Credits**

When this option is selected, the names of the developers will be shown.

## **4. Non-Functional Requirements**

### **4.1 User-Friendly Interface**

The User-Interface will be designed and implemented in such a manner that the game can be understood and played easily as well as stimulate interest in the user. The mechanics will be general, similar to those seen in many online board-games. The main menu will have less than 8 buttons and each of them will be understandable. There is also a “How to play” section where users can get help. The game also offers hints to the users while gameplay to make the game comprehensible. We are planning to conduct a survey to detect the understandability of the game among the possible user profiles. For example, a person who is between 12-60 ages, can understand the user interface functionals at most in 10 minutes.

### **4.2 Game Performance**

Since there will be many different users, modes and maps, and we are also

planning to add animations in “Campaign Mode”; we will make sure these implementations don’t cause any performance issues for the user. The game-play should be smooth with a higher FPS rate than the monitor refresh rate (which on average is around 60~). Also, the game shouldn’t strain or add extra load on the computer and thus affect the performance of the computer.

### **4.3 Extensibility**

When creating any software, extensibility and reusability are always important considerations. As it is often observed that there are continuous updates to all famous applications. Thus, “The Wall” will also be implemented such that it can be extended and updated with time according to the needs and feedback received by the users.

### **4.4 Pseudo Requirements**

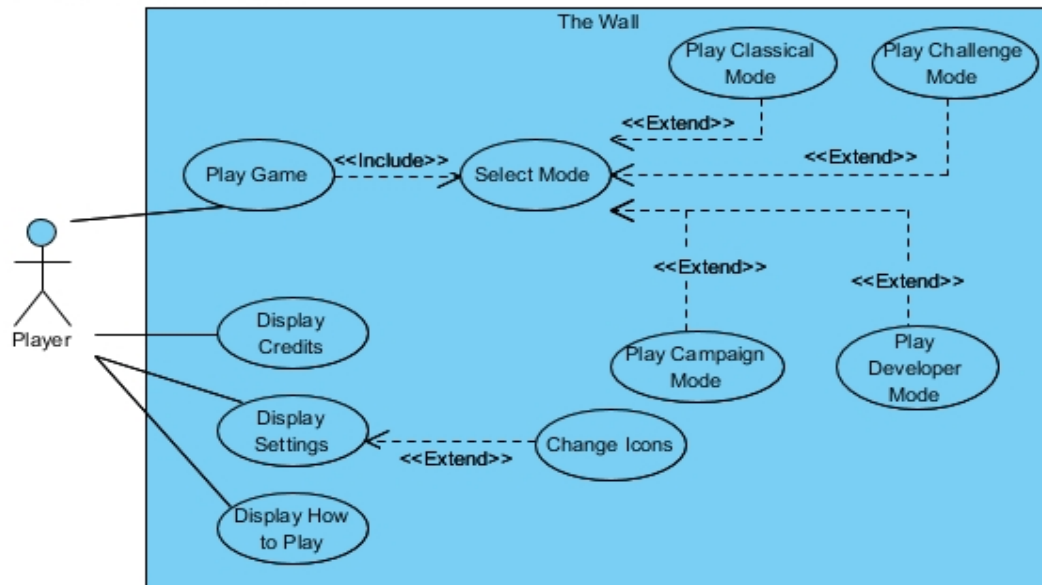
The game will be implemented in Java; graphics will be implemented using Java’s AWT and swing libraries. Also, since it will be an online game, the online login process will be implemented using PHP.

## **5. System Models**

### **5.1 Use Case Model**

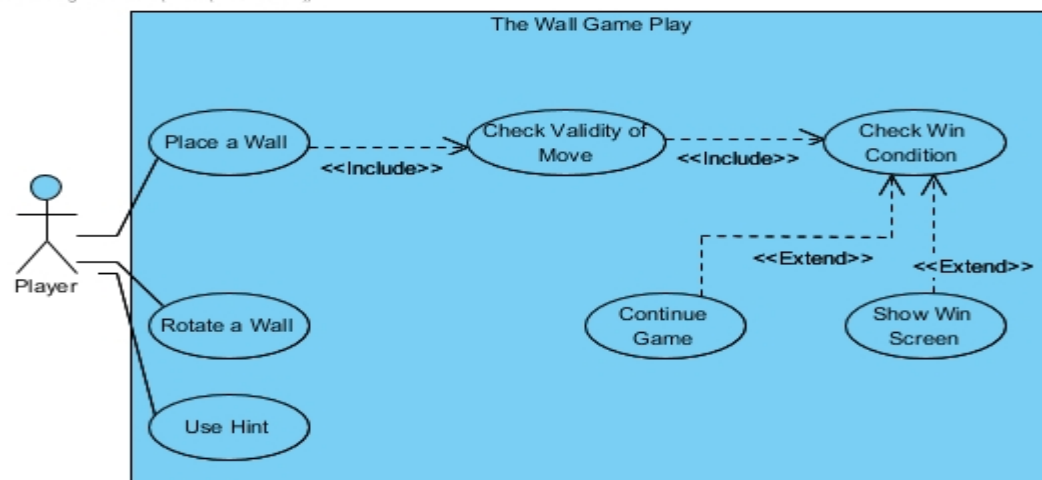
## The Wall Main Menu Use Case Diagram

Visual Paradigm Standard (Ersan(Bilkent Univ.))

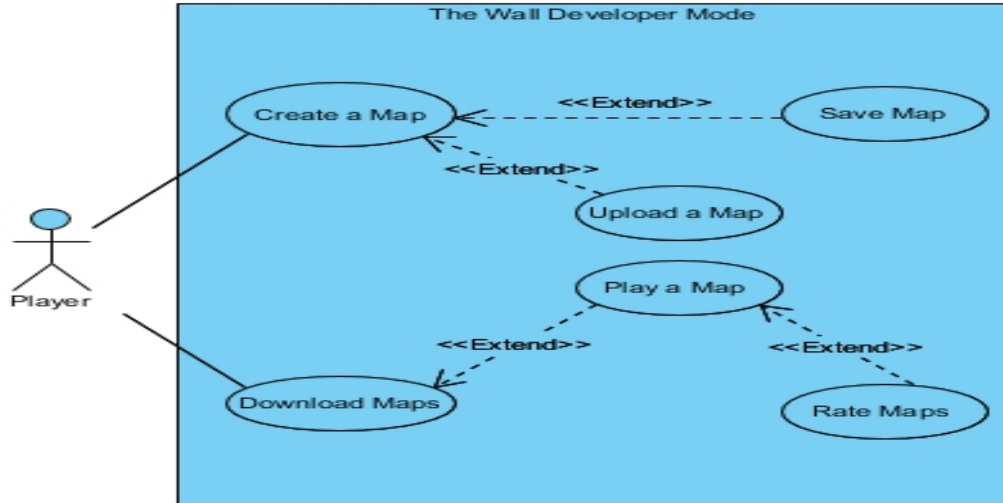


## The Wall Game Play Use Case Diagram

Visual Paradigm Standard (Ersan(Bilkent Univ.))



## The Wall Developer Mode(Create Map) Use Case Diagram



## 5.1.1 Use Case Descriptions

### Use Case #1

**Use Case name:** Play Classic Mode

**Participating actors:** Player

**Entry Condition:** The player opens the game; he first clicks Play Game and secondly Classic Mode button.

**Exit Conditions:**

- 1-Player completes all the provided levels.
- 2-Player wants to exit the game, the first player returns the main menu and then clicks the Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Play Game button
3. Modes screen appears.
4. Player clicks Classic button
5. System prepares levels and allows player to play any unlocked map.
6. Levels screen appears.

