CS 319 - Object-Oriented Software Engineering

Analysis Report

Left For F

Group 2A

Şekip Kaan Ekin

Ali Can Zeybek

Ömer Faruk Geredeli

1. Introduction

We decided to design a game called Left 4 F. It is a RPG game with a turn based combat system where player tries to defeat all possible enemies and clear the game using his/her skills.

This report contains an overview of the game, describes the basic gameplay elements and rules of the game. Then it describes functional requirements, non-functional requirements, use-case models including scenarios and use-case diagrams.

2. Overview

Left 4 F is a turn based RPG game where you try to graduate from Bilkent CS department. Throughout your journey you encounter many assignments, quizzes, midterms, projects and professors. Your goal is to defeat them with your knowledge and graduate without getting F's.

You, as a student, obtain some skills which will represented by weapons. Also you will learn concepts which you can use as a enhancement to your weapons. There will be known enemies such as projects and random enemies like assignments or quizzes. There will also be enemy bosses as teachers. If player as a student could not beat the enemies which represent their school requirements as in real life he/she will get an F and fail.

The game will have a turn based combat system just like in Final Fantasy. The random enemies will pop while walking in the map like walking in tall grass in Pokemon. There won’t be any safe zones in game since in Bilkent there are no safe times for a student for not to worry about lectures. The player will have health points which can be refilled by consumables. There will be learned skills and skills player can use temporary by enhancements. . The enemies will have resistances and weaknesses if player hits an enemy with certain resistance by the same type of ability the enemy will receive less damage than it would receive normally or if player hits enemy with certain weakness by the same type of ability the enemy will receive more damage that it would receive normally. During the game, the player can pause, if he/she is not currently in combat, use items and continue afterwards. Additional to this feature, the game can also be saved and loaded in order to continue from where it is saved after even closed.

2.1 List of Consumables

- **Coffee:** Coffee will be the most common consumable to refill players health but the amount of health to be refilled will be minimum for coffee.

Image 1:Coffee Appearance [1]

- **Energy Drinks:** Energy drinks will not be as common as coffee, but the amount it will refill the players health will be higher than coffee’s.

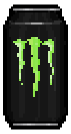


Image 2:Energy Drink Appearance [2]

- **Food:** Food as for all humans will also be an essential for the player to keep up with good health amounts. While the difficulty to finding foods differ different foods will be have different effects on player.
  - **Pizza:** Compared to all the unhealthy food a student might eat, pizza could be considered harmless. Pizzas will be refill the players health without any side effects.
  - **Hamburger:** Hamburgers will also refill the players health without any side effects.
  - **Doughnut:** Doughnuts will completely refill the players health but since they are harmful for health they will reduce the player health by a slight amount.
  - **French Fries:** French fries will be easy to find but by eating them the player won’t get much of a refilment while he will lose more from his maximum health compared to doughnuts.
  - **Hot Dog:** Hot dogs will also refill the players health without any side effects.
  - **Egg:** Eggs are the most healthy food a student possibly consume, so while eggs refill the players health slightly they will increase the players maximum health slightly.

Image 3:Appearance of All Foods [3]

2.2 Enhancements

The list of enhancements are not clear yet. There will be two kind of consumables there one kind provide new skills to player the other kind will improve the general stats of the player.

2.3 List of Enemies

- **Quiz-Assignment:** These two types of enemies will be occur randomly while player tries to reach the other predefined(must defeat) enemies. The difficulty of quizzes and assignments will be randomized yet they will be easier to defeat relatively. Player can also directly run away from these enemies with the current levels escape penalty.

Image 4: Quiz and Assignment Appearance [4]

- **Teacher:** Teachers will be the final bosses at the end of each level, to proceed to next level player must defeat the professor of current level.

Image 5:Teacher Appearance [5]

- **Midterm:** Midterms will be easier according to professors and harder than projects and random enemies. The player must fight with the midterms yet he/she can chose to run away after lowering the midterms health to the levels predefined percentage. If the player chooses to run away after he/she reach the minimum requirements he/she will get an escape penalty which is randomized. This randomization represents the curve grading system in Bilkent.

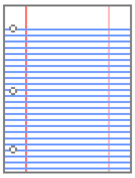


Image 6:Midterm Appearance [6]

- **Project:** Projects will be harder to defeat than quizzes and assignments and player must defeat projects.

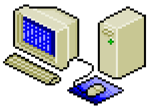


Image 7: Project Appearance [7]

2.4 Levels

Each level will be a year and the game will be containing four of them. The game background will be the same but at each year, the enemies will be more hard to pass than the last year. Player can't play the next levels without passing the previous ones.
The locations of the enemies and the collectibles will change at each level. If the player fails to pass the enemies, he/she will get an "F" and must repeat the level.

2.5 Characters

Players can choose which character to play in the first year and has to play with it until the game is done, or they can choose to start over with a new character. There will be two characters, one is a male and the other will be a female character.

3. Requirement Specifications

3.1 Functional Requirements

3.1.1 Play

The main idea of the game is to pass 4 years and graduating. The player will start the game with 1 energy drinks, 1 coffee, 100 points of energy and 1 attack. The energy may decrease by the attacks of the enemies or skipping enemies. When players encounter an enemy, the battle screen will come up and from there, they can choose to attack, enhance their attacks or to increase their energy. The energy of the enemies will vary accordingly to their difficulty level.

Players can collect collectibles that scattered randomly in the map. They can use them in the battles or in the pause menu. The effects of each collectible will be shown in the tutorial screen.

3.1.2 Tutorial Screen

Tutorial will show the players how to move and use their items. It will also show that how the turn based fights work and the effects of the collectibles.

3.1.3 Options Menu

In the options menu, the game sound which will be playing on the background can be changed or muted.

3.1.4 Pause Menu

While playing, players can access to the pause menu by pressing P. In the pause menu, player can choose to resume, options menu or the main menu. Also, a button will open a new popup to show the items that the player has and they can use them here.

3.1.5 Credits Screen

Credits menu will contain the names and contact information of the developers.

3.1.6 Exit

On the main menu, there will be an exit button which will close the game.

