

Bilkent University

Department of Computer Engineering

**CS 319 Term Project**

Section 2

Group 2I

A-day-in-Bilkent

Analysis Report

Project Group Members

1-Enes Varol

2-Mahammad Askari Iqbal

3-Asaf Kağan Bezgin

4-Alper Kılıçaslan

5-Imran Hajiyev

Supervisor: Uğur Doğrusöz

**Contents**

1. **Introduction**

**2. Overview**

**2.1 Game Play**

**2.2 Characters**

**2.3 Companions**

**2.4 High score**

**2.5 Shop**

**2.6 Settings**

**2.7 Credits**

**3. Functional Requirements**

**3.1 Play Game**

**3.1.1 Single Player Game**

**3.1.2 Multiplayer Game**

**3.2 High Score**

**3.3 Help**

**3.4 Settings**

**3.5 Credits**

**3.6 Characters**

**3.7 Companions**

**3.8 Shop**

**4. Nonfunctional Requirements**

**4.1 Game Performance**

**4.2 User-Friendly Interface**

**4.3 Extendibility**

**4.4 Responsiveness**

**5. System Models**

**5.1 Use case model**

**5.2 Dynamic models**

**5.3 Object and class model**

**5.4 User interface and screen mock ups**

**6. Conclusion**

**7. Improvement summary**

**8. Glossary & references**

**1. Introduction**

A Day In Bilkent is an Arcade Style SHMUP - or Shoot'em Up – game where the player's main mission is to destroy the incoming enemies which are trying to do the same. For to be specific, the game is inspired by the combination of "Asteroids", "EverWing" & "Space Invaders" but in the contrary of these games, A Day In Bilkent has different types of items, powers, etc... and it has a Bilkent Theme as the name refers and enemies, bonuses, extras, etc... are going to be related with Bilkent University. The player is going to experience the campus life in an arcade style shooting game.

**2. Overview**

A Day in Bilkent is an Arcade Style SHMUP game such as Asteroids, EverWing, and Space Invaders. The gameplay is not complicated. Every age of gamers can easily play this game and have some fun. In the game, user plays as one of the Bilkent student, fighting with TAs, assignments, quizzes, and bosses, which are Teachers. Teachers are chosen according to the courses. Player will face lots of enemies and bosses until they die. After they die, player can check their GPA, which is high score, and try it again to get a better GPA.

**2.1 Gameplay**

A Day in Bilkent is 2D shoot'em up game, and player plays as a student. In the game, player and companions always shoot projectiles. Enemies are TAs, assignments, quizzes, teachers. Some of the enemies cannot shoot but if collision happens, they damage the player. Teachers are bosses and they have some attack patterns. Player's companions have also their own attack pattern and they can also damage the enemies. Power ups are available during the game. There are 5 power ups available. 2 of them makes the player disadvantageous. Power ups are shield, double bullet, rage mode, Allnighter, and Mayfest. Shield bonus gives you a temporary shield for the enemy attacks. Double bullet doubles the player's bullet frequency. Rage mode gives the player a temporary attack bonus and movement speed. Allnighter slows the player’s movement speed. Mayfest reduces attack damage, movement speed, and health for a short time. If player dies, game will end. There are also 2 game modes. Player can play single player mode or multiplayer mode. In single player mode, player will choose the game type; survival mode and story mode. In survival mode enemies will spawn infinitely so that player tries to get their score as high as they can. In story mode player will face 8 different bosses, and enjoy the context. In multiplayer mode, 2 players can play the game at the same time using same keyboard. Players can change character before the game in multiplayer mode.

**2.2 Characters**

There are 4 different characters available. Back Bencher has high health, Quick Learner has high speed, High Achiever has high attack damage and Default character where all the stats are average. Player can choose character before the game.

**2.3 Companions**

Companions are player like characters. They can shoot and damage the enemies. Player has 2 companions. Player can change the companions through Select Companion screen. Companions have their own skills that affect the gameplay.

**2.4 Highscore**

After the game ends, player can see the high score, which is GPA in A Day in Bilkent.

**2.5 Shop**

Player can buy some items to enhance their gameplay. Items that sold in the shop are three time use only. Player has to have enough coins to buy items. Available items are MIPS Green Card, Coffee, YemekSepeti, and Cheat Sheet. MIPS Green Card gives player an attack boost. Coffee gives speed. YemekSepeti gives health. Lastly Cheat Sheet increases all the stats.

**2.6 Settings**

Player can change the difficulty and volume, also player can turn on or off the music.

**2.7 Credits**

Player can see the credits through the credits screen.

**3. Functional Requirements**

**3.1 Game**

The player is trapped inside a container / box referred to as the "Class Room". The player spawns in the room and is surrounded by different enemies with different powers. These enemies attack the player and if the attack hits, the player’s life decreases. When life decreases to 0, the player dies and the game ends. The player is also aided with an array of different companions, each having their own special abilities. Collectables are also dropped by killing enemies, and they spawn randomly to give special powers to the player. The game cannot be paused at any point. This features is to eliminate any unfairness caused by some players pausing the game at crucial times to better understand the situation at hand. The game aims to increase reaction time and quick thinking of the player. The player starts a single player game with a Character selected from the Main Screen - Character option - and goes through levels, each representing a Semester of Bilkent University in story mode. The player scores points with number of enemies destroyed and time survived. The main objective is to defeat the boss or the "Professor" at the end to pass the level or "Semester". Objective of this game is to get the highest score or “GPA” possible. Upon losing, the player is brought back to the end game screen. Same goes for the survival mode. Only difference between story mode and survival mode is survival mode is endless. It pushes the limits of the players.

**3.2 High Score**

This section allows for the player to view their top 10 highest scores from the game runs. Each player saves their high scores with their initials and this allows a sense of competition between different players. Not only will this want to make play more to score more using updates, they will want to compete with their peers and excel more in the game. This section can only be found on the main menu.

**3.3 Settings**

The Settings section enables the user to select a wide range of options to customize the game for their own liking. Properties that will be changeable include:

a) Difficulty

b) Sound and Music Volumes

These customizations would allow for a more personal environment to the game and would provide the player ease to play in a way that he so desires. This section can only be operated using the main menu.

**3.4 Credits**

This section is accessed independently from the main menu. It is a shout out to all the people involved to make this game possible.