7. Player clicks one of the map buttons
8. Player completes all the levels provided
9. Player returns opening page
10. Player clicks Exit button

**Alternative Flow:**

1. Player selects another level
2. Player completes all the provided levels
3. Player clicks Exit button

**Use Case #2**

**Use Case name:** Play Campaign Mode

**Participating actors:** Player

**Entry Condition:** The player opens the game; he first clicks Play Game and secondly Campaign Mode button.

**Exit Conditions:**

1-Player finishes story which mean player completes all the provided levels in Campaign

2-Player wants to exit game, first player returns main menu and then clicks Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Play Game button
3. Modes screen appears.
4. Player clicks Campaign button
5. New game or load game screen appears.
6. Player clicks New Game button
7. System prepares levels and allows player to play first map
8. Player completes the level provided, won coins

10. Game provides option for continue story or close game
11. Player clicks continue story button
12. Player completes all the maps in the story
13. The system displays congratulation message and returns main menu

**Alternative Flow:**

1. Player clicks Continue Game Button
2. Player continues play from his/her last map
3. Player completes all the remaining maps
4. The system displays congratulation message and returns main menu

**Use Case #3**

**Use Case name:** Play Challenge Mode

**Participating actors:** Player

**Entry Condition:** The player opens the game; he first clicks Play Game and secondly Challenge Mode button.

**Exit Conditions:**

- 1-Player completes all the challenges provided from system
- 2- Player cannot succeed to complete the challenge in given time
- 3-Player wants to exit game, first player returns main menu and then clicks Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Play Game button
3. Modes screen appears.
4. Player clicks Challenge button
5. System prepares a challenge
6. Player plays the challenge
- 7.If player is successful, player won coins.
8. New Challenge or close game option screen appears

9. Player clicks close game button

#### **Use Case #4**

**Use Case name:** Play Developer Mode (Create a map)

**Participating actors:** Player

**Entry Condition:** The player opens the game; he first clicks Play Game, secondly Developer Mode button and thirdly Create Map button.

**Exit Conditions:**

1-Player creates a map and uploads it to the system

2-Player wants to exit game, first player returns main menu and then clicks Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Play Game button
3. Modes screen appears.
4. Player clicks Developer button
5. Player clicks Create Map button
6. System provides an empty game board and game units
7. Player creates his/her own map
8. Player clicks Save Map button
9. System checks if it is a valid map
10. Player clicks upload map button
11. System uploads map to the server
12. Player clicks Exit button

**Alternative Flow:**

1. Player clicks Save Map button
2. System check if it is valid map
3. System displays error message

#### **Use Case #5**

**Use Case name:** Play Developer Mode (Play a map)



**Participating actors:** Player

**Entry Condition:** The player opens the game; he first clicks Play Game, secondly Developer Mode button thirdly Play Map button.

**Exit Conditions:**

- 1-Player plays a map completes it
- 2-Player wants to exit game, first player returns main menu and then clicks Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Play Game button
3. Modes screen appears.
4. Player clicks Developer button
5. Player clicks Play Map button
6. System provides an screen which contains shared maps
7. Player selects a map
8. Player completes selected map
9. Player clicks return main menu button
10. Player clicks Exit button

**Alternative Flow:**

1. Player selects a map
2. Player completes the map
3. Player vote for the completed map

## **Use Case #6**

**Use Case name:** How to Play

**Participating actors:** Player

**Entry Condition:** The player opens the game; he clicks How to Play button

**Exit Conditions:**

- 1-Player wants to exit game, first player returns main menu and then clicks

Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks How to Play button
3. How to Play screen appears.
4. Player clicks return main menu button
5. Player clicks Exit button

**Use Case #7**

**Use Case name:** Settings

**Participating actors:** Player

**Entry Condition:** The player opens the game; he clicks Settings button

**Exit Conditions:**

1-Player wants to exit game, first player returns main menu and then clicks Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Settings button
3. Settings screen appears.
4. Player clicks voice button for closing voice of game
5. Player clicks Change Icons button
6. Icons page appears
7. Player clicks one of the icons for blue knights, if he has enough coins
8. Player clicks one of the icons for red knights, if he has enough coins
9. Player clicks return main menu button
10. Player clicks Exit button

**Use Case #8**

**Use Case name:** Credits

**Participating actors:** Player

**Entry Condition:** The player opens the game; he clicks Credits button

**Exit Conditions:**

1-Player wants to exit game, first player returns main menu and then clicks Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Credits button
3. Credits screen appears.
4. Player clicks return main menu button
5. Player clicks Exit button

**Use Case #9**

**Use Case name:** Exit Game

**Participating actors:** Player

**Entry Condition:** The player opens the game; he clicks Exit button

**Exit Conditions:**

1-Player wants to exit game, player clicks Exit button.

**Main Flow:**

1. Player opens the game
2. Player clicks Exit button
3. System asks player if s/he is sure or not
4. Player clicks “Yes” if s/he wants to close the game

**Use Case #10**

**Use Case name:** Using Hints

**Participating actors:** Player

**Entry Condition:** The player opens the game; he plays the game.

**Exit Conditions:**

- 1-Player wants to exit game, player clicks Exit button.
- 2-Player runs out of time.

**Main Flow:**

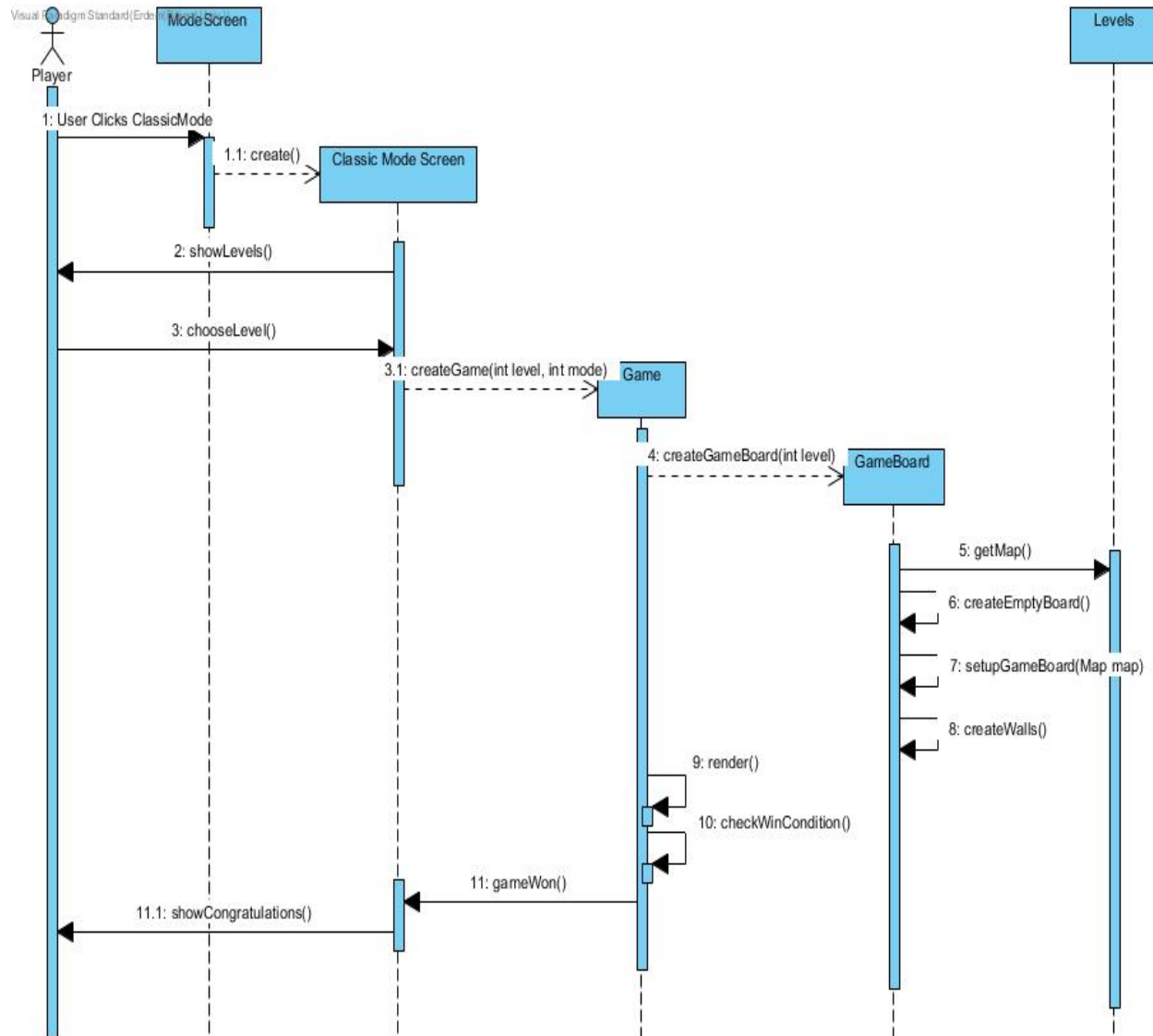
1. Player opens the game
2. Player clicks Play Game button