3.2 Nonfunctional requirements

3.2.1 User-friendly Interface

A game needs to attract the user. Therefore, our game will be visually comfortable and appealing. Also our group will be careful with the details of the map and the enemies so that they will look good.

3.2.2 Reliability

A bug-free game is a reliable one. Thus, our main concern will be around this issue. Our aim is to consider all cases and solve the problem in the development stage, so that we will have a reliable program.

3.2.3 Entertainment

There will be a funny and appealing story in the game which users can relate to. Varying enemies and collectibles will make the user smile and want to play it again. Difficulty system in each level and the randomized positions of the enemies will avoid the repetition in the game.

4. System Models

4.1 Use Case Model of Left 4 F

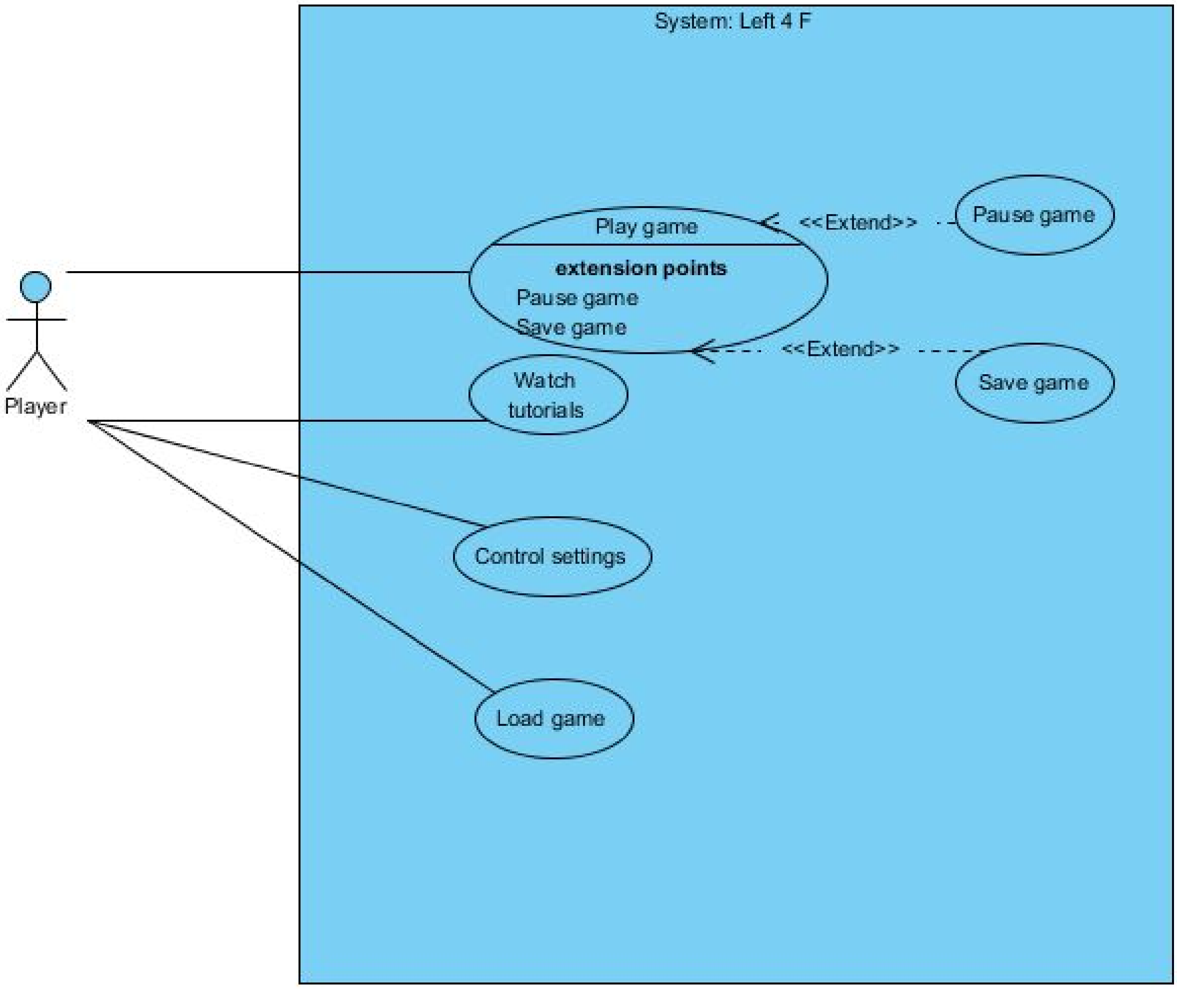


Figure 1: Use Case Model of Left 4 F

4.1.1 Play game
Use case name: Play game
Primary Actor: Player

Entry condition: Player has already opened the game and on the main menu

Exit condition: 1) Player finishes the game successfully

2) Player exit the game
3) Player loses the game

Main flow of events: 1) Player select his character
2) Player starts to play Left 4F
3) Player passes levels and does assignments
4) Player finishes the game
5) Player returns to the main menu

Alternative flow of events: 1) Player loses his lives
2) Player pauses Left 4F
3) Player saves Left 4F
4) Player can exit Left 4F

4.1.2 Watch Tutorials

Use case name: watch tutorial
Primary actor: Player

Entry condition: Player clicks watch tutorial in the main menu

Exit condition: Player returns to the main menu

Main flow of events : 1) System provides videos to the player
2) Player watches videos
3) Videos end
4) Player returns to the main menu

Alternative flow of events: 1) Player skips tutorials

4.1.3 Control Settings

Use case name: Control settings
Primary actor: Player

Entry condition: 1) Player has already opened the game on the main menu
2) Player pauses Left 4F while playing

Exit condition: 1) Player returns to the main menu
2) Player continue to play Left 4F

Main flow of events: 1) Player changes musics or mute off
2) Player views help
3) Player returns to the game

Alternative flow of events: -

4.1.4 Load Game

Use case name: Load game

Primary actor: Player

Entry condition: Player has already opened the game

Exit condition: Player returns to the main menu

Main flow of events: 1) Player selects the chosen game
2) System loads the game
3) Player starts to play Left 4F