**3.5 Characters**

This section is a display of all the different Characters that are present for the player to choose from. The name of the character and a small bio of the character is visible for each character that is on display. Along with that, each character has an option to be "equipped" or chosen and that specific character will be used for the game by the player. Characters have certain stats that will be displayed on the bio. This section can only be accessed through the main screen.

**3.6 Companions**

This section is a display of all the different companions. Companions have their own power to help the player. Player can choose only two companions. This section can only be accessed through the main screen. In multiplayer mode, there are no companions.

﻿**4. Non-Functional Requirements**

**4.1 Game Performance**

The game mechanics aim to be fluid. All Game Objects that are movable including the character, companions, enemies, collectable etc's movements will be fluid in motion. They will be produced in a manner that shall conserve the games frame rates hence to reduce lagging of the game to a minimal. Sound as well be dealt with and shall be run on another thread hence to allow for quicker response to the player. The stimulus of sound and sight are crucial to the games success and thus shall be optimized to work effectively on even a Graphically Low end Computer. A smooth running game will entice the user to play more and have not rage quit on losing because of no fault of their own.

**4.2 User-Friendly Interface**

The Interface of the whole game will be simple and intuitive. The player will not have to work on guess work to operate the App. The help section which explains the player the mechanics of the game are coupled with pictures to allow the player to understand the mechanics in a glance. During the game run, the position of player health, score and other details will be positioned in such a manner that it shall not interfere with the main game. Hence, the player will not be distracted from these vital information blocking the visuals of other game objects.

**4.3 Extendibility**

The characters can be used to enhance the game. Characters allow us to improve a new game mode, which is Story mode. Due to the theme chosen for the game, a lot more depth can be given to it in future releases. This depth can be:

a) An enhanced story line

b) Story lines for Other departments of Bilkent

c) Different types of Collectables, Enemies, Companions etc.

d) Increase in weapons for all enemies.

e) More Shooting patterns for bosses.

f) Different powers of Companions and Characters.

With progress of releases and input from players, we can modify the game to add the suggested features and more in future releases while keeping the game mechanics intact. With our design, different game modes can also be introduced like:

a) Multiplayer

b) Time Attacks

c) Survival Mode

d) Story Mode

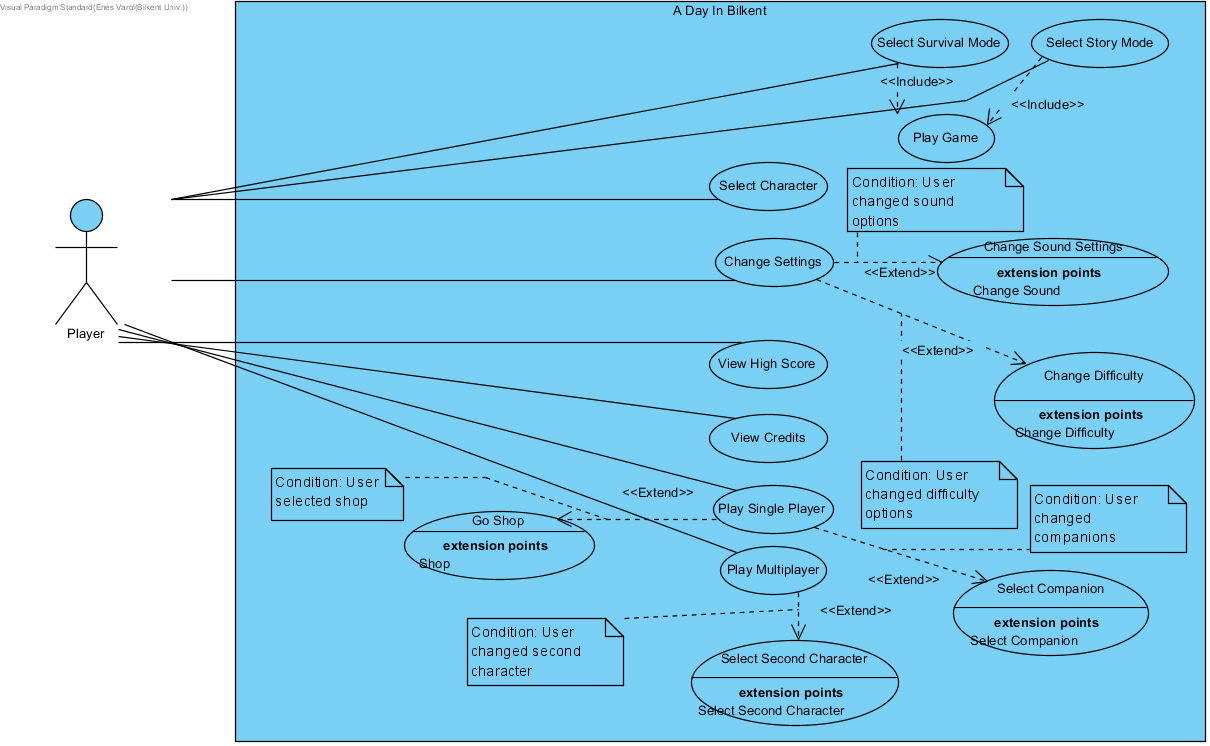
The responses from the playing community shall guide in the future extendibility of the game. All of this is a result of the theme of the game and the design of the game to divide all features as a separate and independent component.

**4.5 Responsiveness**

The game loop that will be designed - the heart of the whole game - will be such that shall maintain a steady frame rate and shall not be unresponsive. It shall be designed in such a manner that during gameplay, the player will not have to face any disadvantages because of a lag. The buttons for the main menu and other sections that have User Input shall be designed in such a way that they will be fluid in operations and not become jittery whilst in use. This serves the purpose to reduce any cause of frustrations for the player so they can have a relaxing experience.

**5. System Models**

**5.1 Use case model**



**Use Case: Select Survival Mode**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects survival mode.
* System creates the game and starts it.

**Pre-conditions:** Player must be in main menu.

**Post-conditions:**

**Entry conditions:** Player should select "Play Survival" from screen.

**Exit conditions:**

1. Player and his friend finishes the story mode.
2. Player and his friend can’t finish the story mode.

**Success Scenario Event Flow:**

1. Player selects "Play Survival" from game screen.

2. System starts the game.

3.Player moves the character via keyboard.

4.System updates character’s view.

5. Player’s friend moves the character via keyboard.

6. System updates character’s view.

7.Characters shoots.

