CS319 Object-Oriented Software Engineering

Project Analysis Report

Ferhat Serdar Atalay

Aylin Çakal

Ali Bulut

Ismail Serdar Taskafa

Table of Contents

1. Introduction

Who Doesn't Want to be a Zillionaire Anyway is a java game we chose as a project. The game is mixture of famous TV show *Who Wants to be a Billionair*e and the box game *Trivial Pursuit*. Who wants to be a Billionaire is a TV show that competitors try to answer 15 questions correctly with some lifelines. At the end of the competition, competitor earn money according to its number of correct answer. In the box game, Trivial Pursuit, the players try to answer questions in 6 different categories. In our project, the game format will be similar to Who Wants to be a Billionaire. Differently, in order to win the game the players have to get 5 points in each category.

Currently game will have the following features:

-Single Player Mode

-Level Selection [Easy - Medium -High]

-Different Categories (6 Categories)

-3 Lifelines [extendable]

-Pass

-Visual / Audio Questions

To implement the game, we are planning to use java programming language. We chose java because it is known by all group members, and UI is easy to use and also it is very suitable for object-oriented programming. Moreover, the development process will heavily build upon object oriented programming, allowing us to learn and practice the principles.

2- Game Overview

2.1 Gameplay

“Who Doesn't Want to be a Zillionaire Anyway?” is a single player game. After selecting Play Game in main menu, user can decide whether to play on Easy, Medium, or Hard mode. This selection will affect the whole game and during the gameplay each question will more or less be the same. There are 6 categories which come in a random order. For skipping to the next one, the player has to answer 5 questions right in each category. Each category contains 15 questions, which the player can skip as many times as desired. Any skipped question will be added to the end of the queue. The player can use each of 3 lifelines only once, which one of them is given initially 2 of them are unlocked as user completes category 2 and 4 respectively.

2.2 Categories

There are 6 categories in our game, which are the following:

- Geography
- Entertainment
- History
- Arts & Literature
- Science & Nature
- Sports

Those categories come in a random order. The player can skip to the next category when they collect 5 points in the current one. After completing the 6th category, the player wins the game.

2.3 Questions

Each category has 15 questions which can be either in text, image, or audio format. Texts are regular questions where image and audio questions mostly based on the knowledge about the piece, or something else related with it. The questions have 4 different answers which are labeled as A, B, C and D.

The wrong answers will cause player to lose points, forcing them to answer an additional question right to recompense the mistake. Each wrong answer causes to lose 1 point while each right answer brings 1 point. Before consuming all of 15 questions, player has to answer 5 questions right; more than they answer wrong. Player can skip questions as many times as they desire. Those unanswered questions will line up at the end of the question queue, allowing user to answer later.

2.4 Players Initial Items

Player can choose among 3 different difficulty types which are Easy, Medium, and Hard. The categories will be shuffled after choosing the difficulty level, and appropriate questions will come accordingly.

Throughout the gameplay, the player can pass questions as many time as they desire. The first lifeline, Ask the Audience, will be given initially.

2.5 Lifelines

There are 3 lifelines which may help player throughout the game. The lifelines will never guarantee the right answer, or make a selection for the player. Initially there is only one lifeline. Each of the other lifelines will be unlocked after successfully completing 2 categories.

Those lifelines are the following:

1. Ask the Audience

This lifeline will be simulated as there is an audience in the studio, and they claim their answer to help the player. In different difficulty levels, the usefulness of this lifeline will change since the audience is less likely to provide a better statistics in harder questioıns.

2. Phone a Friend

Phone a Friend lifeline will give you better results if you choose “the right guy for the job”. Each of the friends will give better results in different categories, which means they can also fool you if you ask for their help in a topic they know nothing about. After getting an recommendation, it is still up to the player to answer the question.

3. 50-50

This is arguably the most helpful lifeline which erases 2 of the wrong answer, allowing even a random answer to be true with a 50% chance.

2.6 Settings

The user can adjust the sound volume, or mute/unmute completely.

3. Functional Requirements

3.1 Play Game

The game is single player. When the player runs the game, Main Menu comes first. Main Menu consists of Play Game, How to Play, Options (Settings) and Credits buttons. The game is mouse controlled. To select the buttons, the player should click on it. When the player selects “Play Game” button by clicking on, Level Selection Panel comes. After selecting the level, the game starts with random category. Play Game button is a tool for level selection.

3.2 Select the Level

After clicking on the Play Game, Level Selection Panel appears. The player should select the level of the game before the category comes. There are three buttons which are Easy, Medium and Hard. If the player selects “easy” button, all categories will consist of the easy questions. In case of “Medium” and “Hard” selections, it will be the same. The player is directed to Level Selection Panel consisting of the level buttons after the Main Menu. The player may select the level of the game by clicking on Easy, Medium or Hard button.

3.3 How to Play

The player should know about some information of the game before starting. How to Play button which is located in the Main Menu informs the player about the rules, levels and mouse controlling levels of the game, button expressions such as lifelines, pass in the game. The player may access How to Play part from the Main Menu by clicking on the How to Play button.

3.4 Options (Settings)

Options part includes some changeable settings of the game. The player may access this button from the Main Menu by clicking on the Options button.

Changeable settings of the game are as follows:

- Adjusting the volume
- Mute / unmute the volume

3.5 Credits

Credits part includes the information about the game's developers and their email addresses. This button is located in the Main Menu which is the initial screen. The player may access the Credits part from the Main Menu by clicking on the Credits button.

4. Non Functional Requirements

4.1 Game Performance

The game will appear high-quality with not only text based questions but also audio and visual questions which do not affect the speed of the game. Another point is frame ratio which would affect the game performance. It should not depend on the computer. This desktop game will be compatible with all computers.

4.2 User-Friendly Interface

The User Interface is directly connected with the player's game approach. Graphical User Interface will be tried to be prepared in the best way. In the game, the lifelines, the questions and the player’s total score will be written in the panels will offer player a relaxed atmosphere. The game will be fully screened and the player will feel a real competition environment.

4.3 Clarity of the Questions & Answers

Throughout the game, the player should understand the text based questions, recognize visual questions, and should not have any problems hearing audio questions. Therefore, the resolution of the images used will be such that the player can recognize them easily. The audio files used will be of the same quality and will not be difficult to hear. In short, the contents of the game should be very clear.