Alternative flow of events: -

4.2 Dynamic models

4.2.1 Sequence Diagrams

4.2.1.1 Changing music from Options Menu

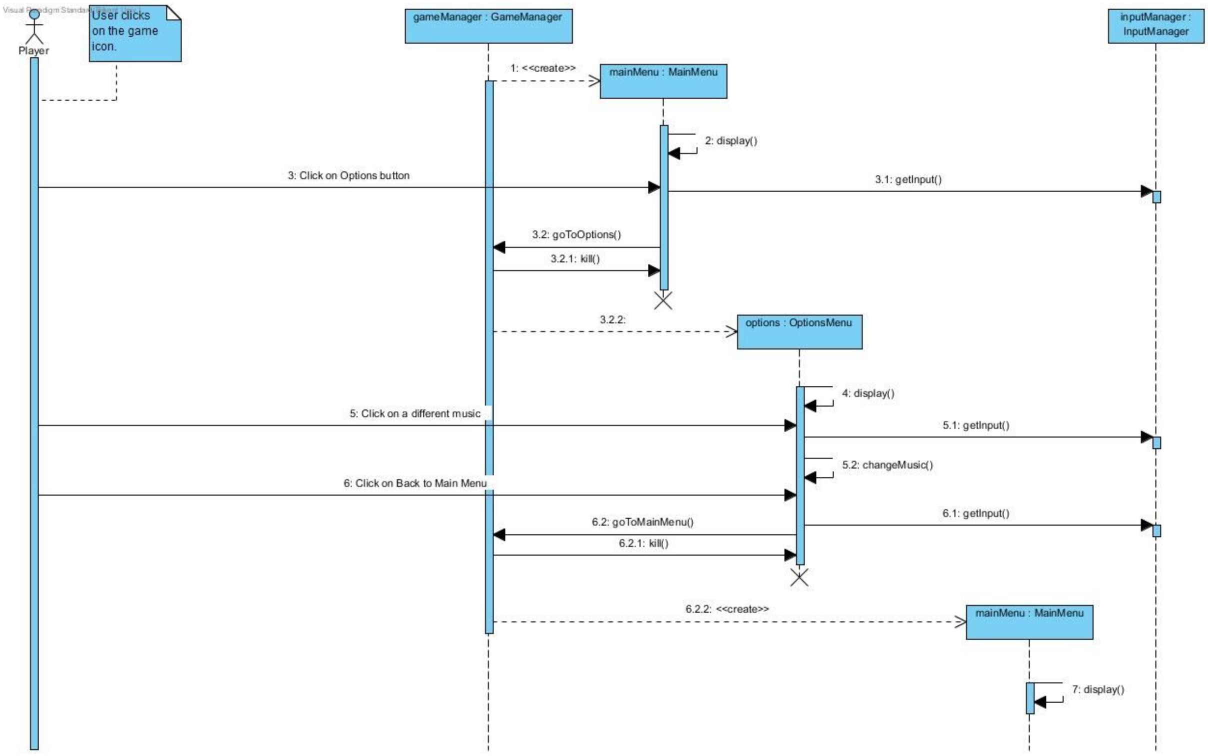


Figure 2: Sequence Diagram of changing options

Scenario: In this scenario the user opens the game by clicking it on the desktop. The game opens up with Game Manager which is the boundary object and it creates a main menu object. Then user communicates with the main menu and select options. Then, game manager kills the main menu and creates the options menu object which displays itself. User changes the music that is played. User can also choose to mute or unmute the game here. Options menu changes the music which is hold in the game manager. Then the user hits Back and game manager opens a new main menu object after killing options menu object.

4.2.1.2 Credits / Tutorial:

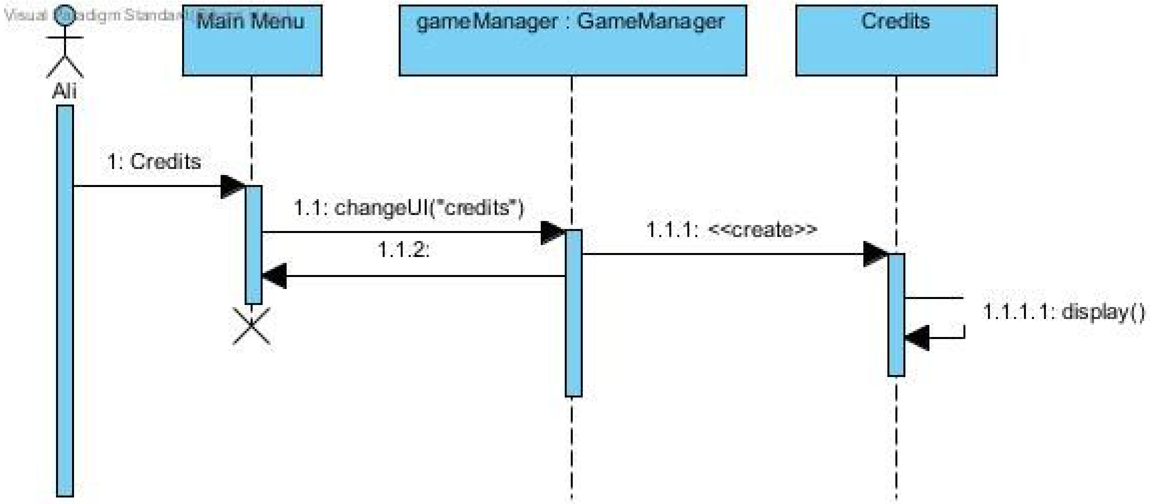


Figure 3: Sequence Diagram of Credits and Tutorial

Scenario: The Credits and Tutorial can be accessed the same way from the Main menu. The user communicates with the main menu boundary object and selects Credits/Manual. The chosen page is displayed. In the end, the user presses back and is redirected to Main again. From there, user chooses to exit.