8.System creates bullets and updates it.

9.Bullets hit an enemy.

10.System deallocates bullet and enemy and removes from screen.

11.Player kills all enemies.

12.System finishes the first wave and initializes the next wave.

**Alternative Event Flow:**

1. If Player want to return to main menu:

a. Player selects "Go Back" from screen.

**Use Case: Select Story Mode**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects play story mode.
* System creates the game and starts it.

**Pre-conditions:** Player must be in main menu.

**Post-conditions:**

**Entry conditions:** Player should select "Play Story " from screen.

**Exit conditions:**

1. Player finishes the story mode.
2. Player can’t finish the story mode.

**Success Scenario Event Flow:**

1. Player selects “Play Story” from screen.

2. System starts the game.

3.Player moves the character and companions via keyboard.

4.System updates character and companions’ view.

5.Character and companions shoots.

6.System creates bullets and updates it.

7.Bullets hit an enemy.

8.System deallocates bullet and enemy and removes from screen.

9.Player kills all enemies.

10.System finishes the first level and initializes the next level.

**Alternative Event Flow:**

1. If character dies:

a. System shows “Game Over” screen and goes back to main menu.

**Use Case: Select Character**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects select character.
* System creates the select character screen and displays it.

**Pre-conditions:** Player must be in main menu.

**Post-conditions:**

**Entry conditions:** Player should select "Select Character" from screen.

**Exit conditions:**

1. Player selects “Go Back” from screen.

**Success Scenario Event Flow:**

1. Player selects “Select Character” from screen.

2. System creates the select character screen and displays it.

3.Player selects his character.

4.System updates character information.

**Alternative Event Flow:**

1. If Player wants to go back to main menu:

a. Player selects “Go Back” from screen.

b. System creates main menu and displays it.

**Use Case: Change Settings**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects settings.
* System creates the settings screen and displays it.

**Pre-conditions:** Player must be in main menu.

**Post-conditions:**

**Entry conditions:** Player should select "Settings" from screen.

**Exit conditions:**

1.Player selects “Go Back” from screen.

**Success Scenario Event Flow:**

1. Player selects “Settings” from screen.

2. System creates the settings screen and displays it.

3.Player changes sound and difficulty settings.

4.System updates sound and difficulty information.

**Alternative Event Flow:**

1. If Player wants to go back to main menu:

a. Player selects “Go Back” from screen.

b. System creates main menu and displays it.

**Use Case: View High Score**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects high scores.
* System creates the high scores screen and displays it.

**Pre-conditions:** Player must be in main menu.

**Post-conditions:**

**Entry conditions:** Player should select "High Scores" from screen.

**Exit conditions:**

1.Player selects “Go Back” from screen.

**Success Scenario Event Flow:**

1. Player selects “High Scores” from screen.

2. System gets high score information.

3. System creates the high scores screen and displays it.

**Alternative Event Flow:**

1. If Player wants to go back to main menu:

a. Player selects “Go Back” from screen.

b. System creates main menu and displays it.

**Use Case: View Credits**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects view credits.
* System creates the credits screen and displays it.

**Pre-conditions:** Player must be in main menu.

**Post-conditions:**

**Entry conditions:** Player should select "Credits" from screen.

**Exit conditions:**

1.Player selects “Go Back” from screen.

**Success Scenario Event Flow:**

1. Player selects “Credits” from screen.

2. System creates the credits screen and displays it.

**Alternative Event Flow:**

1. If Player wants to go back to main menu:

a. Player selects “Go Back” from screen.

b. System creates main menu and displays it.

**Use Case: Play Single Player**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects single player mode.
* System updates the main menu components and shows on the screen.

**Pre-conditions:** Player must open the game from his computer.

**Post-conditions:**

**Entry conditions:** Player should select "Single Player" from screen.

**Exit conditions:**

1. Player selects "Multiplayer" from screen.
2. Player quits the game.

**Success Scenario Event Flow:**

1. Player selects "Single Player" from game screen.

2. System updates the main menu components and displays it on screen.

**Alternative Event Flow:**

1. If Player want to return to multiplayer:

a. Player selects "Multiplayer" from screen.

**Use Case: Play Multiplayer**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects multiplayer mode.
* System updates the main menu components and shows on the screen.

**Pre-conditions:** Player must open the game from his computer.

**Post-conditions:**

**Entry conditions:** Player should select "Multi Player" from screen.

**Exit conditions:**

1.Player selects "Single Player" from screen.

2.Player quits the game.

**Success Scenario Event Flow:**

1. Player selects "Multiplayer" from game screen.

2. System updates the main menu components and displays it on screen.

**Alternative Event Flow:**

1. If Player want to return to multiplayer:

a. Player selects "Single Player" from screen.

**Use Case: Select Second Character**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects select second character.
* System creates the select second character screen and displays it.

**Pre-conditions:** Player must select Multiplayer mode.

**Post-conditions:**

**Entry conditions:** Player should select "Select Second Character" from screen.

**Exit conditions:**

**1.**Player selects “Go Back” from screen.

**Success Scenario Event Flow:**

1. Player selects "Multiplayer" from game screen.

2. System updates the main menu components and displays it on screen.

3. Player selects “Select Second Character” from screen.

4. System creates the select second character screen and displays it.

5.Player selects second character.

6.System updates second character information.

**Alternative Event Flow:**

1. If Player wants to go back to main menu:

a. Player selects “Go Back” from screen.

b. System creates main menu and displays it.

**Use Case: Select Companion**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects select companion.
* System creates the select companion screen and displays it.

**Pre-conditions:** Player must select Single Player mode.

**Post-conditions:**

**Entry conditions:** Player should select "Select Companion" from screen.

**Exit conditions:**

1. Player selects “Go Back” from screen.

**Success Scenario Event Flow:**

1. Player selects “Single Player” from game screen.

2. System updates the main menu components and displays it on screen.

3. Player selects “Select Companion” from screen.

4. System creates the select companion screen and displays it.

5.Player selects his companions.

6.System updates companion information.

**Alternative Event Flow:**

1. If Player wants to go back to main menu:

a. Player selects “Go Back” from screen.

b. System creates main menu and displays it.

**Use Case: Go Shop**

**Primary Actor**: Player

**Stakeholders and Interests:**

* Player selects shop.
* System creates the shop screen and displays it.