4.4 Challenge of the Game

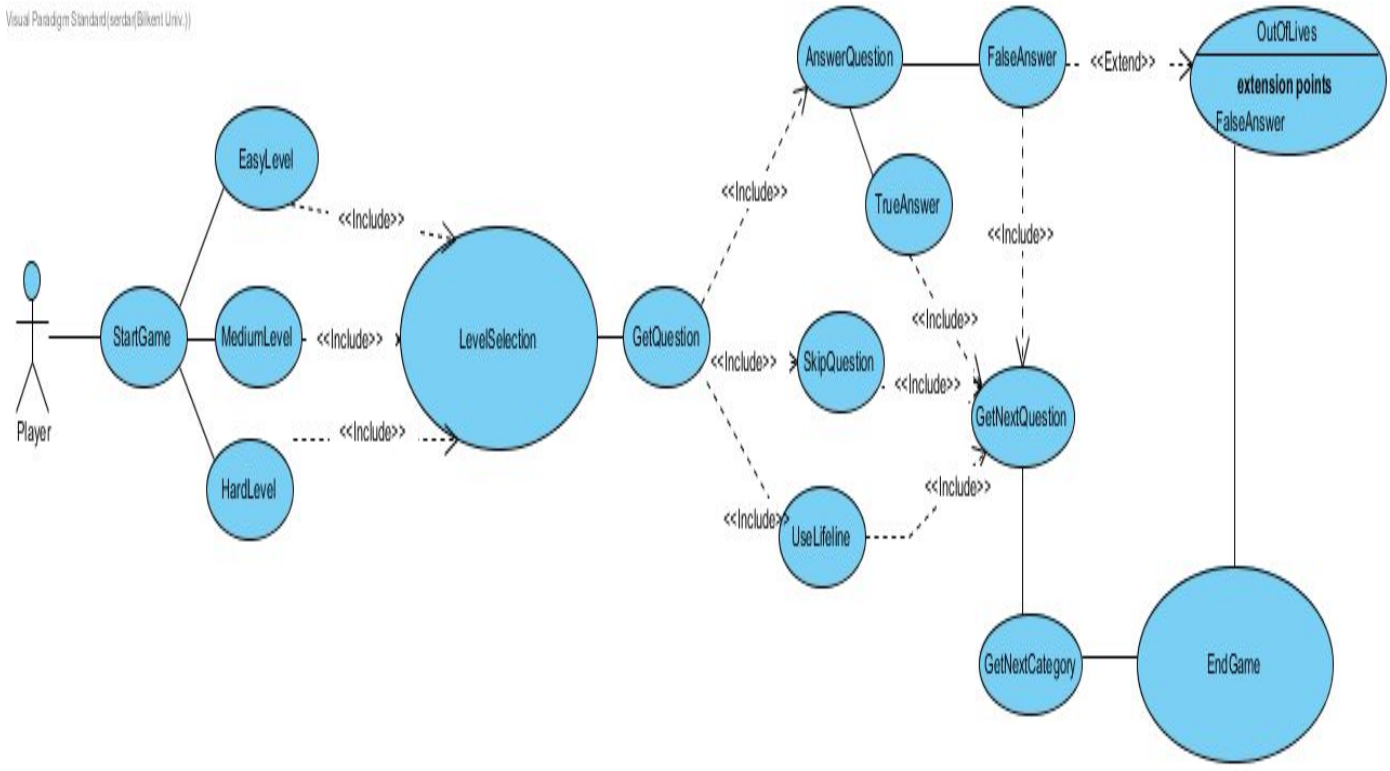
The game should include some challenge for the player to feel the competition air. There are so many elements that make it easier for the player. This is exemplified by the fact that the time and the right of passing questions are unlimited and there are three lifelines. Therefore, the game needs some difficulty. The fact that one false answer will cancel a correct answer, that only three lifelines are given during the entire game and that they are not given at the same time, that they are only used once, and that the categories will be given randomly will add difficulty to the game.

4.5 Extendibility

The game can be expanded so that other lifelines or bonuses can be added at any time, the time may be limited or the number of categories may be increased.

5. System Models

5.1 Use Case Model



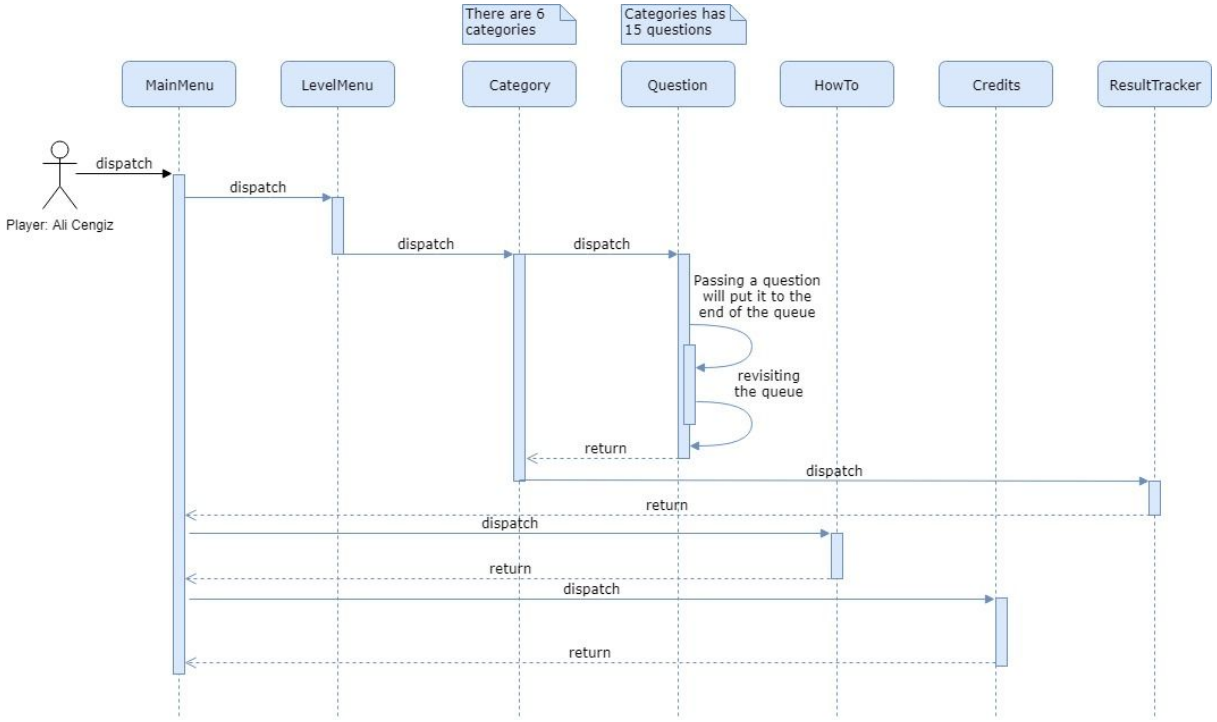
When the player opens the game, the game will first get the desired difficulty level from them. Then, the user will answer questions presented by the game. When presented with a question, user can answer the question, skip it or use lifeline. If the user answers the question and the answer is wrong, number of lives will be deducted. When the user runs out

of lives, the game will end. A new category is opened after all questions in the current category has been answered either correctly or incorrectly. If there is no category left, the game ends.