4.2.1.3 Play:

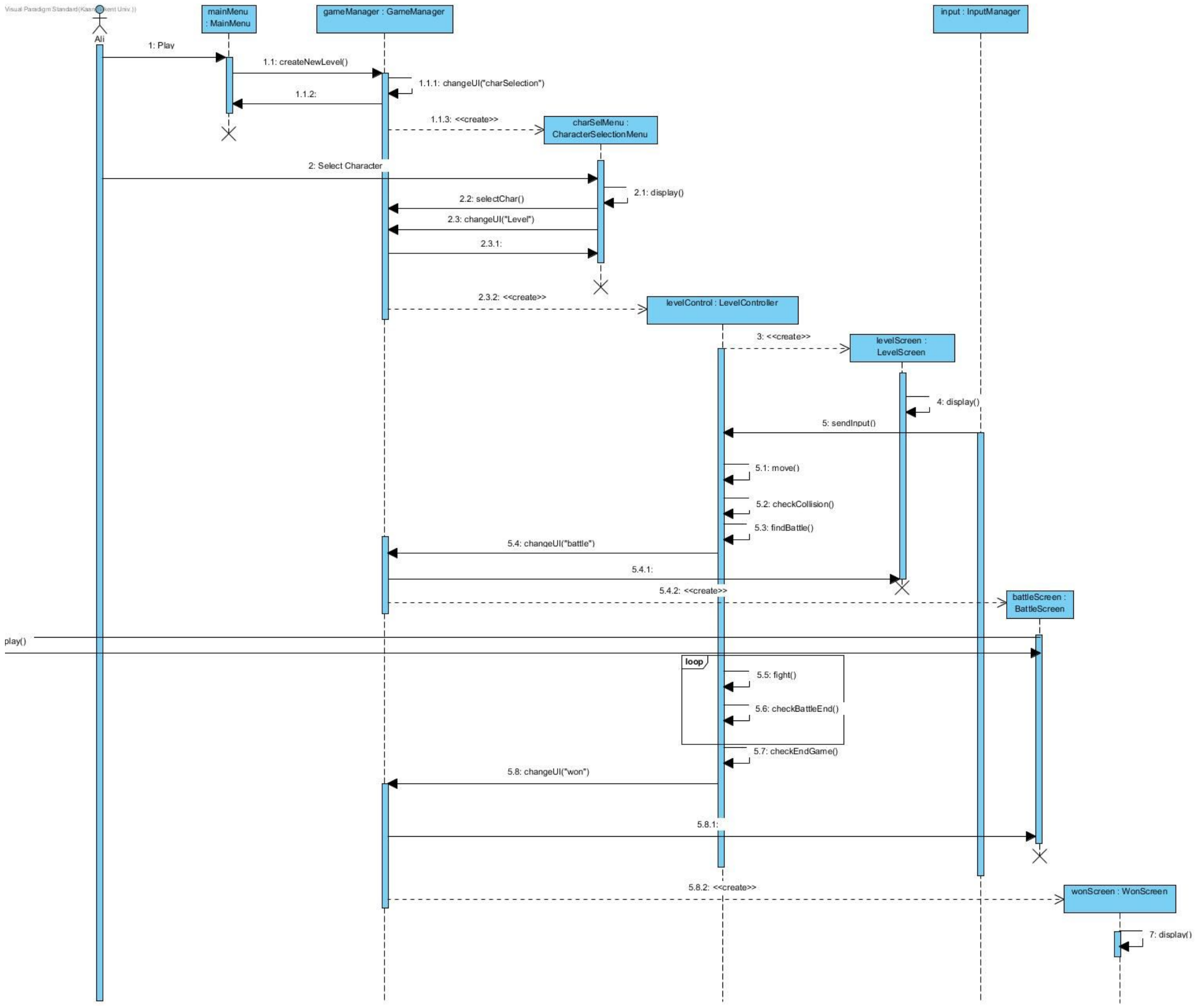


Figure 4: Sequence Diagram of Play

Scenario: The player interacts with the boundary object which is the main menu and clicks the Start button. The character selection menu opens up and asks the user to choose a character. User chooses a character and game manager creates a new level. Player, enemies and a new map with randomly placed collectibles arranged.
After the game initialization, the continuity will be kept in a loop with several conditions, which are what keys the user pressing. In this scenario, level controller takes the user input which is "D" and moves the character accordingly. If a collision with any collectible is encountered, manager will increase the number from the player properties. If not, play will go normally.
After moving, if an enemy is at the player's position, battle screen will open up immediately. There, turn will be considered and if it's enemy's turn, it will attack accordingly to its programming. If it's player's turn, it will wait for user to give an input and do it. At the end of each turn, a controller will check if the player's or enemy's health is reached 0. Then the battle screen will close and game will resume from where it stopped.

If player wins the battle the program checks the battle end, then win screen willbe displayed.