**Pre-conditions:** Player must select Single Player mode.

**Post-conditions:**

**Entry conditions:** Player should select "Shop" from screen.

**Exit conditions:**

1.Player selects “Go Back” from screen.

**Success Scenario Event Flow:**

1. Player selects “Single Player” from main menu.

2. System updates the main menu components and displays it on screen.

3. Player selects “Shop” from screen.

4. System creates the shop screen and displays it.

5.Player buys items.

6.System updates currency and item information.

**Alternative Event Flow:**

1. If Player wants to go back to main menu:

a. Player selects “Go Back” from screen.

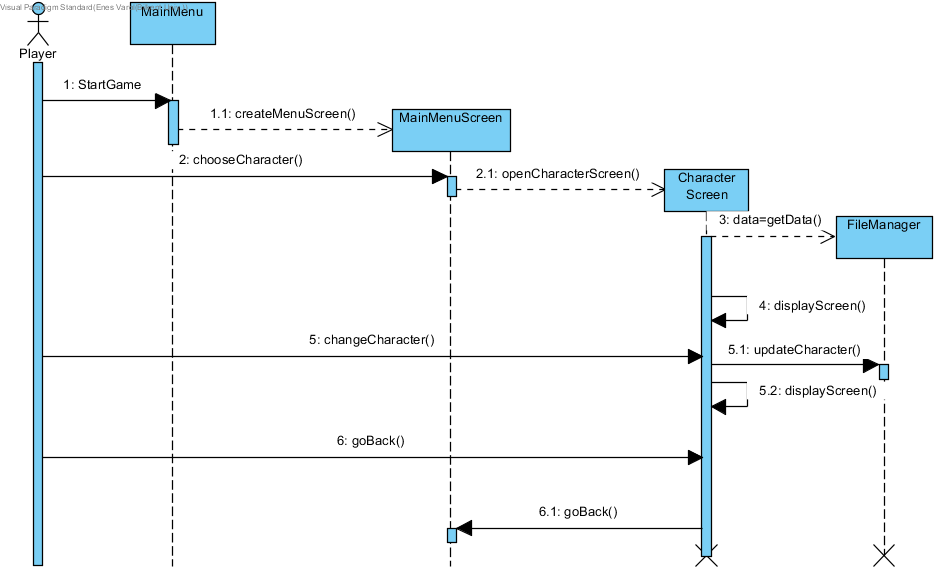
b. System creates main menu and displays it.

**5.2 Dynamic models**

**Select Character**

**Scenario:** Player wants to select character.

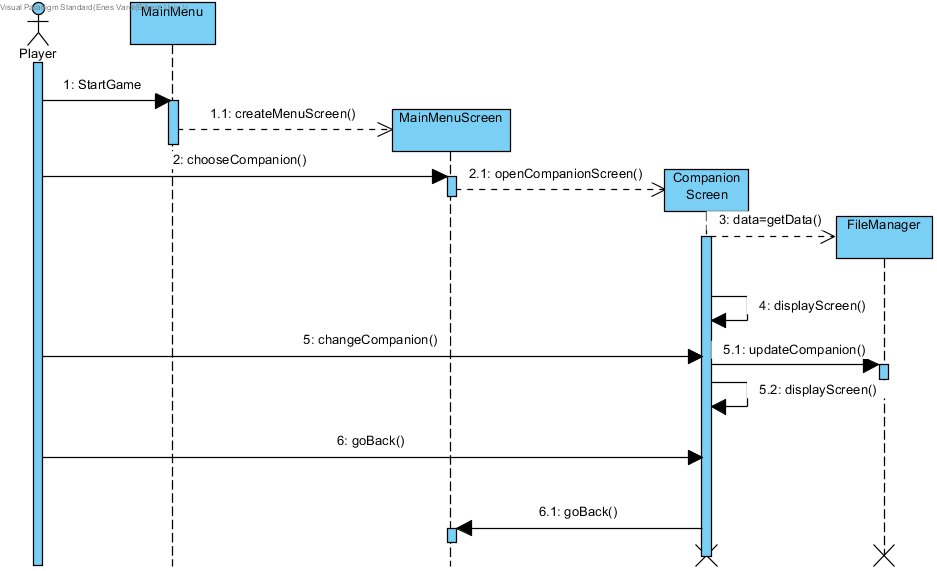
Player wants to select character, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player clicks select character button from MainMenuScreen and chooseCharacter() method will be called. MainMenu screen initializes CharacterScreen via openCharacterScreen(). CharacterScreen gets data from FileManager(data=getData()) after that runs the displayScreen() method and shows it on screen. If player changes the character, then CharacterScreen sends the info with updateCharacter () method to the FileManager. If player decides to go back goBack() method will be called and CharacterScreen goes back to MainMenuScreen.



**Select Companion**

**Scenario:** Player wants to select companion.

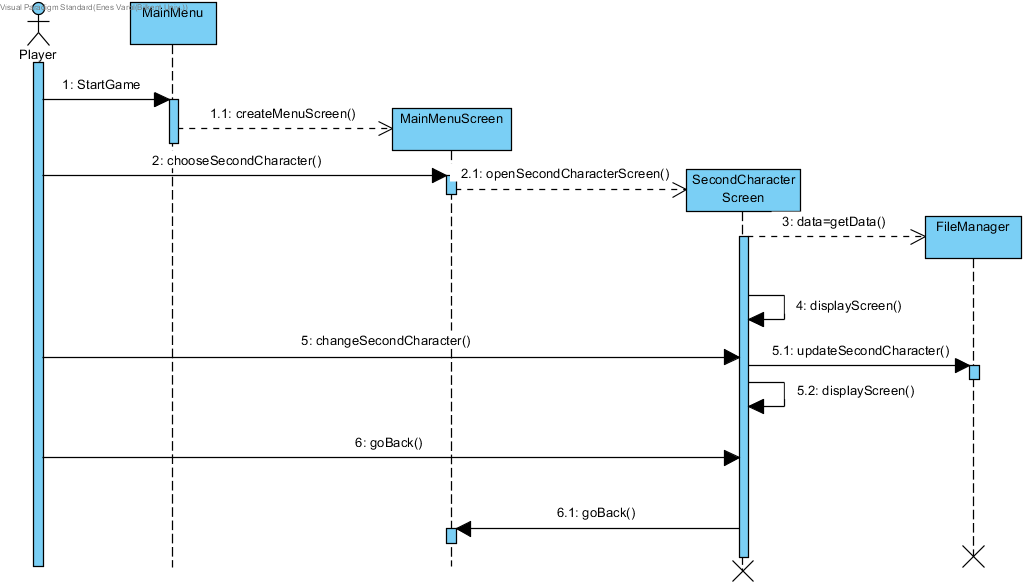
Player wants to select companion, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player clicks select companion button from MainMenuScreen and chooseCompanion() method will be called. MainMenu screen initializes CompanionScreen via openCompanionScreen(). CompanionScreen gets data from FileManager(data=getData()) after that runs the displayScreen() method and shows it on screen. If player changes the companion, then CompanionScreen sends the info with updateCompanion() method to the FileManager. If player decides to go back goBack() method will be called and CompanionScreen goes back to MainMenuScreen.