3. Player clicks Challenge button
4. Game System prepares proper maps and starts a countdown.
5. Player wants to use a hint. Clicks hint button.
6. Player has enough coins so game display a wall's right place.
7. Player cannot completed challenge.
8. New Challenge or close game screen appears

## 5.2 Dynamic Models

### 5.2.1 Sequence Diagrams

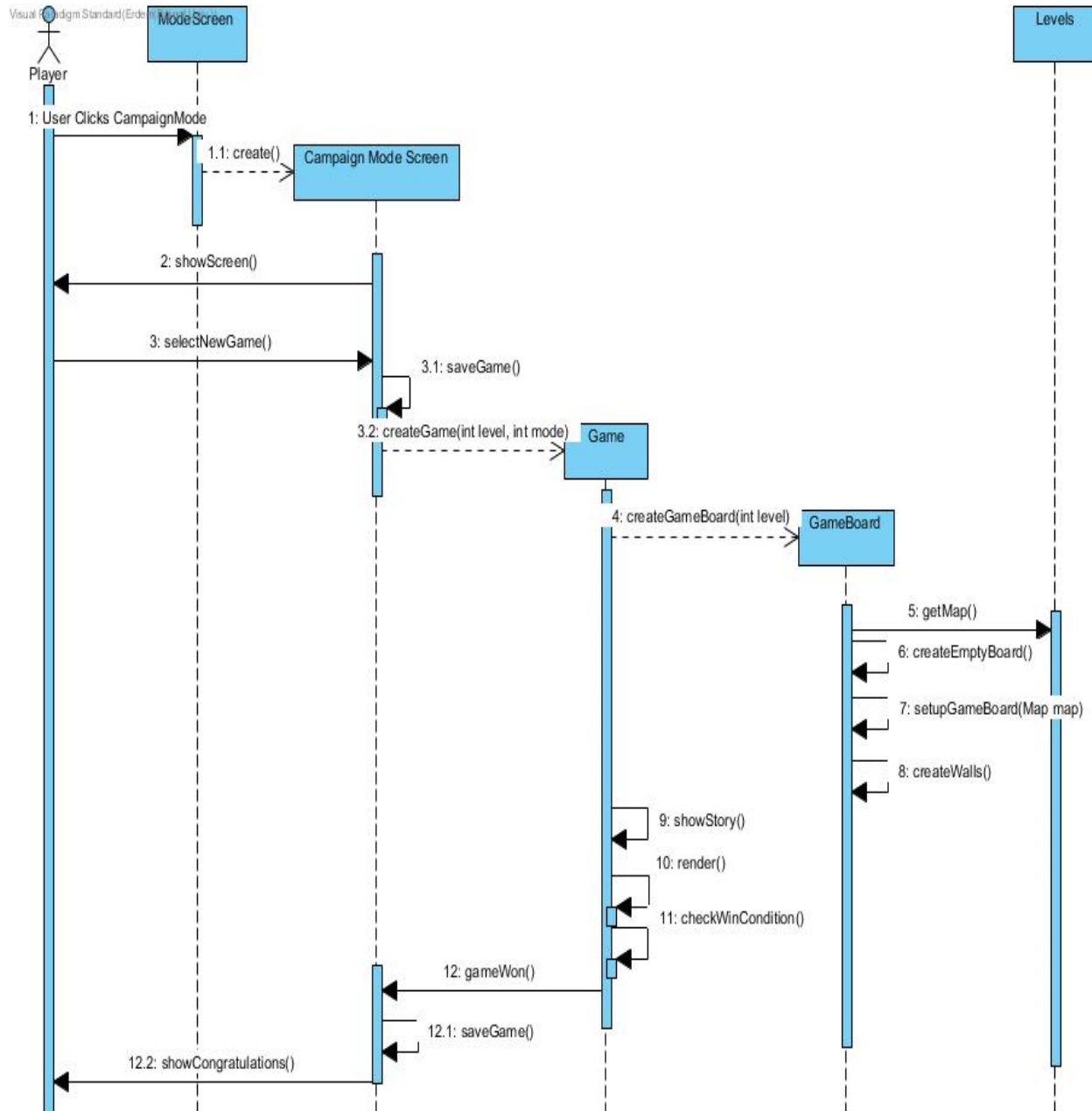
#### 5.2.1.1 Classic Mode



Scenario: In this scenario, the player selects the classic mode. After his selection, available levels are shown to the player. The player selects the level, controller class game created and the map information taken from levels class. Game board setups according to that information. After the game board setup done, the player can start to make moves. Game renders itself every time. After every move of player; game

controls if the player wins the game or not. If all win conditions satisfied player passes the level and message shown to the player.