4.2.1.4 Load Game

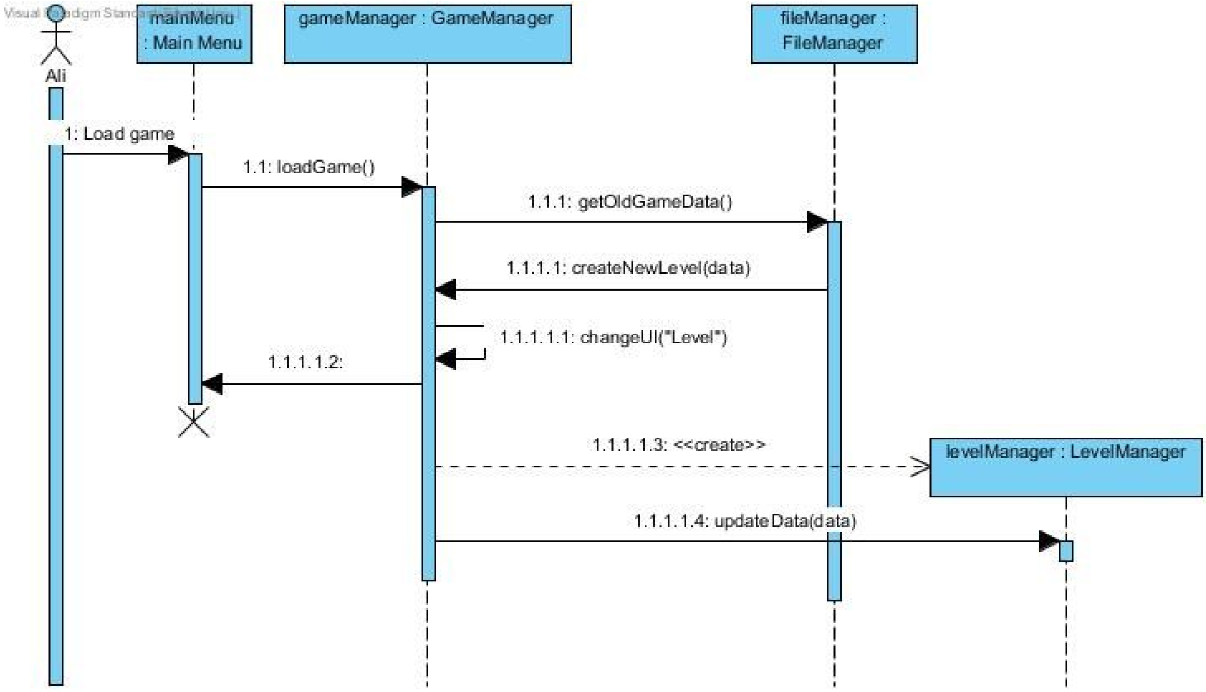

Visual Paradigm Standard
Ali
mainMenu : Main Menu
gameManager : GameManager
fileManager : FileManager
1: Load game
1.1: loadGame()
1.1.1: getOldGameData()
1.1.1.1: createNewLevel(data)
1.1.1.1.1: changeUI("Level")
1.1.1.1.2:
1.1.1.1.3: <<create>>
levelManager : LevelManager
1.1.1.1.4: updateData(data)

Figure 5: Sequence Diagram of Load Game

Scenario: Ali chooses "Load a Game" button from main menu . GameManager kills the MainMenu object and asks FileManager to get the last game's attributes. Then GameManager creates a new LevelController object with the attributes of the last game. After that, GameManager creates a new Player object. Lastly,it will change the displayed UI to a new LevelScreen.

4.2.1.5 Finding and Using an Item

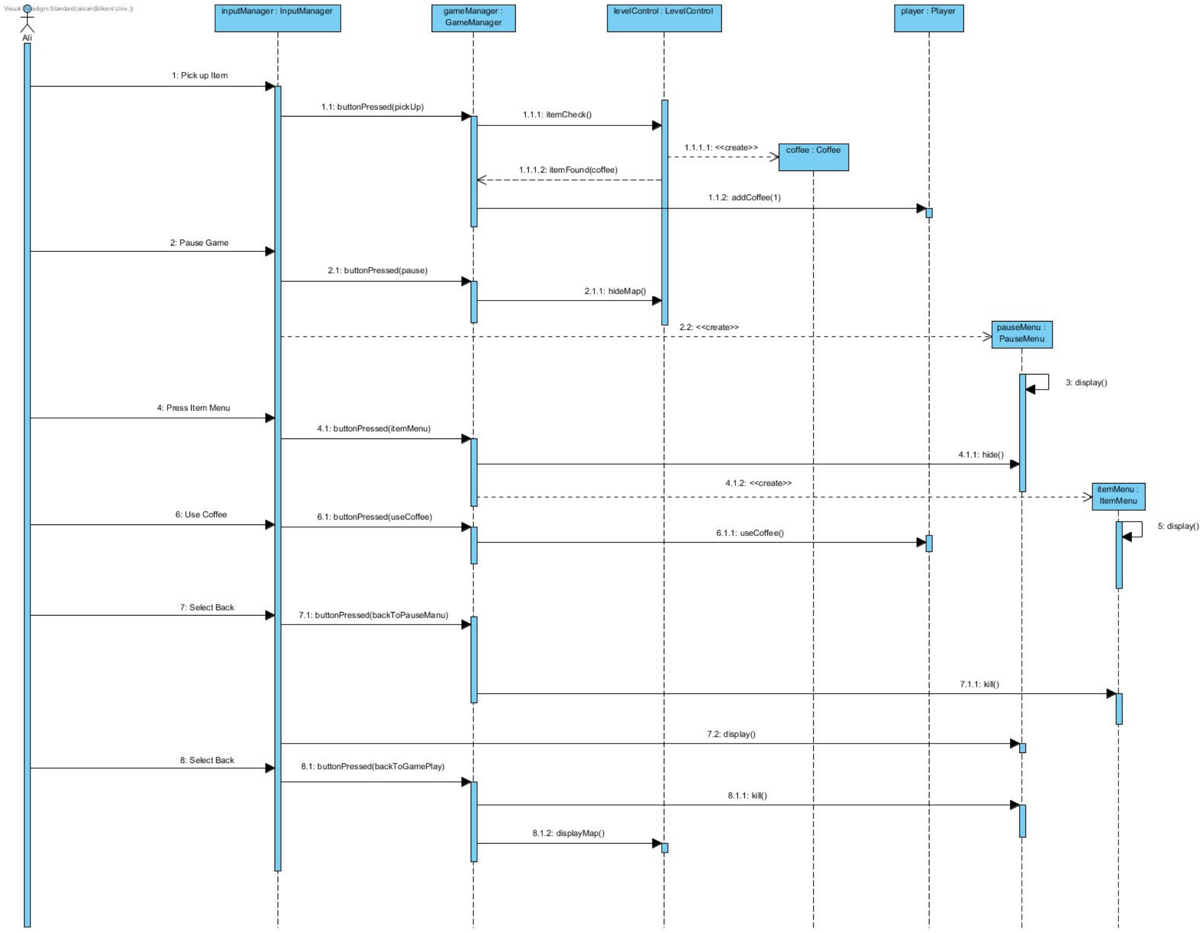


Figure 6: Sequence Diagram of Finding and Using an Item

Scenario: When player comes across and picks an item, game will recognize the item and create an instance of that object. In this case Ali founds a “Coffee” object and “LevelControl” creates an instance of “Coffee” class then this object is added to the player's inventory. To use an item player pauses the game and “GameManager” signals “LevelControl” to hide game map then “GameManager” creates an instance of “PauseMenu” where player can select “Items”, “Options”, “Return to Main Menu” and “Resume”. However in this scenario “Items” is selected. “GameManager” hides the “PauseMenu” and creates a new “ItemMenu”. When “Coffee” is selected by Ali, “GameManager” update “Player” object. To return back to game the “GameManager” closes “ItemMenu” and “PauseMenu” respectively and signals “LevelControl” object to show game map.