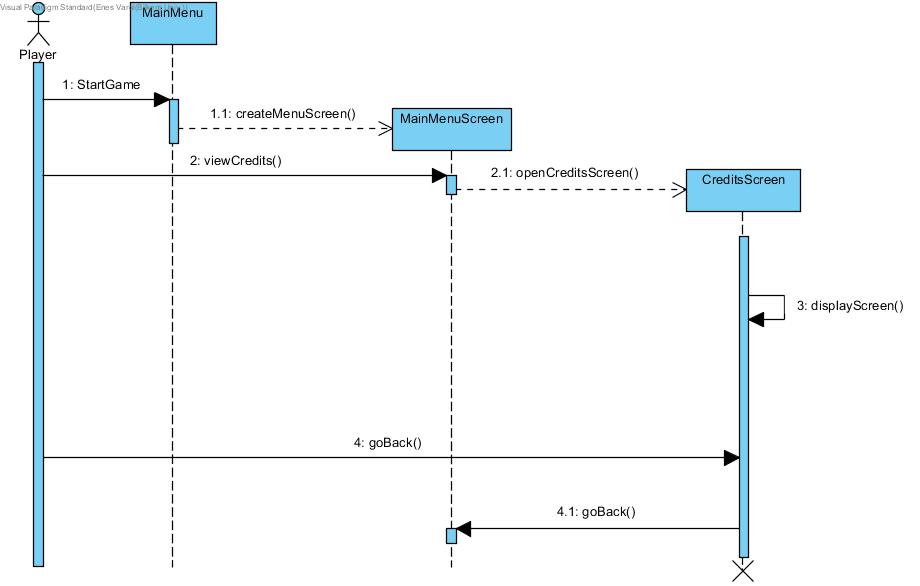


**Select Second Character**

**Scenario:** Player wants to select second character.

Player wants to select second character, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player clicks select second character button from MainMenuScreen and chooseSecondCharacter() method will be called. MainMenuScreen initializes SecondCharacterScreen via openSecondCharacterScreen(). SecondCharacterScreen gets data from FileManager(data=getData()) after that runs the displayScreen() method and shows it on screen. If player changes the companion, then SecondCharacterScreen sends the info with updateSecondCharacter () method to the FileManager. If player decides to go back goBack() method will be called and SecondCharacterScreen goes back to MainMenuScreen.

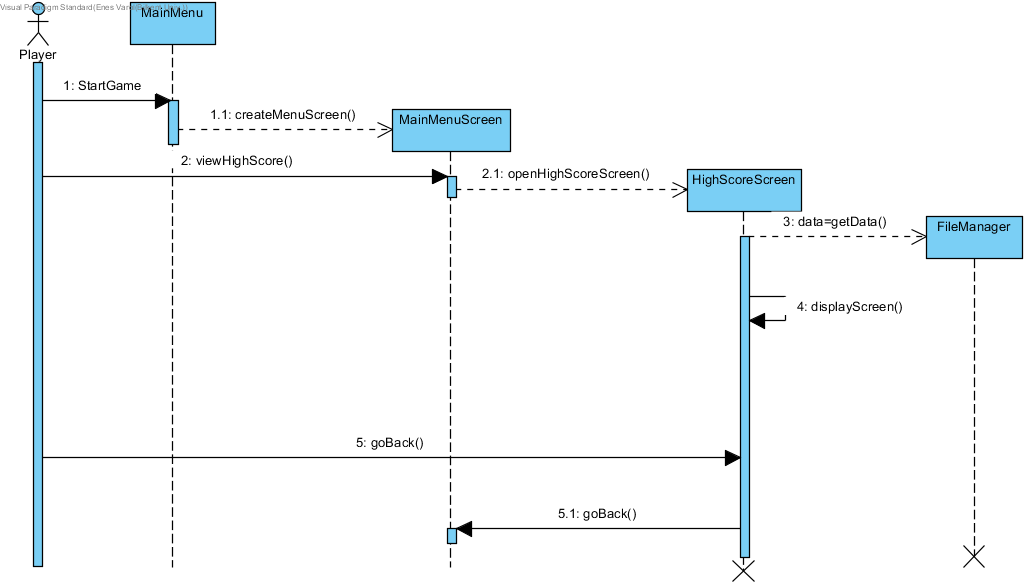


**View Credits**

**Scenario:** Player wants to see the credits.

Player wants to see the credits, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player clicks credits button from MainMenuScreen and viewCredits() method will be called. MainMenu screen initializes CreditsScreen via openCreditsScreen(). CreditsScreen runs the displayScreen() method and shows it on screen. If player decides to go back goBack() method will be called and CreditsScreen goes back to MainMenuScreen.