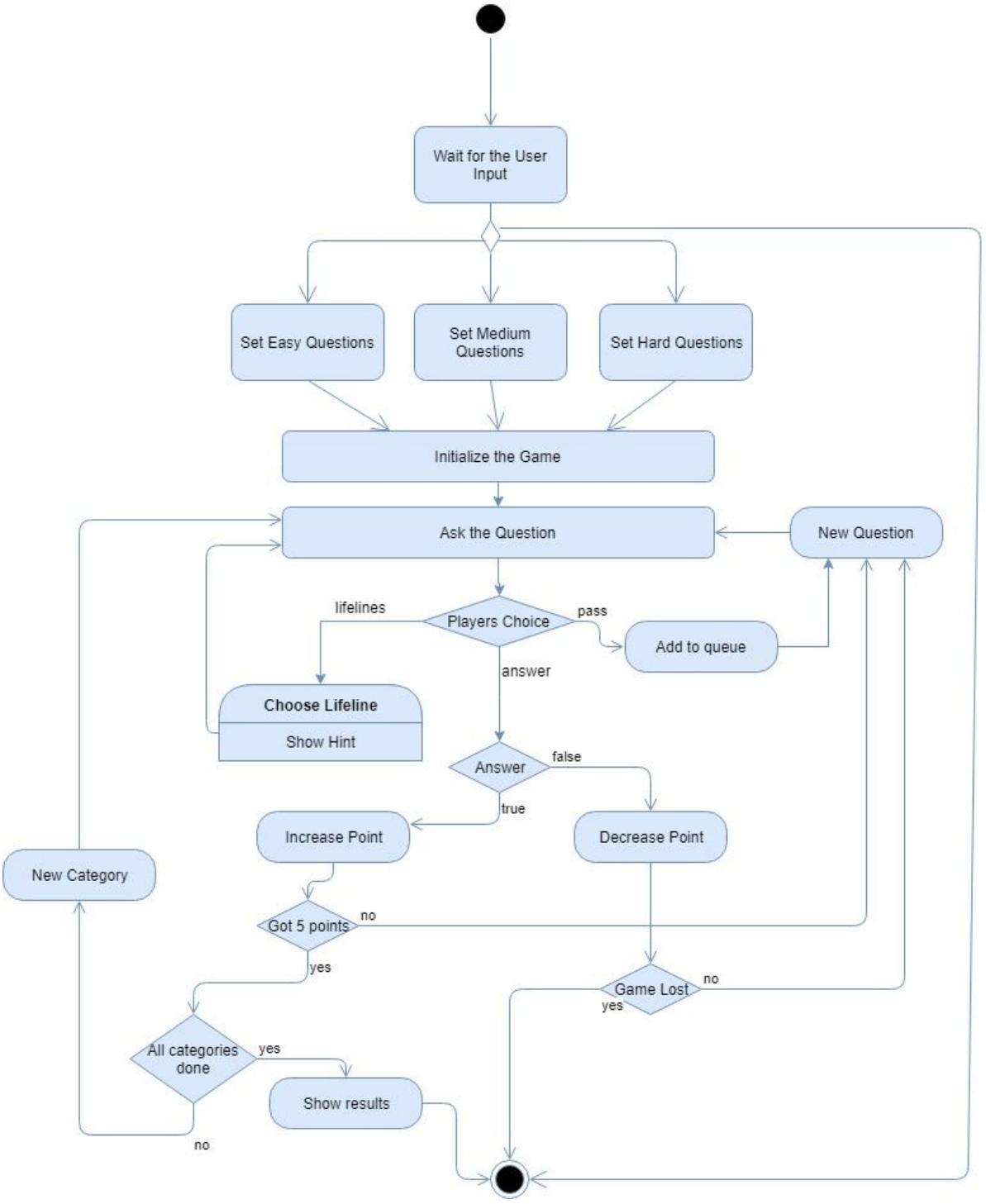
- LevelSelection: There are three difficulty levels for the game: easy, medium and hard. The player can choose any of them them.
- GetQuestion: The game presents a question to the player. At that point, user can answer the question, skip it or use lifeline.
- SkipQuestion: Skips the current question and gets a new one. The previous question will be added to the end of question queue and it will eventually be answered.

5.2 Dynamic Models

5.2.1 Sequence Diagram



5.2.2 Activity Diagram



The activity diagram above, illustrates how system runs the game.

At the beginning, the system waits the user to input. At this point, the user(player) may start the game, or exit it. When the user exits, system goes to the final node. If the user starts the game, the system waits the user for the level selection. For each selection, the next thing is setting selected level of the game. Then, the system initializes the game.

When the game starts, the system asks the question. Then, the game continues according to players choice such as using lifelines, pass or answering.