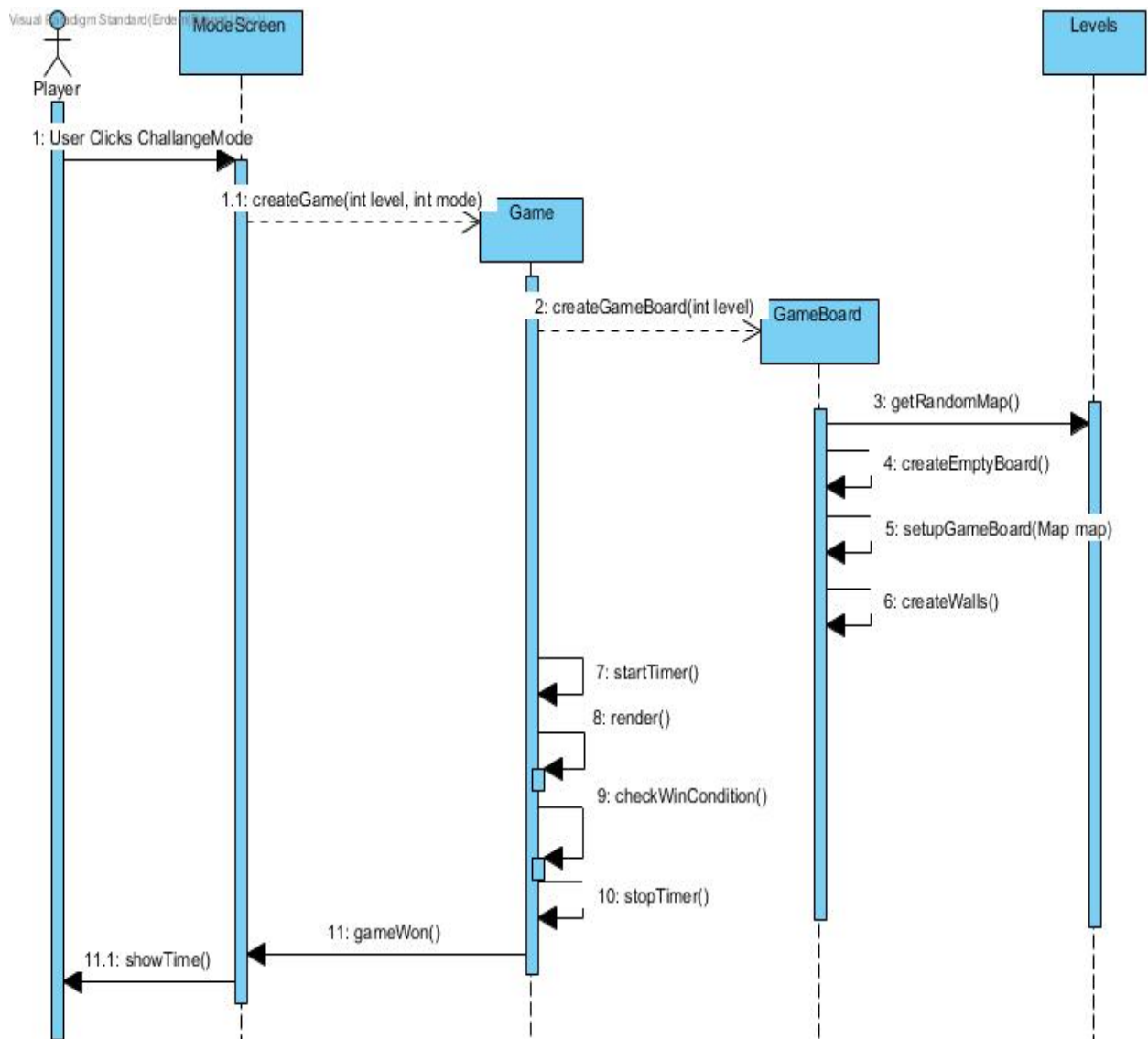
### 5.2.1.2 Campaign Mode



Scenario: In this scenario, the player selects the campaign mode. After his selection, player selects start new game. New game selected so old save file deleted and saved as a new game. Controller class game is created. That level's map

information taken from levels class. Game board setups according to that information. After the game board setup done, the player can start to make moves. Story pop ups show to the player. Game renders itself every time. After every move of player; game controls if the player wins the game or not. If all win conditions satisfied player passes the level, game saves itself and message shown to the player.

### 5.2.1.3 Challenge Mode

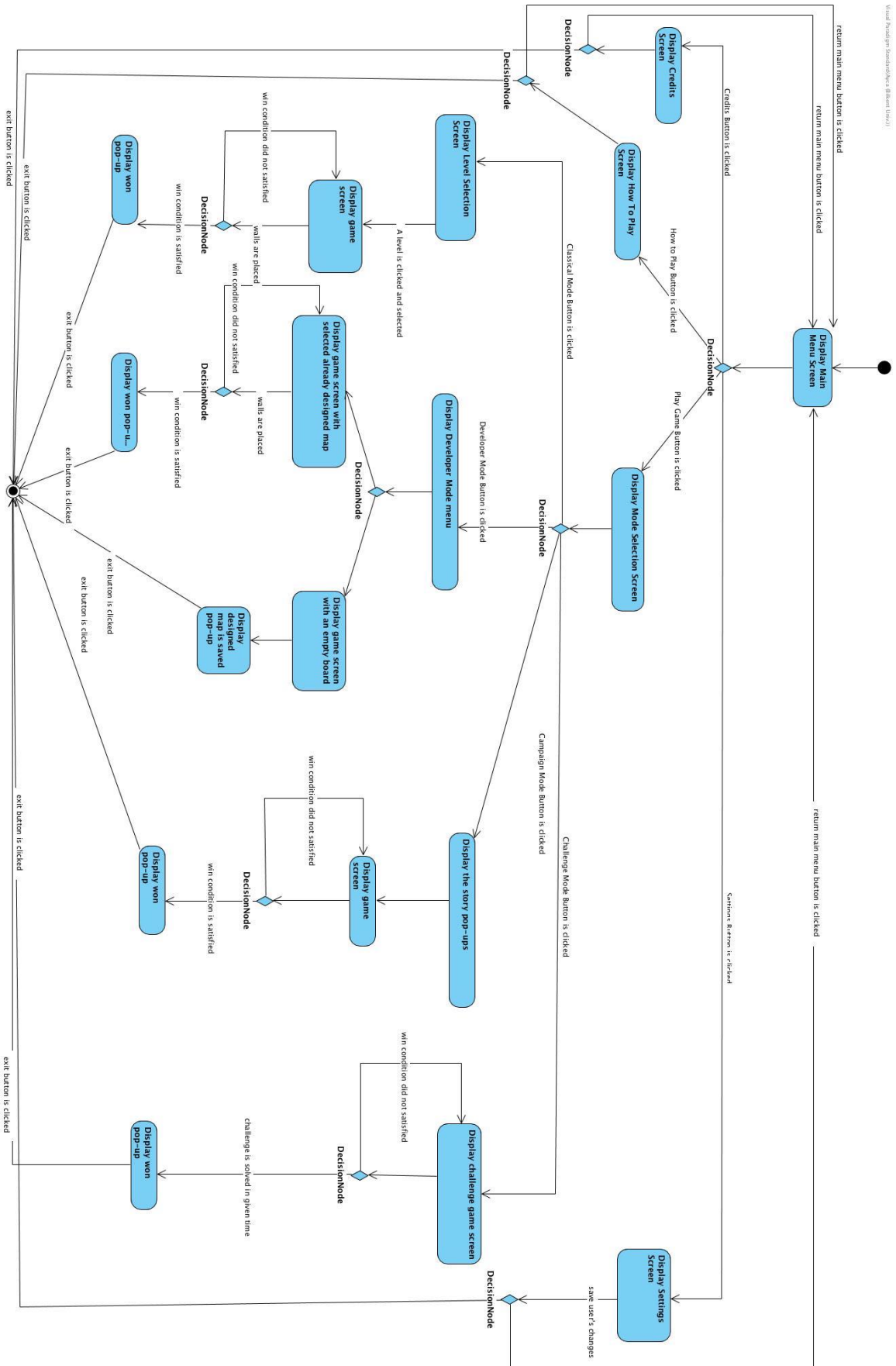


Scenario: In this scenario, the player selects challenge mode to play. After his selection, Controller class game is created, random map is selected and game board

setups according to that maps. After the game board setup done, the timer starts to count the time and player can start his moves. Game renders itself every time. After win condition satisfied, the timer stops and passed time shown to the player.

### **5.3 Activity Diagram**





Initially, program displays Main Menu screen which has 4 buttons such as Credits, Settings, How to Play and Play Game buttons.

If user clicks Credits Button the developer of the games will be displayed in Credits Screen.

If How to Play Button is clicked How to Play Button rules of the game will be displayed in How To Play Screen.

If Settings Button is clicked than there is Setting Screen will be displayed which that user can change and set special options such as sound control and background control.

If Play Game Button is clicked Mode Selection Screen will be displayed which user can select the game modes. In Mode Selection Screen there will be 4 buttons: Classical Mode Button, Campaign Mode Button, Developer Mode Button, Challenge Mode Button.

If user clicks Classical Mode Button then Level Selection Screen will be displayed. When the user clicks a level, Game Screen will be displayed which is ready to play. While the win condition is not satisfied ( user did not kept all the blue knights and castle inside the walls and red knights outside the walls), user keeps playing otherwise if win condition is satisfied, a “won” pop-up will be displayed.

If user clicks Campaign Mode Button then story pop-ups will be displayed. After, the Game Screen will be displayed with a game board and units(additional and default units); game is ready to play. While the win condition is not satisfied, user keeps playing. Otherwise; if win condition is satisfied, a “won” pop-up will be displayed.