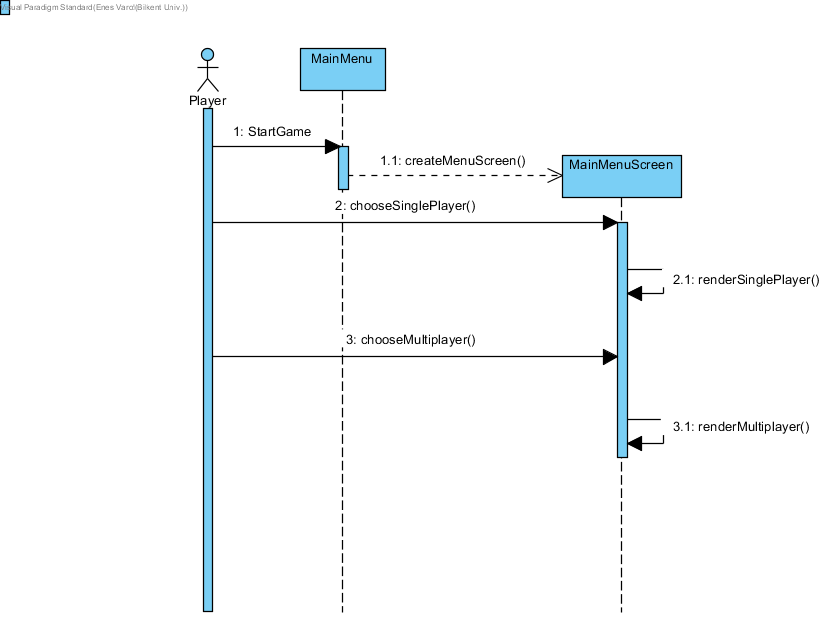
**View High Score**

****

**Scenario**

Player wants to see highscores, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player clicks Highscores button from MainMenuScreen and viewHighScore() method will be called. MainMenu screen initializes HighScoreScreen via openHighScoreScreen(). HighScoreScreen gets data from FileManager(data=getData()) after that runs the displayScreen() method and shows it on screen. If player decides to go back goBack() method will be called and HighScoreScreen goes back to MainMenuScreen.

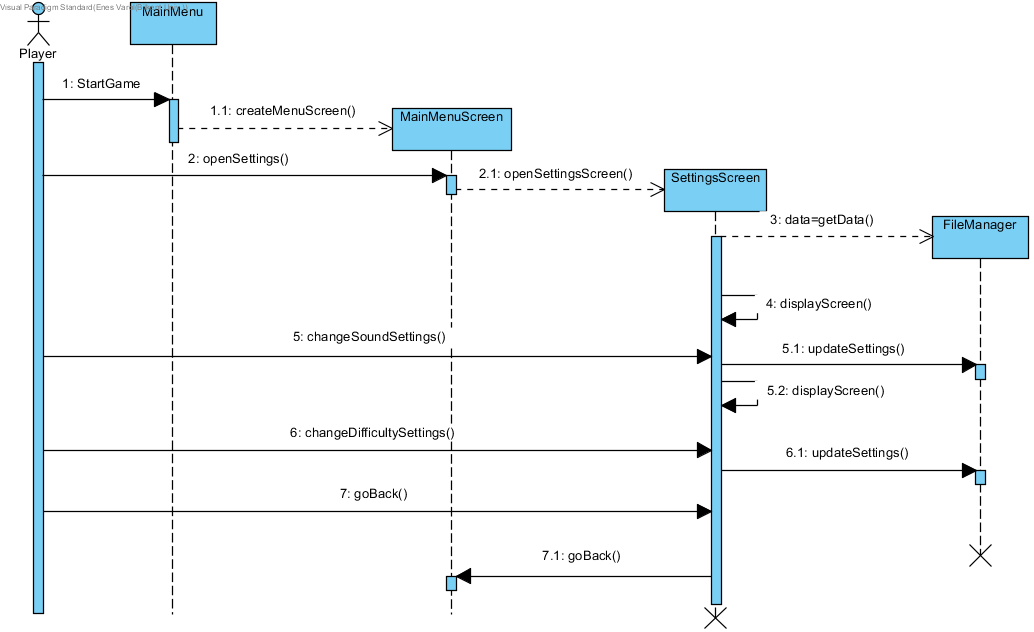
**Choose Game Mode**

****

**Scenario**

Player wants to choose a game mode, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player clicks single player button from MainMenuScreen and screen runs renderSinglePlayer() method via chooseSinglePlayer(). Player clicks multiplayer button from MainMenuScreen and screen runs renderMultiplayer() method via chooseMultiplayer().

**Change Settings**

****

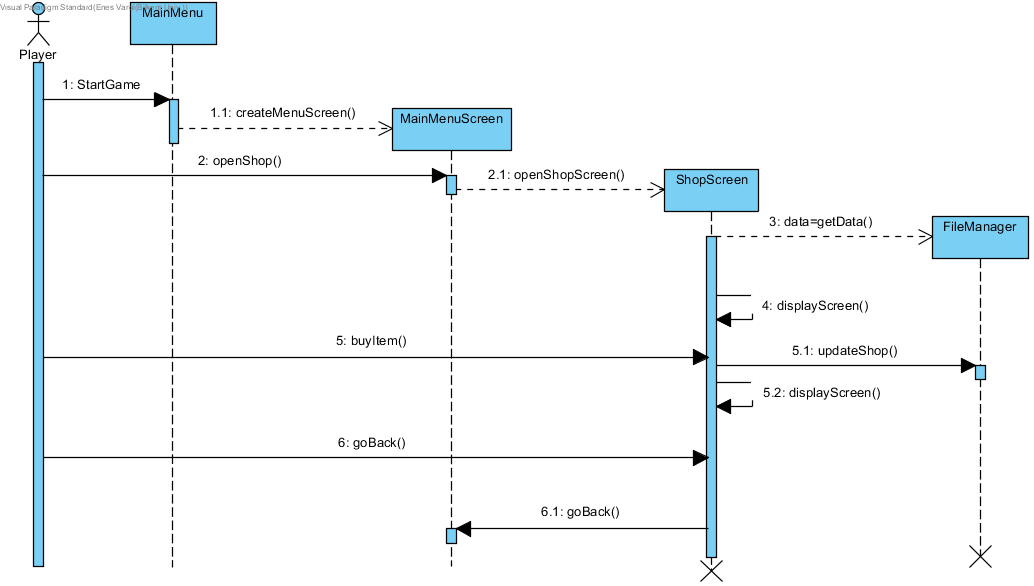
**Scenario:** Player wants to change the settings

Player wants to change settings, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player clicks settings button from MainMenuScreen and openSettings() method will be called. MainMenu screen initializes SettingsScreen via openSettingsScreen(). SettingsScreen gets data from FileManager(data=getData()) after that runs the displayScreen() method and shows it on screen. If player changes the sound level or the music level, then SettingsScreen sends the information with updateSettings() method to the FileManager. If player changes the difficulty, then SettingsScreen sends the information with updateSettings() method to the FileManager. If player decides to go back goBack() method will be called and SettingsScreen goes back to MainMenuScreen.