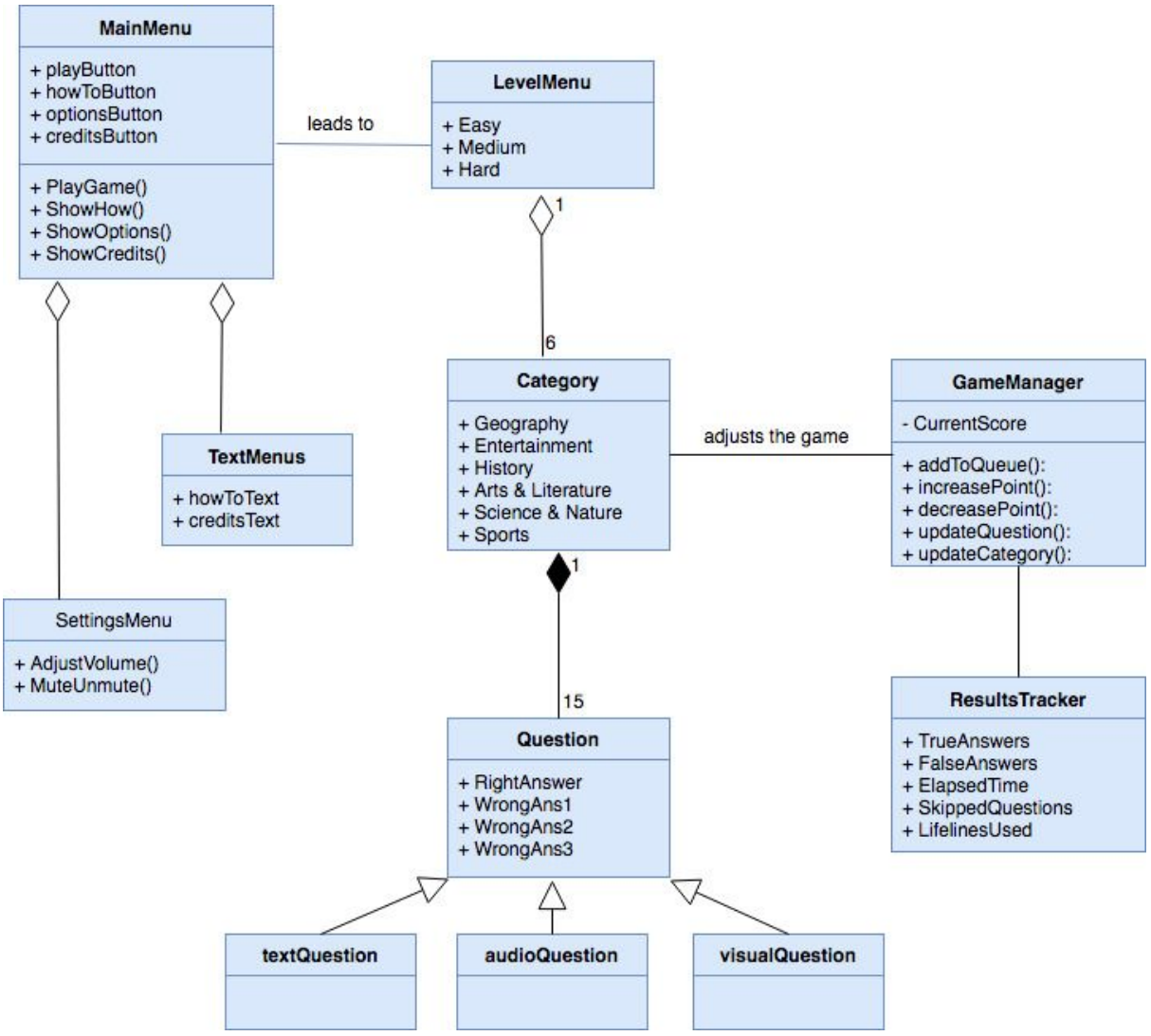
Case 1: If the user answer the question, the system checks whether it is true or not. If it is true, the system increases the player's point. It updates the questions till the 5 points. If the player's score is 5 points, the system updates the category automatically. If it is false, the system decreases the player's score point and updates the questions till the limit. If it exceeds the limit of false questions, the game is over and the system returns main menu to wait for user input.

Case 2: If the player wants to pass the question, the system adds this question to the queue and updates the question. Then, the next question comes. If the player wants to use lifeline, the system displays the result of the lifeline to the user and then the the player answers the question. Therefore, the process repeats this way.

Case 3: If the player wants to use lifeline, the system displays the result of the selected lifeline, then asks the question again. Therefore, the user may answer, pass or use another lifeline again.

Until all categories are done, the system updates the updated category. When all categories are done, the system displays the user's results.

5.3 Object and Class Model



MainMenu Class:

This class will be our main menu. The java program will start at this class. Purpose of this class is providing interface to user. Users can access settings, credits, how to play, and play game through this menu.

SettingsMenu Class:

This class is for keeping sound settings of game. This class will also provide setting interface for users. Users will be able to make adjustments using this menu.

TextMenus Class:

TextMenus will keep both credits and how to play menus. Depends on user's choice, this class whether provide credits menu or how to play menu. Both menu will contain text. Credits will keep credits menu and howTo will keep information about rules and gameplay of the Quiz Game.

LevelMenu Class:

In this menu, users will choose difficulty of the game.

Category Class:

This class will keep questions in different categories. It also determines the order of the categories in each game.

GameManager Class:

This class is going to manage the game. In each question, depends on the selection, this class will handle what will happen next. It also holds current score at a game.

ResultsTracker Class:

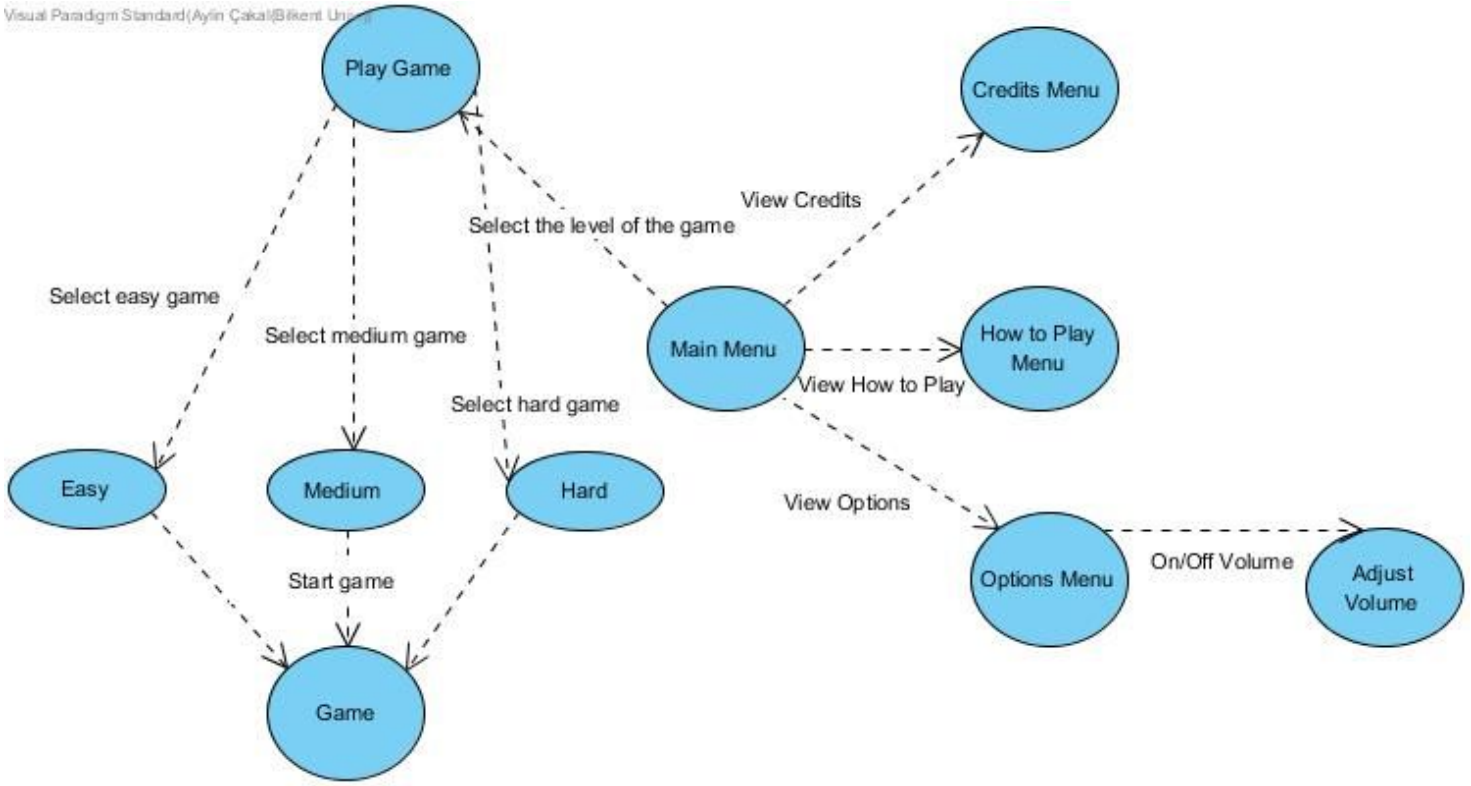
This class will track the game status. It will keep correct and wrong answers, elapsed time, unanswered questions and status of lifelines whether a lifeline is used or not.

Question Class:

This class is for keeping questions' contents, it's answer options and type of questions. Question types could be audio, visual or text based.