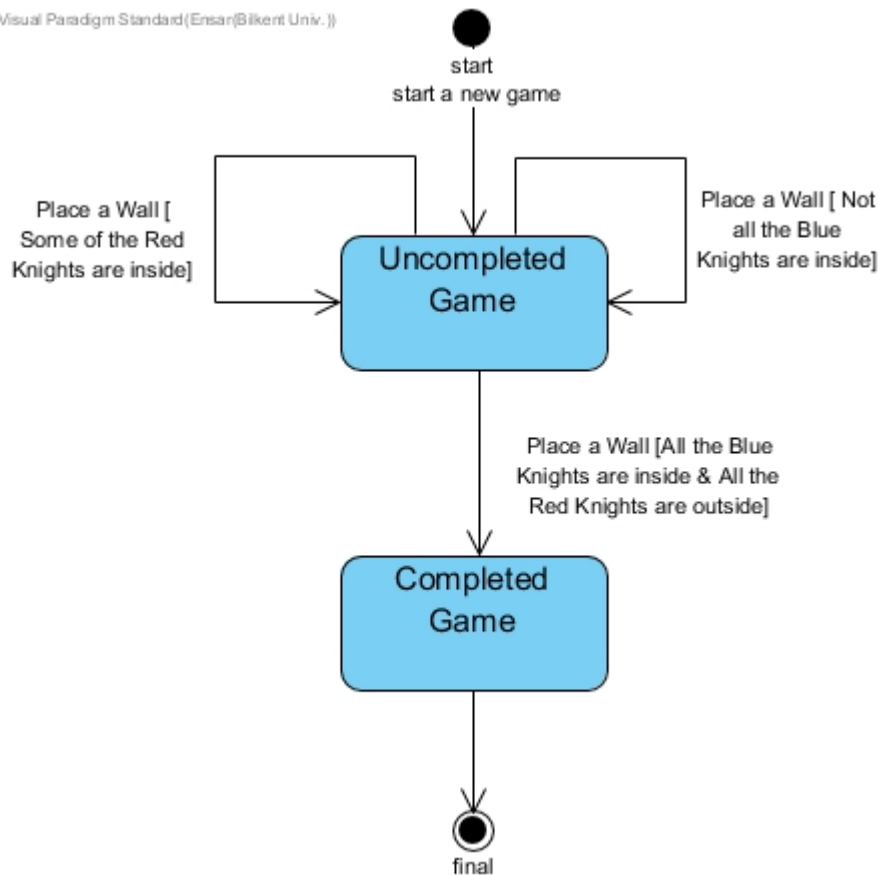
If the Developer Mode Button is clicked then if the playing with already designed maps is selected, A Game Screen with already designed map will be displayed; game is ready to play. While the win condition is not satisfied, user keeps playing. Otherwise; if win condition is satisfied, a “won” pop-up will be displayed. If user clicks to Create a New Map button than the Game Screen will be displayed with an empty game board, after; the designed map

will be saved.

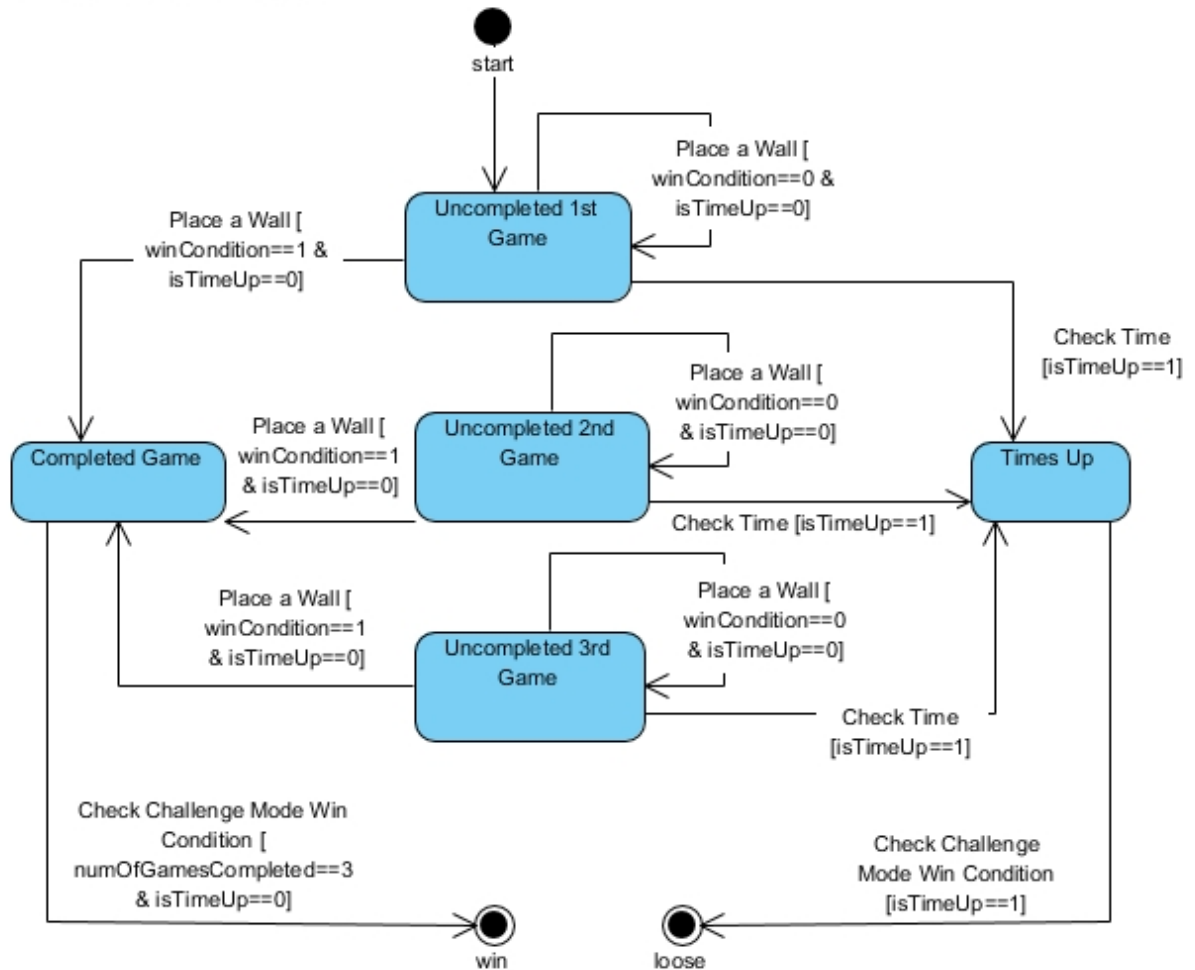
If user clicks Challenge Mode Button then the Game Screen will be displayed with a game board and units(additional and default units) and the remaining time; game is ready to play. While the win condition is not satisfied (also user should solve in given time), user keeps playing. Otherwise; if win condition is satisfied, a “won” pop-up will be displayed.

## 5.4 State Diagram

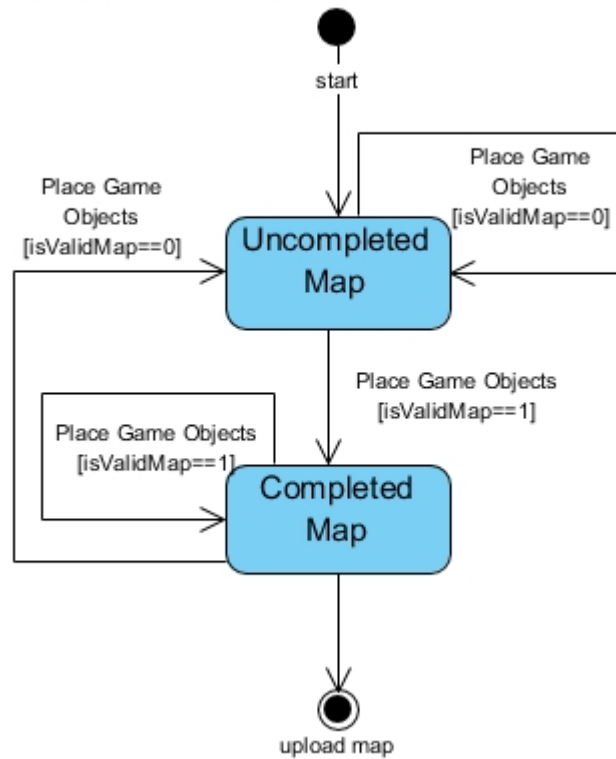
Visual Paradigm Standard(Eraser)(Bilkent Univ.)