**Open Shop**

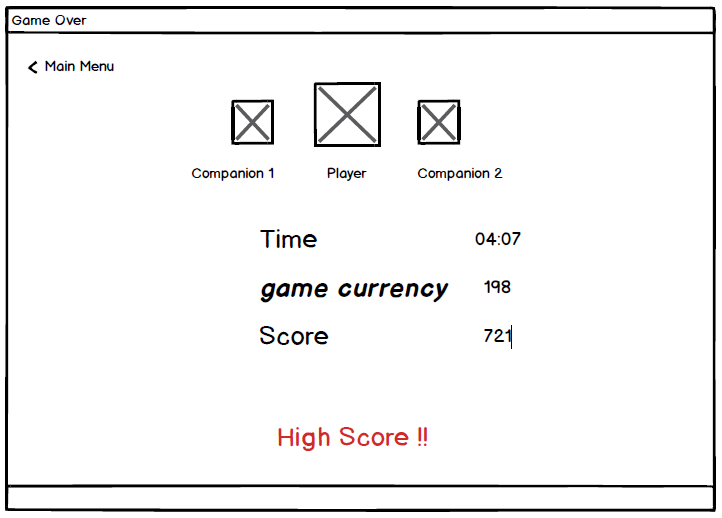
**Scenario:** Player wants to open shop and buy items.

Player wants to open shop and buy items, he/she starts to game. MainMenu initializes MainMenuScreen via createMenuScreen() method. Player presses the proper button from main menu in order to enter shop page and openShop() method will be called. MainMenuScreen initializes the ShopScreen via openShopScreen(). ShopScreen gets data from FileManager(data=getDATA()) and displays it on the screen with displayScreen() method where player has different options to buy an item for improving the attributes of his/her character during the game if item is equipped. Items in the shop is being bought by the game currency, via buyItem() method and the information sends to FileManager with updateShop() method after that ShopScreen updates itself with displayScreen() method. Initially the player has no items, by playing and gaining game currency, player might decide whether buying an item or not. Game currency is also saved in the file manager therefore, when user tries to buy an item which does not match with the balance of the player, then the system does not allow player to buy that item. Player has also an option to only look for items and go back without buying any, in that case if the player presses go back button, goBack() method will be called and MainMenuScreen initializes.