5.4 User Interface: Navigational Path and Screen Mock-ups

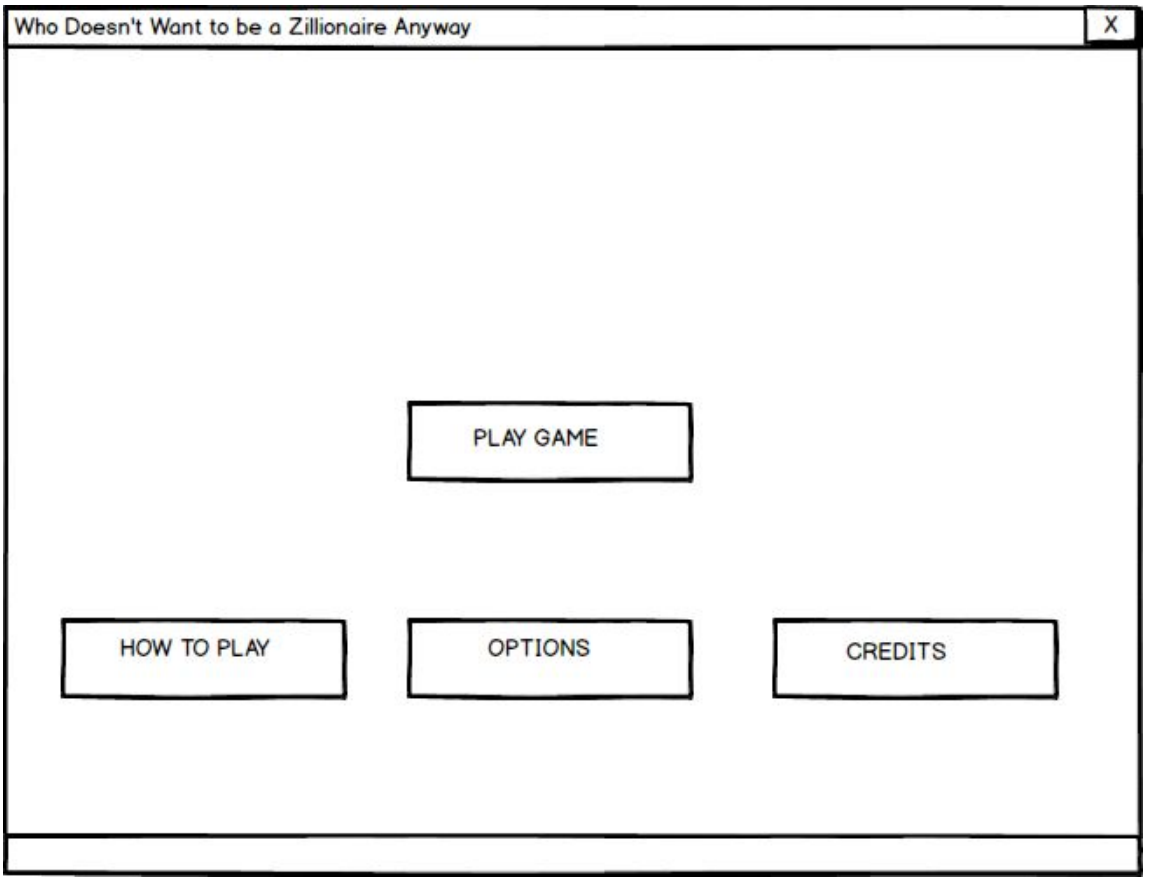
5.4.1 Navigational Path



5.4.2 Screen Mockups and Icons

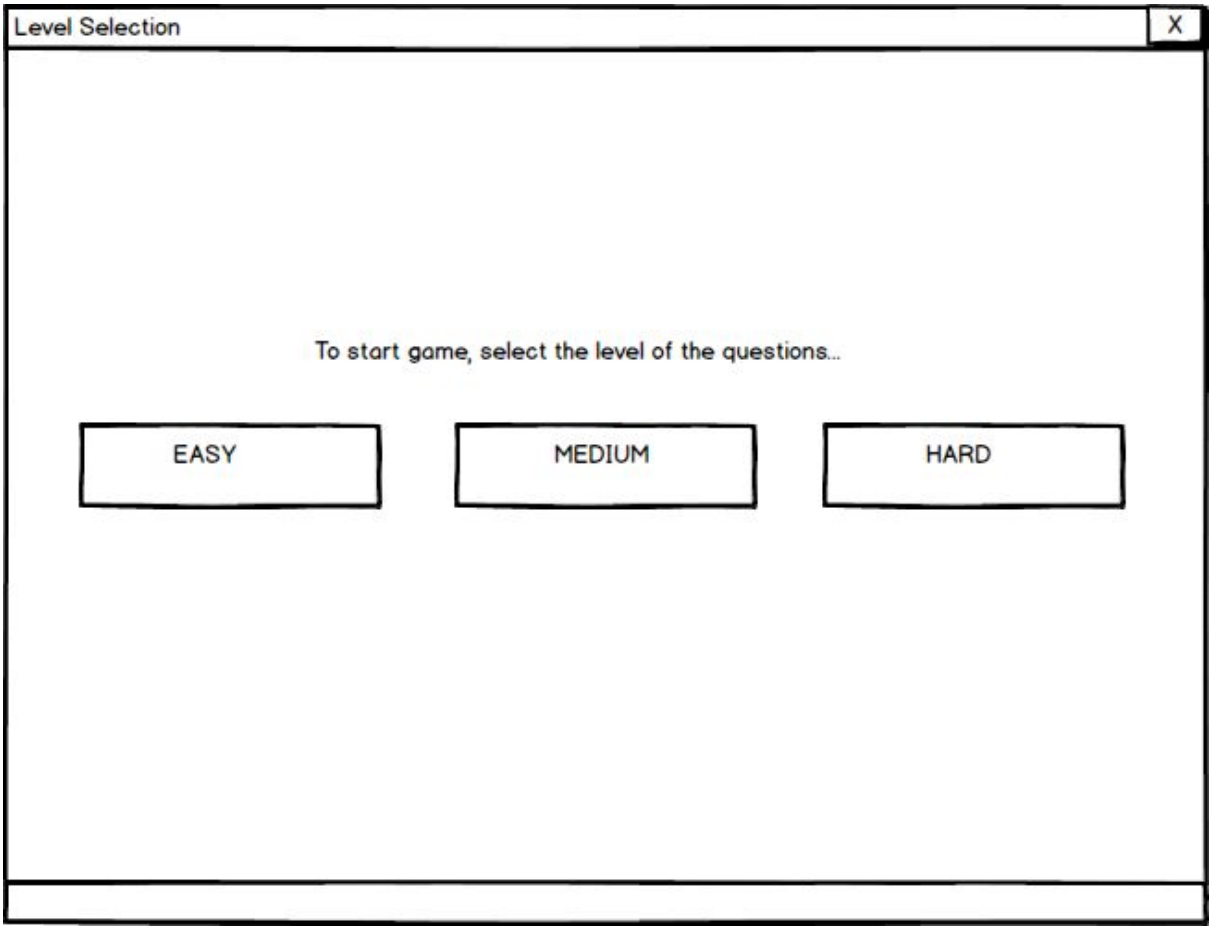
5.4.2.1 Main Menu Mockup Screen

The game will have simple game menu with a basic background.



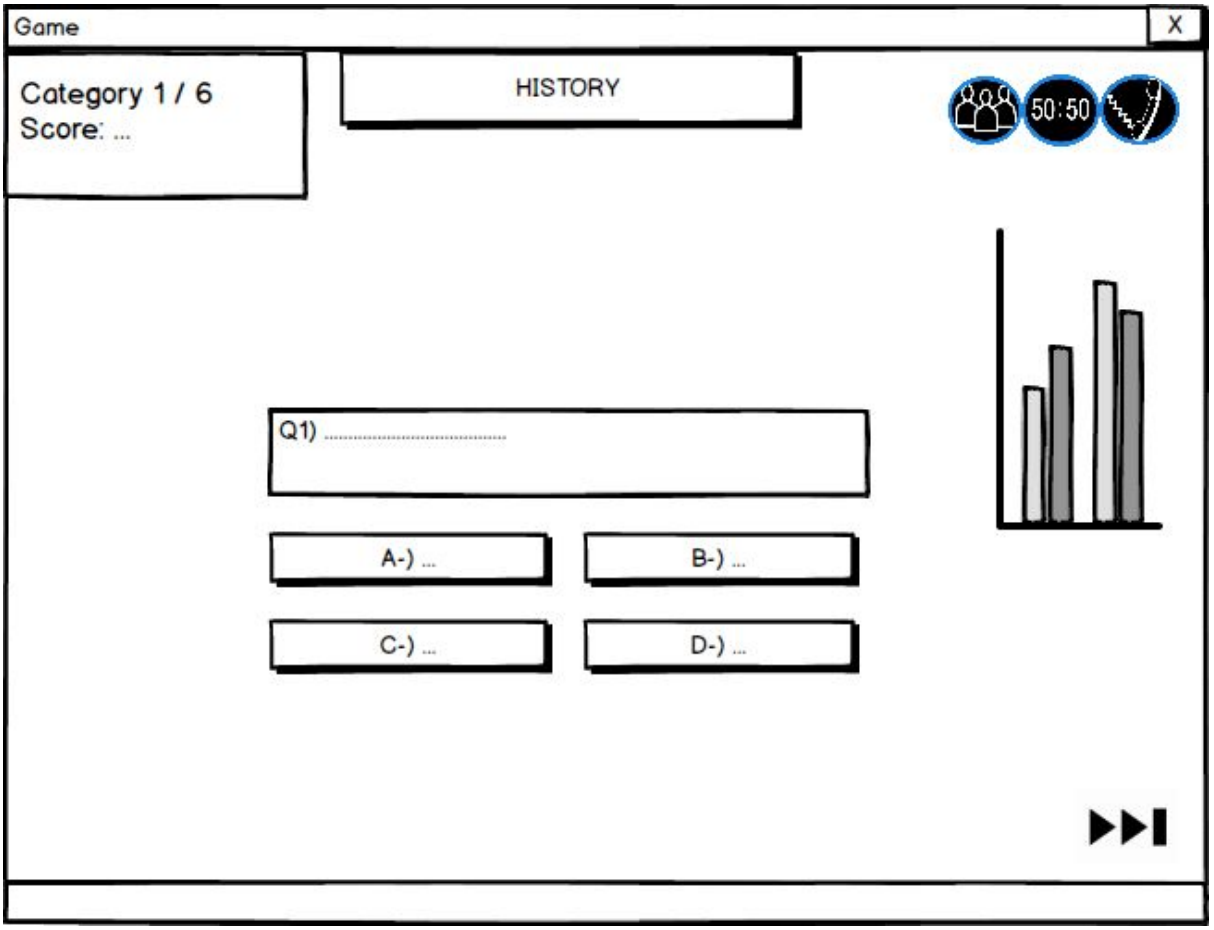
5.4.2.2 Level Selection Mockup Screen

This screen provides three options for level selection to the user.