Classical mode game play has 2 states which are uncompleted and completed game states. When player place a wall into game board, game check if all the red knights are outside of the walls and all the blue knights are inside the walls. If both statements are true, then state transition occurs and our new state will be completed game state.

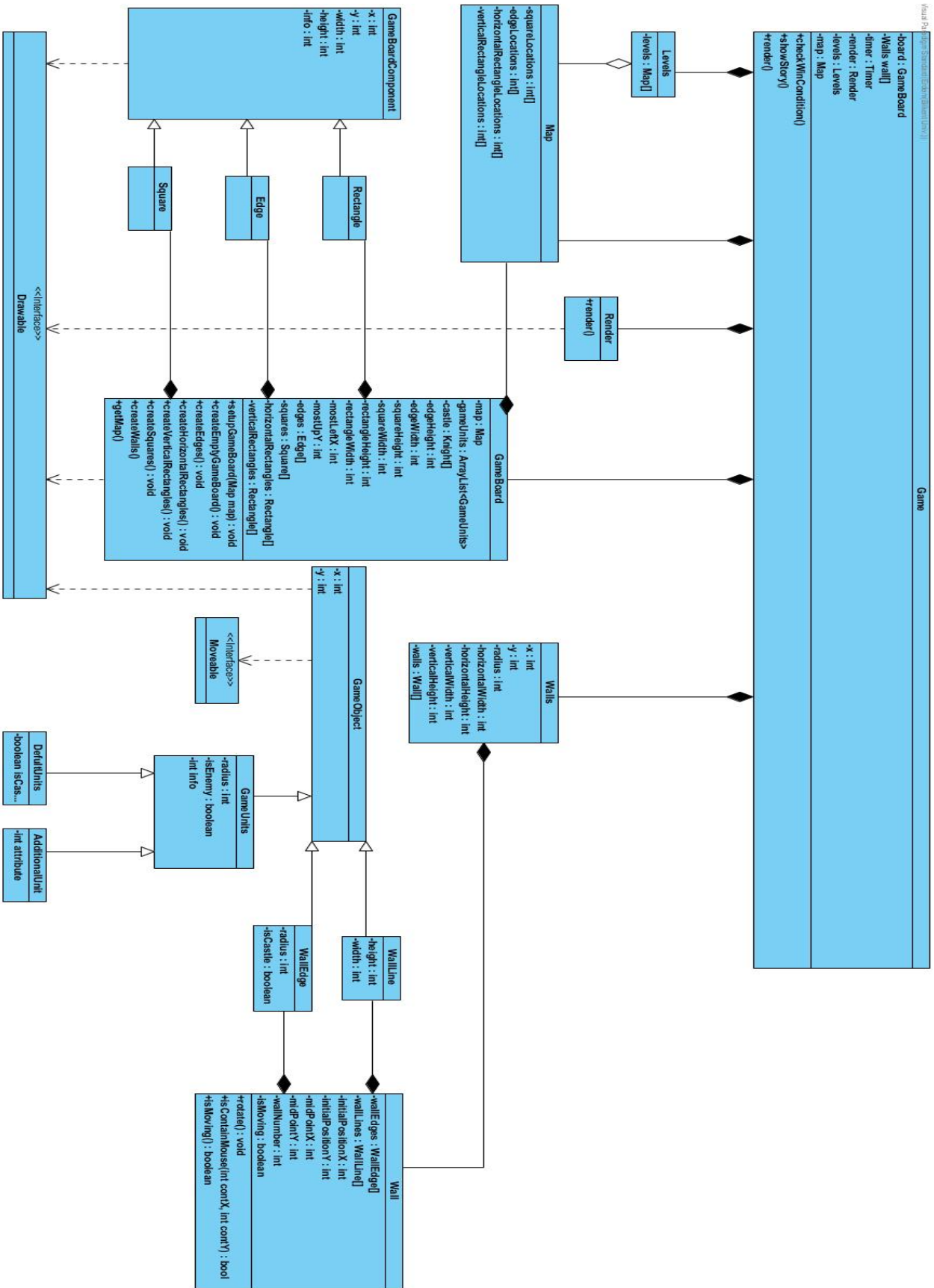


In challenge mode, we have 5 states which are completed game, uncompleted 1st game, uncompleted 2nd game, uncompleted 3rd game and Times up. We start with uncompleted 1st game state. Transition to next state occurs when times is not up and win condition is true which means all the red knights are outside of the walls and all the blue knights are inside the walls. This process goes like this until all 3 games are completed or time is up in any of the states. If time is up then we lose the game. If time is not up and all 3 map completed in the proper time then we win the game.



In developer mode create map scenario, we have basically 2 states which are uncompleted map and completed map. We start the scenario with uncompleted map state. When player place a game object into game board, our game checks the validity of map. If it's a valid map, then we go to next state, completed map. If player make changes after reaching this state, game engine check moves of player. According to this moves state can be change again if validMap condition goes false. We finishes this state diagram with update or save map options.

## 5.5 Object and Class Diagram



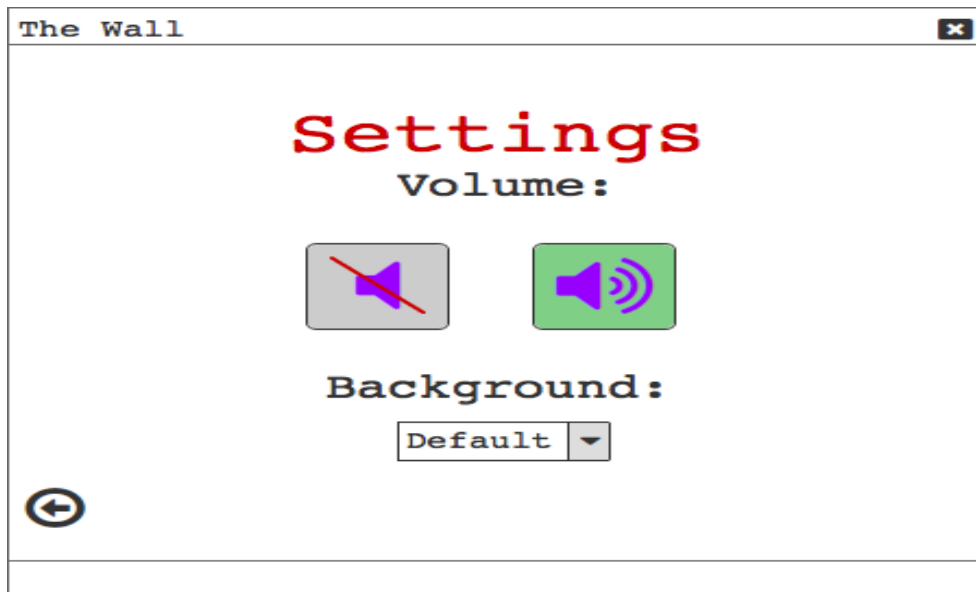
## 6. User Interface

### 6.1 Start Menu



In starting screen there will be 4 options for player: Play Game button which is for start the game, Credits for developers and references, Settings for changing volume, changing background color and changing icons for game units. And How to Play button informs player about the game.