4.2.2 Activity Diagram

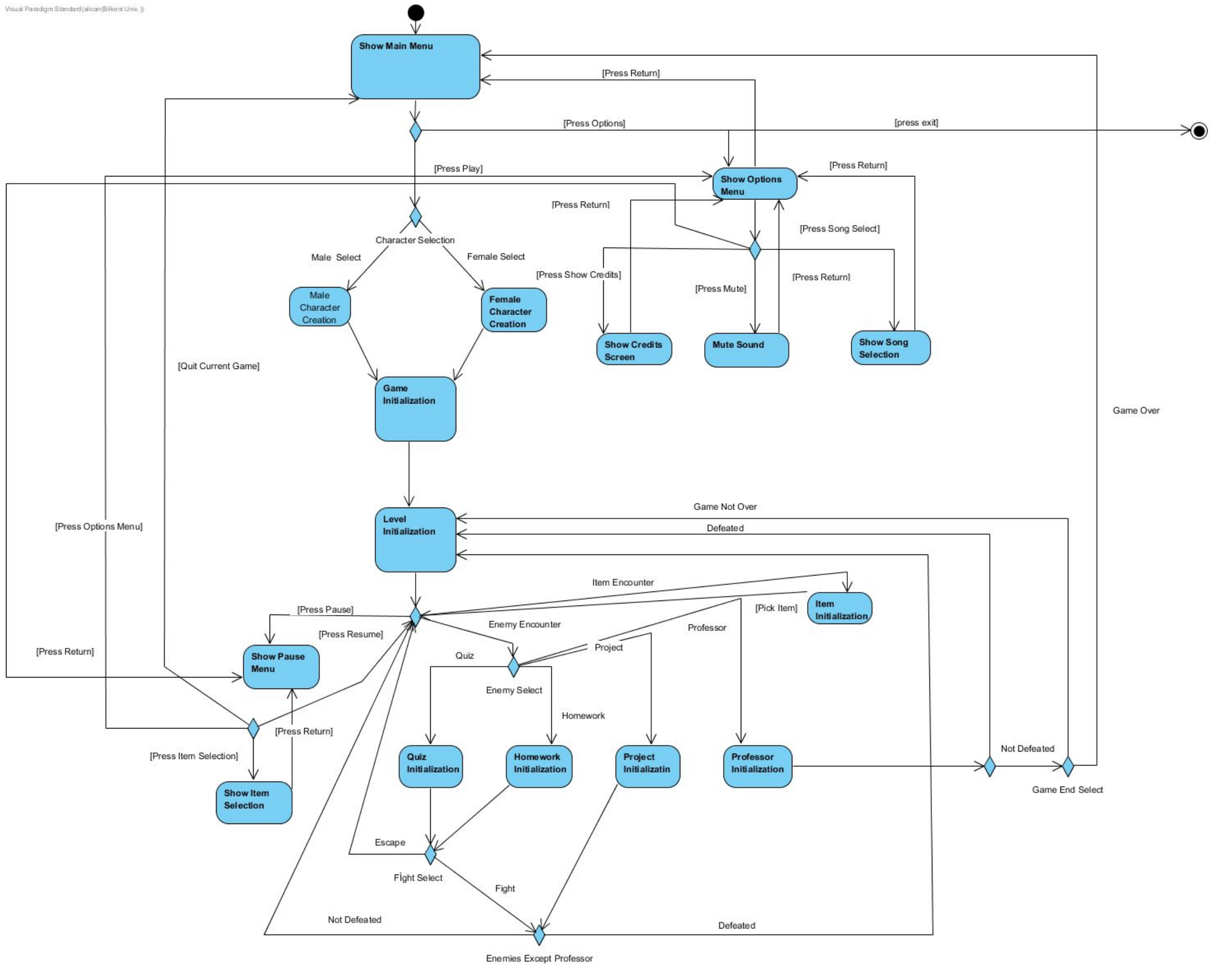


Figure 7: Activity Diagram of the Game

The user will first encounter the main menu where he/she can start a new game by pressing Play, close the game by pressing Exit, or go to options menu by pressing Options. After pressing play the system will show the character selection screen to user where he/she could select a male or a female character to play with. If the user presses Options the system will show the options menu where the user could select a song to play, mute the sound or show the Credits Screen.

As the user is inside the game, he/she will encounter different items and enemies. If the user encounters with an item he/she has to collect it. If the user encounters with an enemy, accordingly to the type of the enemy he/she can fight or escape. The general rules such as the escape penalty will be shown in the beginning of the each level as tutorials.

On the fight menu user has two options to run away or to fight. If the user run away from an enemy his/her life points will be decreases accordingly to the type of the enemy and the general rules of the level player is currently in. If the user chooses to fight there will be maximum of four skills he/she can use which will vary accordingly to the players previous experiences throughout the game.

The skills a player can use will be determined by the items and enhancements he/she collected throughout the game. Players should choose their arms wisely to be successful at this game since the skills they obtain by those items have specific types which will be effective or not effective to a certain enemy. The items could also be life point refillment like coffee.

The user can pause the game to select items, to return to main menu or to go to options menu.

There will be different types of enemies as quizzes, homeworks, projects and professors. When a players hit points hit zero he/she will get a F and has to repeat the level. After defeating final professor the game will be and the user will be led to main menu.