5.4.2.3 Game Mockup Screen

Game screen includes the name of the current category at the top, the lifelines on the top right-hand corner, the score panel on the top left-hand corner, the questions and their answers at the center, the skip (pass) button at the right bottom, the result of the selected lifeline which is next to the question.

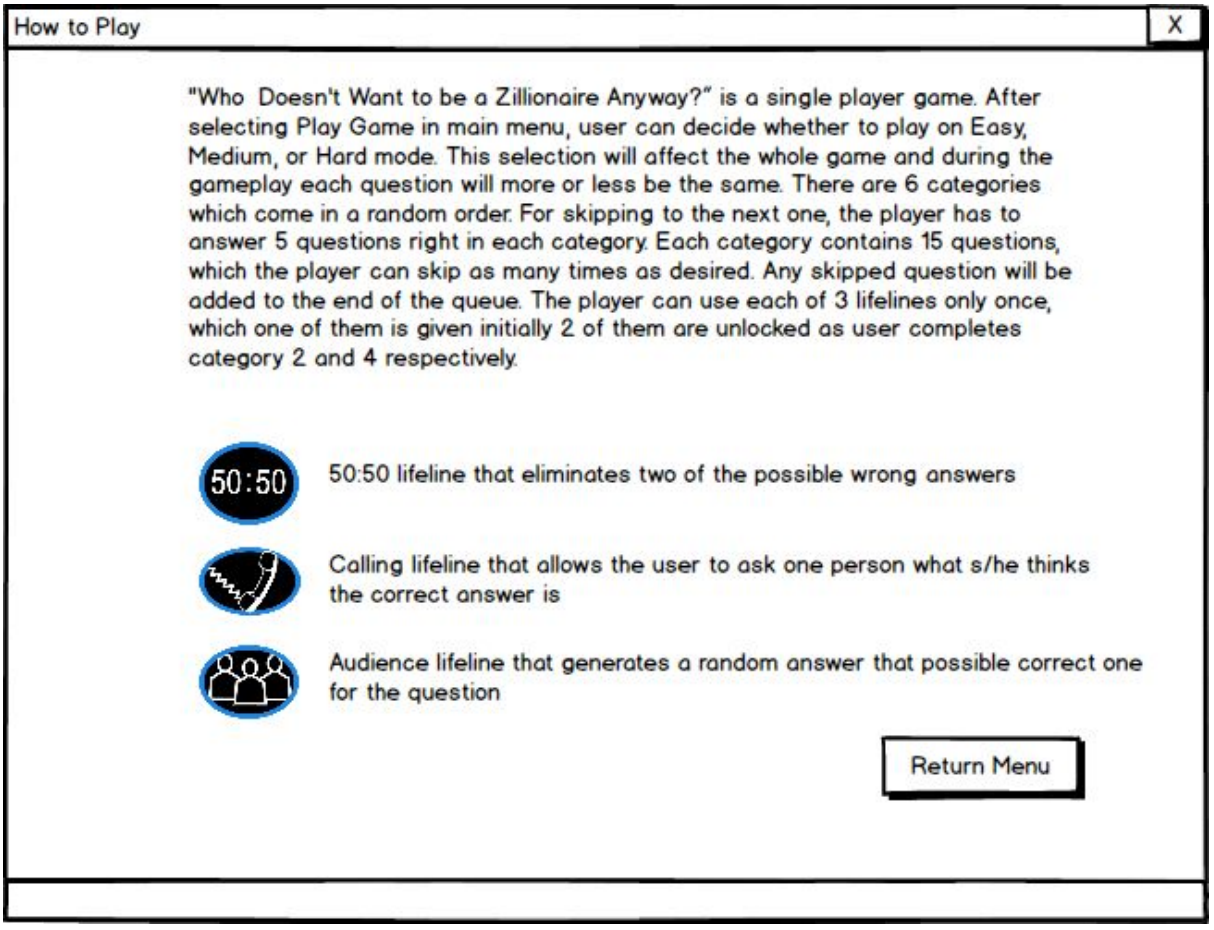


5.4.2.4 Results Panel Mockup Screen

This screen includes the number of true, false, skipped questions and lifelines used for entire game.It also display the elapsed time during the game.