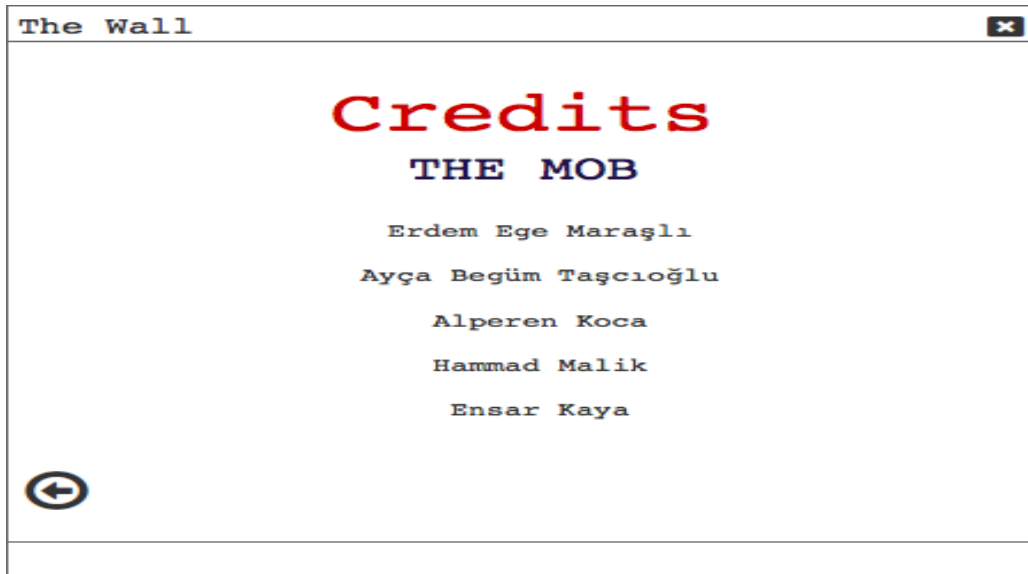
## 6.2 Setting Screen



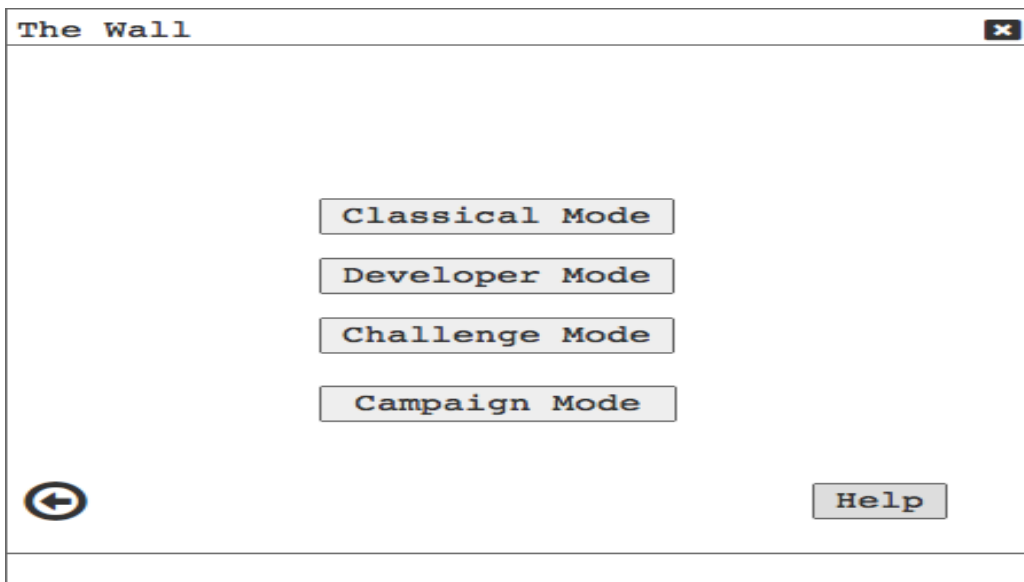
Users can close and open the sound and change the background. We are also planning to add change icon option to the settings panel which will let users change their knight's icon with enough coin.



## 6.3 Credits Screen



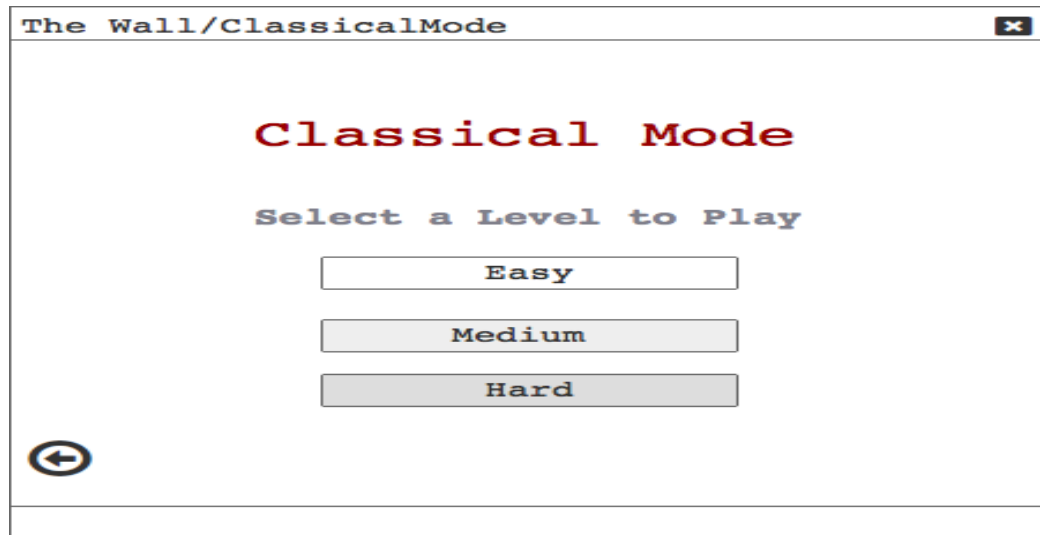
## 6.4 Play Screen



When user clicks Play Game button, this screen will be displayed by the system. In this screen player has 4 options such as Classical mode, Developer mode, Challenge

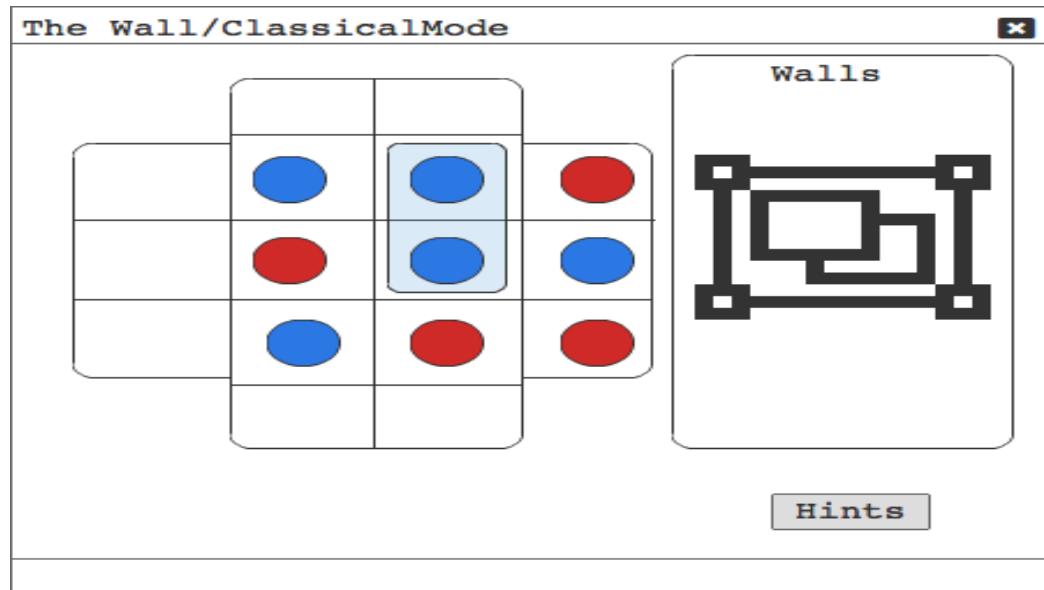
mode and Campaign mode.

## 6.5 Classical Mode



If the player selects classical mode this screen will be display. There are 3 options for difficulty levels.