4.3 Object Model

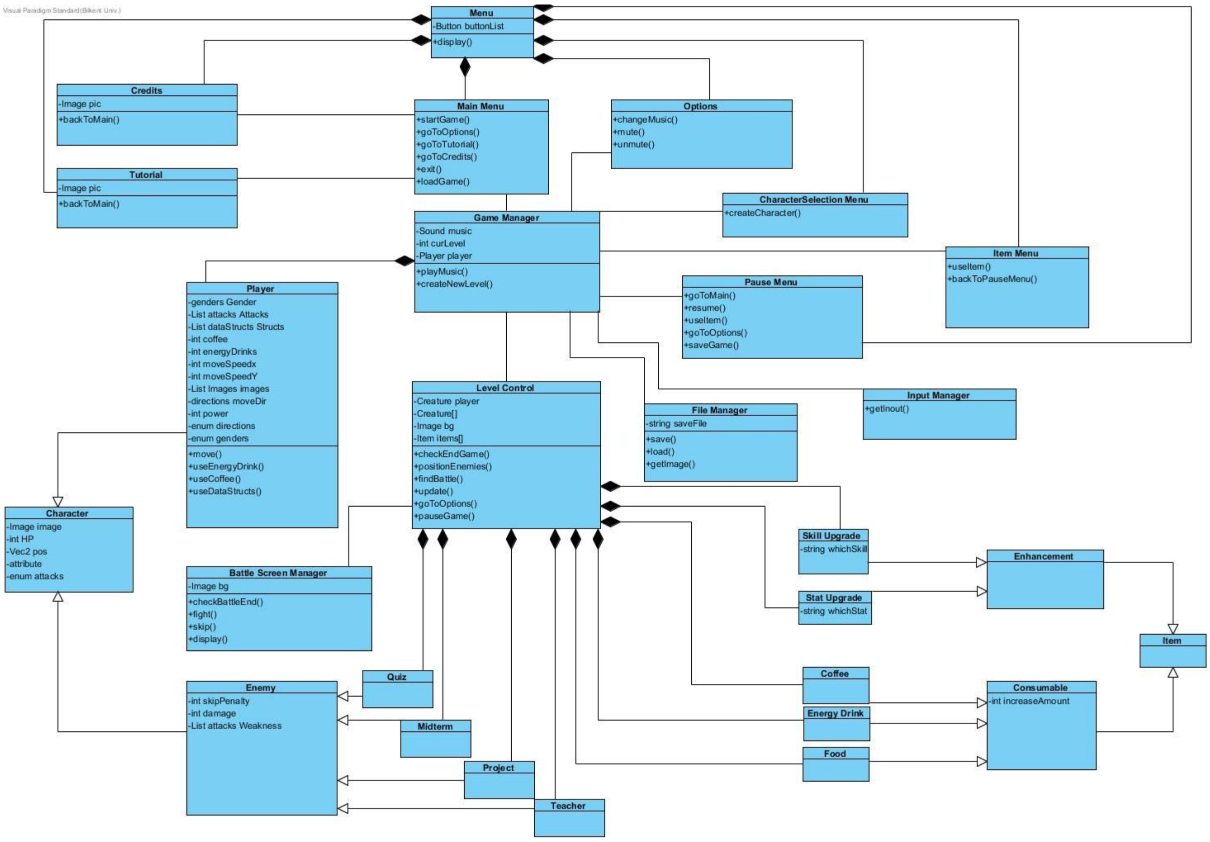


Figure 8: The class diagram of the Left for F is illustrated above. There are 28 classes.

First of all, Menu class, is a parent class to *Main Menu, Credits, Tutorial, Options, Character Selection Menu, Pause Menu* and *Item Menu classes*. These Menu classes each have their own functions but Menu class holds the buttons and display function which every menu class has.

There are 4 manager classes. *Game Manager* class is the main organizer of the game. It is supposed to control the background, creating and killing menus, creating new levels and Level Controllers of these levels, controlling the inputs which is gotten from *Input Manager* and saving or loading the game via *File Manager*. *Level Controller* is responsible for moving the player, picking up collectibles, creating *Battle Screens* where player fight with the enemies and checking the game end. File Manager also supposed to get the images that is requested from other classes.

Left for F's entity objects are derived from 2 base classes which are *Character* class and *Item* class. Character class is the parent class of Enemy and Player classes. There are 4 enemy classes which are Quiz, Midterm, Project and Teacher which inherits Enemy class. Item class is parent class for *Consumables* and *Enhancement* classes. *Coffee, Energy Drink* and *Food* classes inherits Consumables class, *Skill Upgrade* and *Stat Upgrade* classes inherit Enhancement class.

Our goal for this game was a clean and understandable structure which we acquired with inheritance of classes.