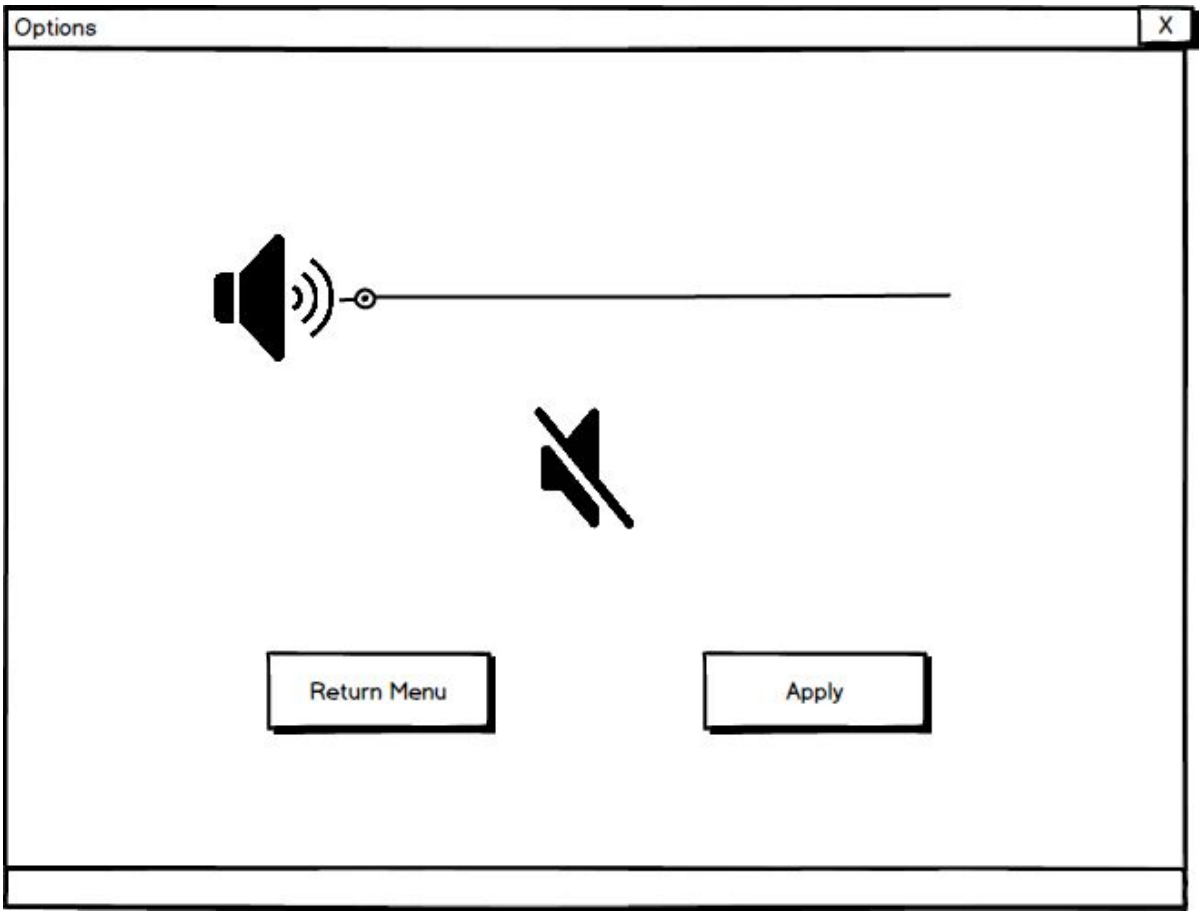
5.4.2.5 How to Play Mockup Screen

This screen provides the information about how to play “Who Doesn’t Want to be a Zillionaire Anyway?” and some icons of the game.



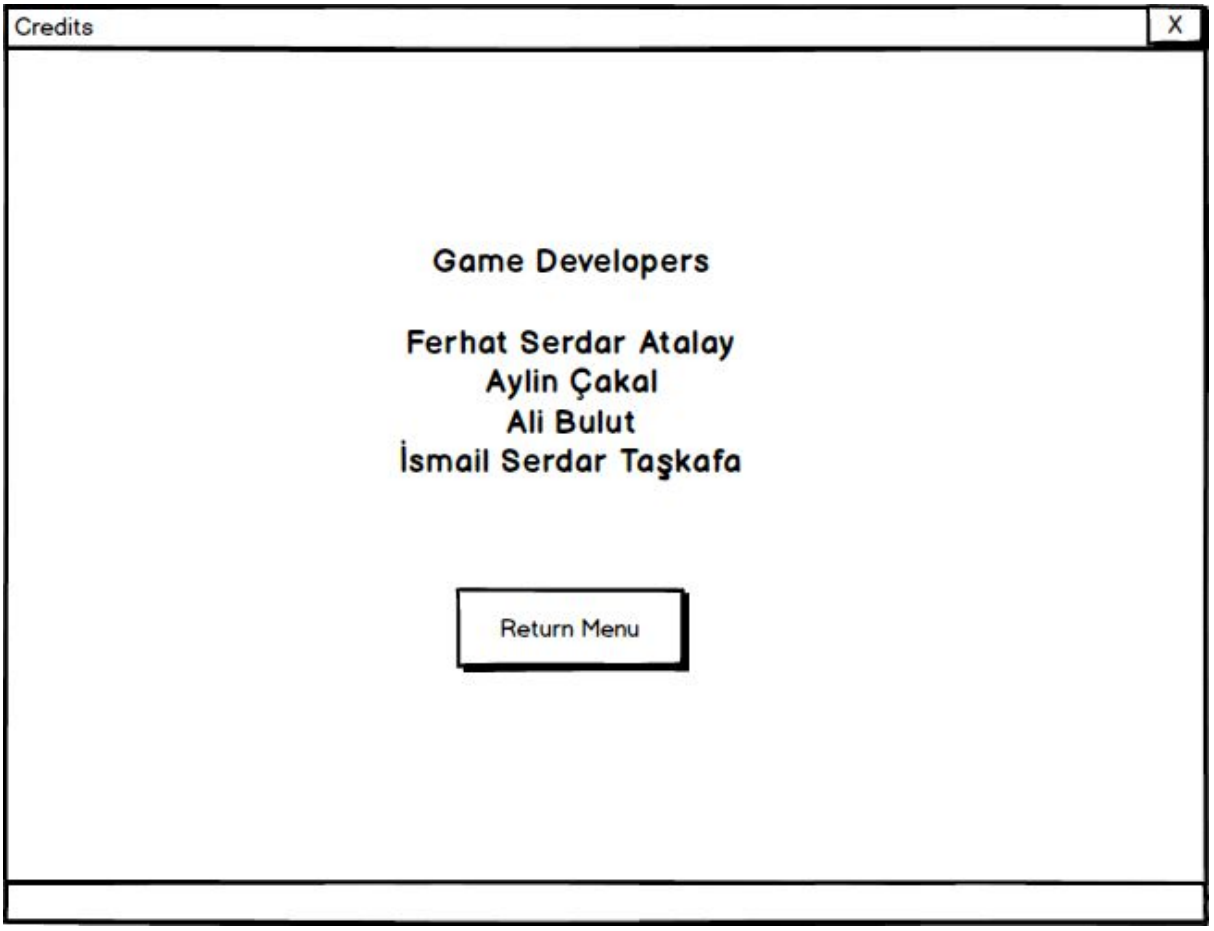
5.4.2.6 Options Mockup Screen

This screen allows the user to adjust the volume down or up and mute or unmute.



5.4.2.7 Credits Mockup Screen

This screen has information about this game developers.



6. Glossary & References

https://www.hasbro.com/common/documents/dad2af521c4311ddbd0b0800200c9a66/1EF487DA19B9F369103F2355AD9CC400.pdf
http://www.puffgames.com/whowantstobeamillionaire/