### 6.5.1 Classical Mode Game Play



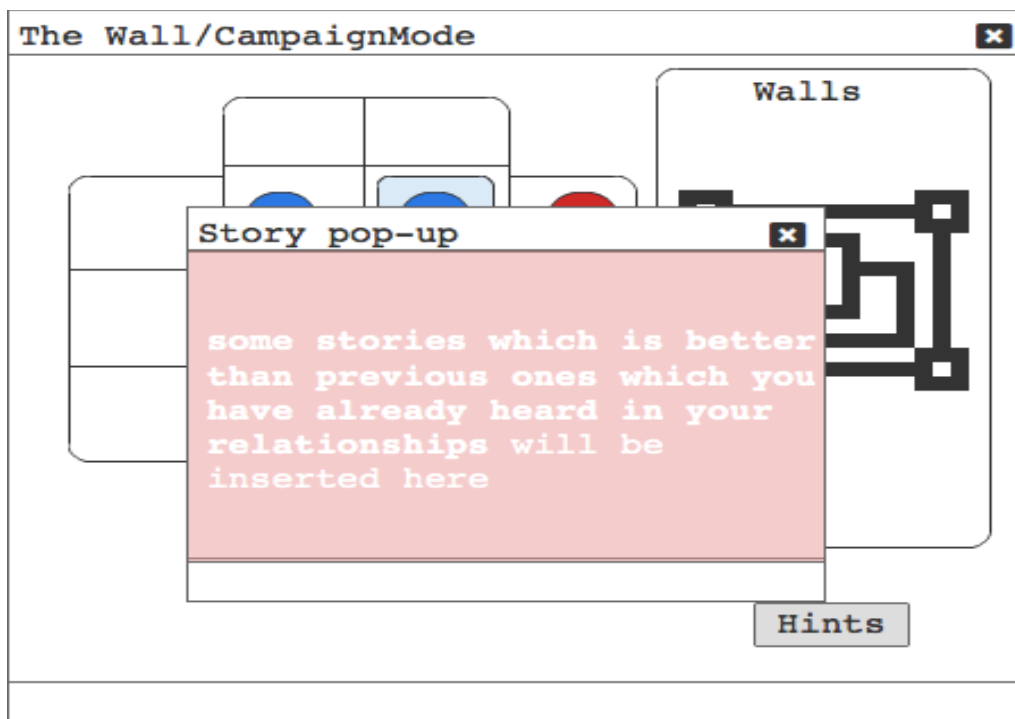
The game board will setup by system according to player choose. Walls will be stand on the right hand side and game board will be stand on the left hand side

### 6.6 Campaign Mode



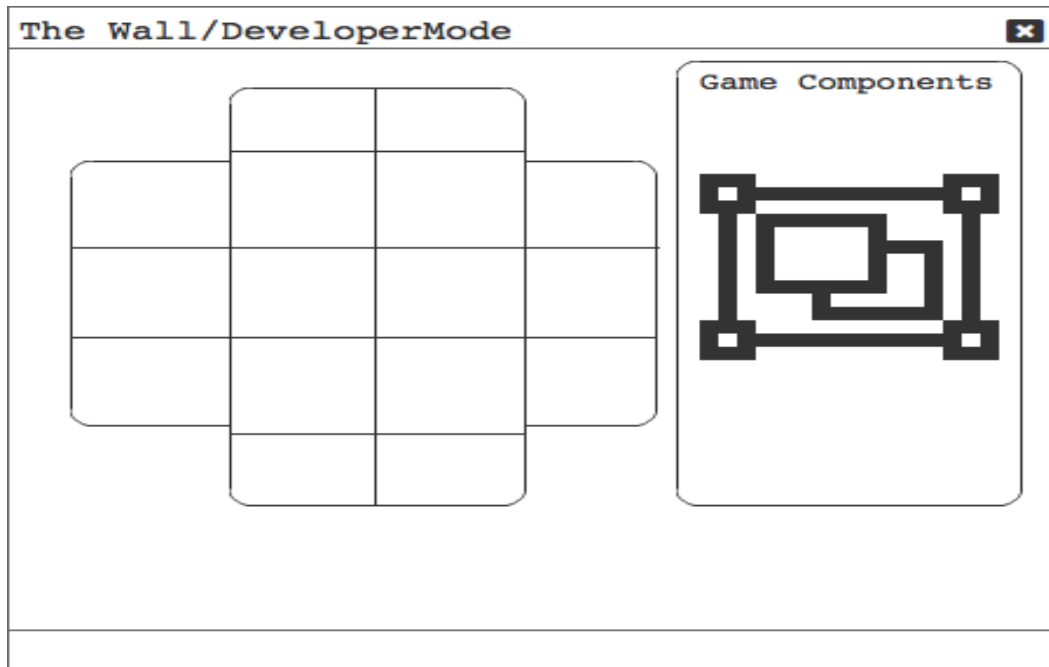
If the player clicks on the Campaign button, this screen will be display. Player can start a new game or continue his/her old game.

### 6.6.1 Campaign Mode Play Screen



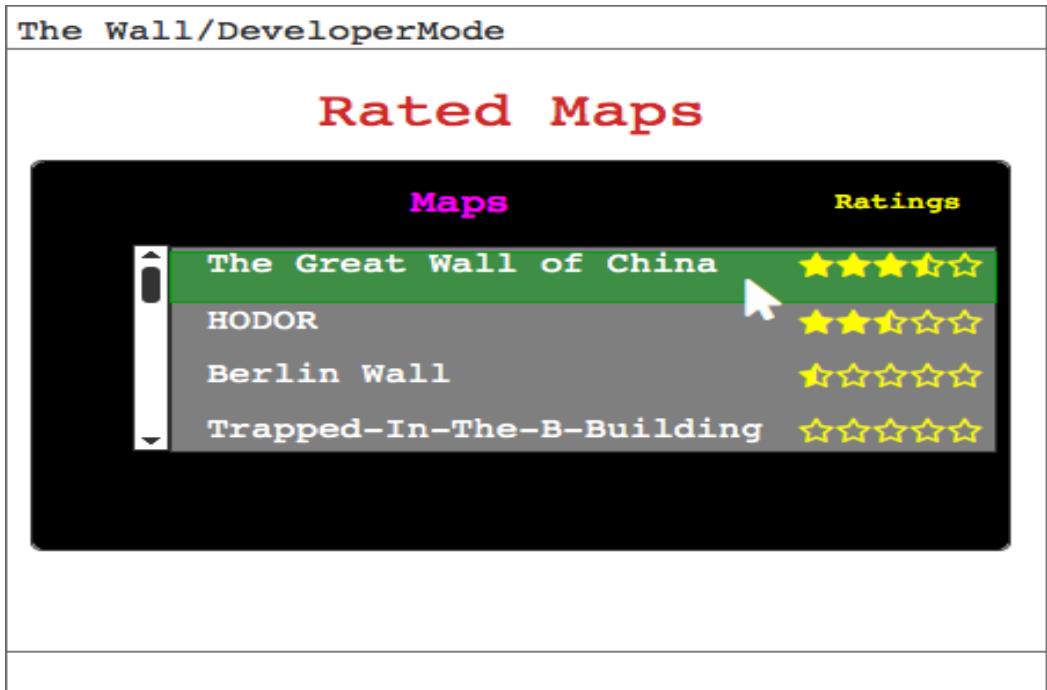
## 6.7 Developer Mode

### 6.7.1 Create Map



If the player selects Developer button there will 2 options for player: Create Map and Play Map. In create map case, this screen will be display. The player will have the full access to walls and game units to place unless the whole map is invalid.

### 6.7.2 Play Map



The player can choose maps which are created by other users. They can also see the rating of the map and if a player completes a map, s/he can vote the map..

### 7. References

*Walls & Warriors*. [www.smartgames.eu/uk/one-player-games/walls-warriors](http://www.smartgames.eu/uk/one-player-games/walls-warriors).