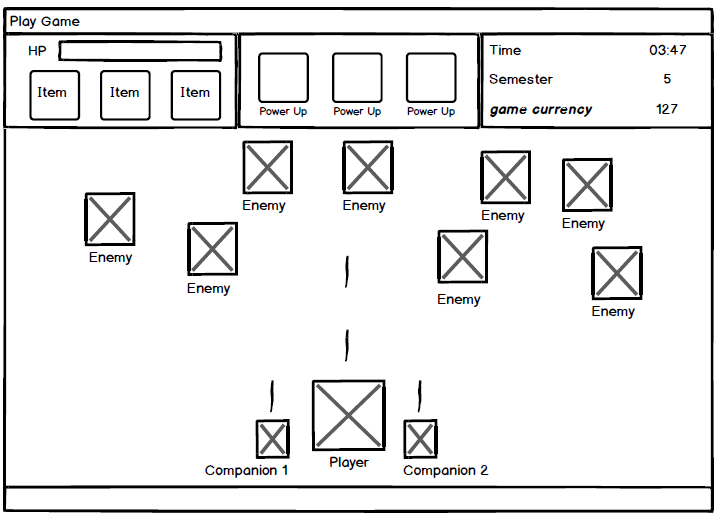


**5.4 Screen Mock-ups**

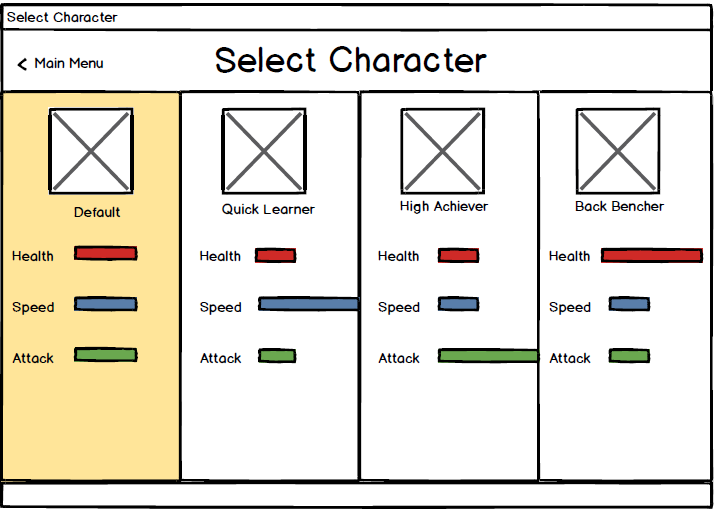
**Game Over**



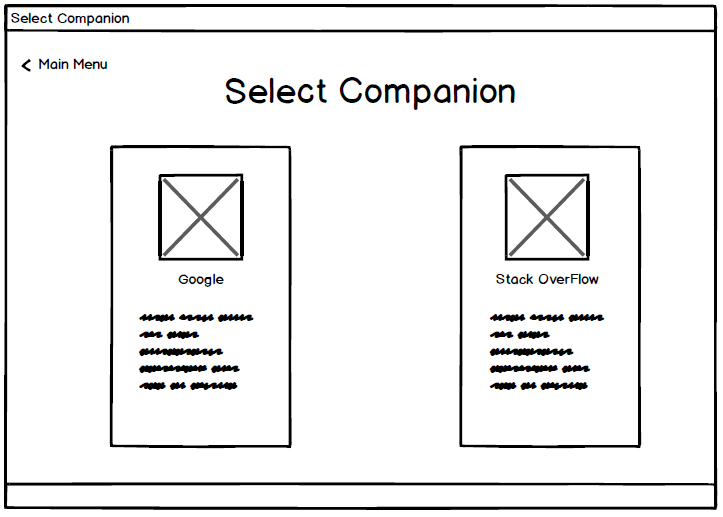
**Play Game**



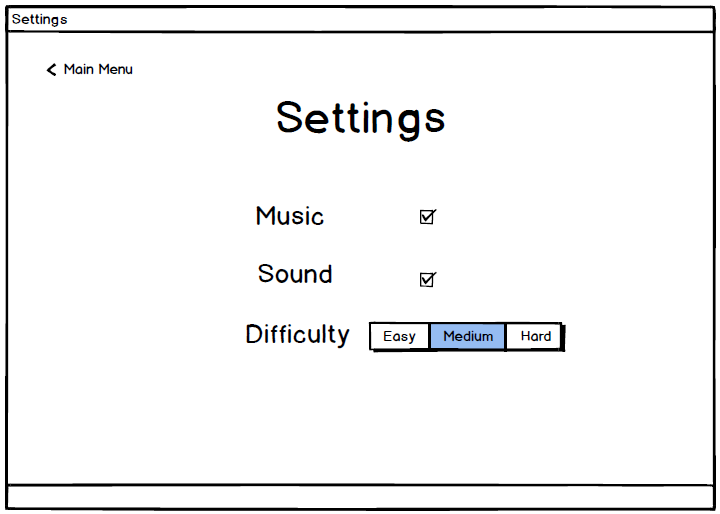
**Select Character**



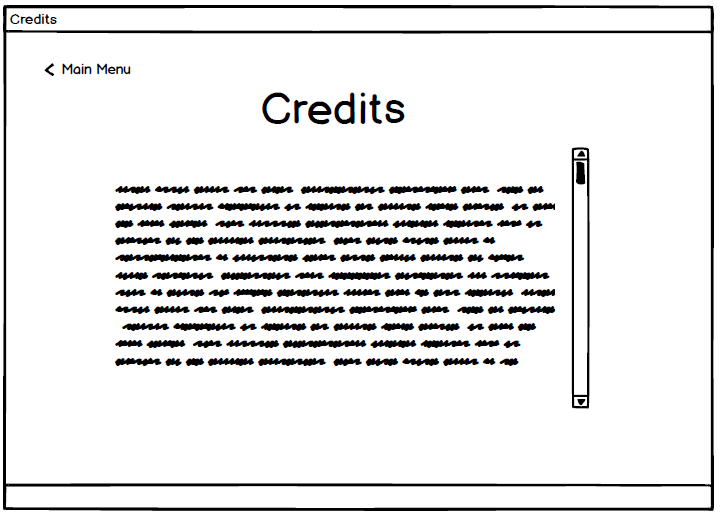
**Select Companion**



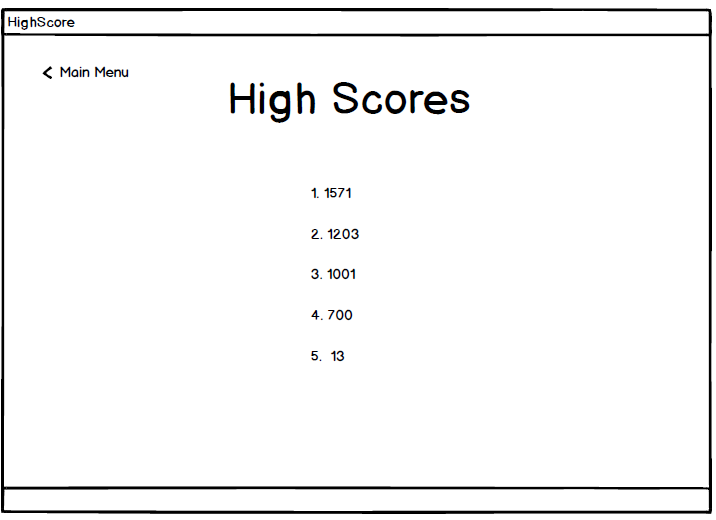
**Settings**



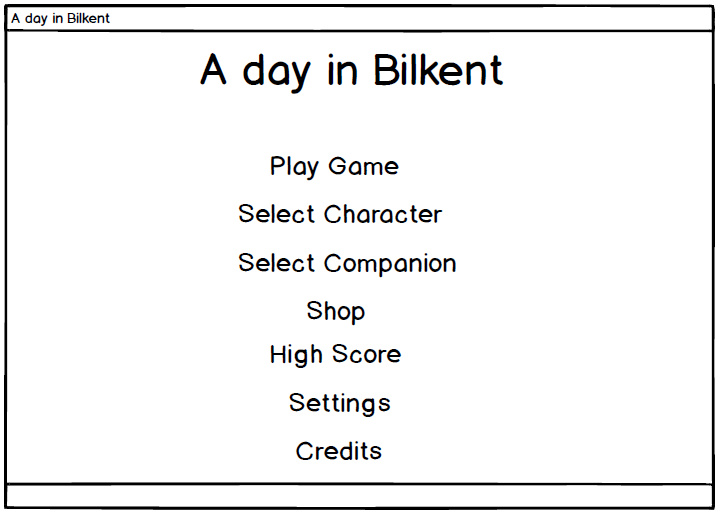
**Credits**



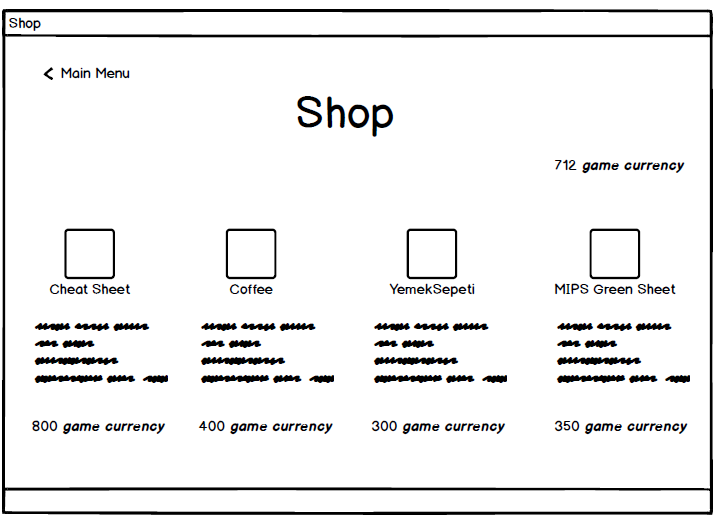
**High Scores**



**Main Menu**



**Shop**



**6. Glossary**

1- Asteroids: <http://www.freeasteroids.org/>

2- EverWing: <http://www.playeverwing.com/>

3- Space Invaders: <http://www.pacxon4u.com/space-invaders/>