5. User interface

5.1 Navigational Path

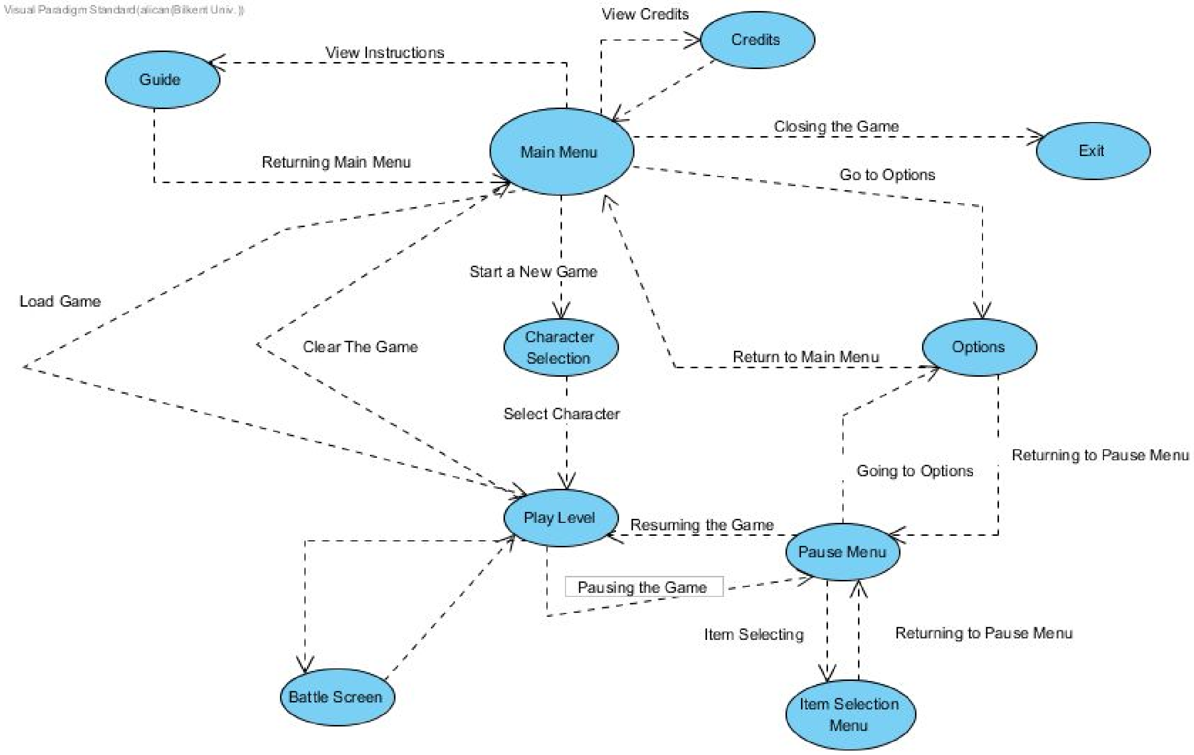


Figure 9: Navigational Path of the Game

Main menu is the first thing that is shown when the game opens. In main menu, player can choose to star a new game, load the previously saved game, change settings, view the guide document, credits or terminate the program.

Start A New Game: When this button is pressed, program starts a new game with the specified settings and map.

Load Game: When player hits the load game button, the game starts with the previously saved game data.

Options Menu: This interface allows user to control sound settings.

View Instructions: When this button pressed, help document will be shown to give the player information about game.

View Credits: When this button pressed, information about developers will be shown.

Quit Game: This option terminates the program and writes back all required information that will be saved from memory to hard disk.

5.2 Screen Mockups

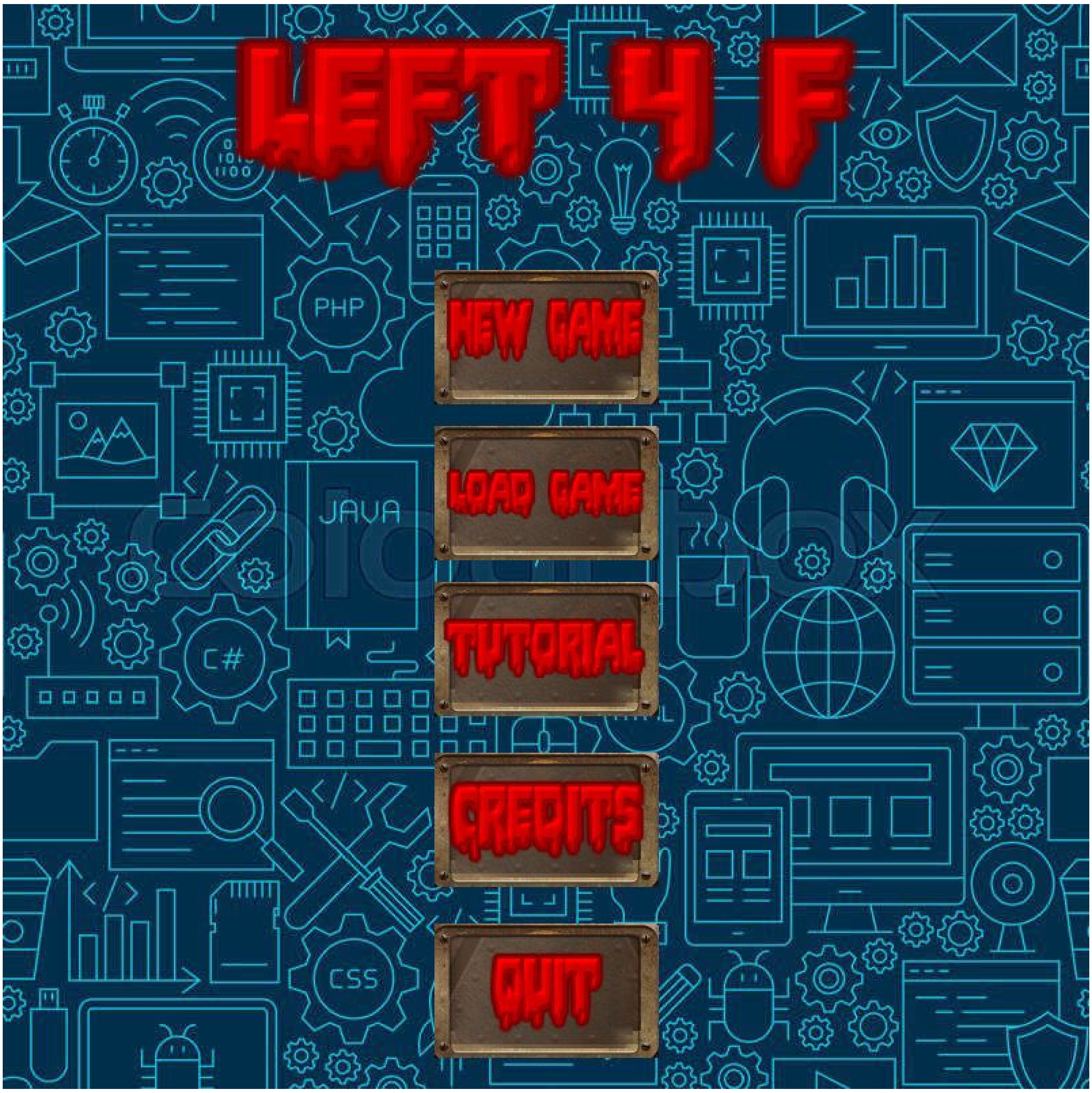


Figure 10: Main Menu [8][9][10]

This menu will be displayed when the user opens the game.

As explained above sections, clicking the New Game will start a new game and display the character selection screen.

Load Game will load an already saved game.

Options take the user to options menu.

Quit closes the game window.

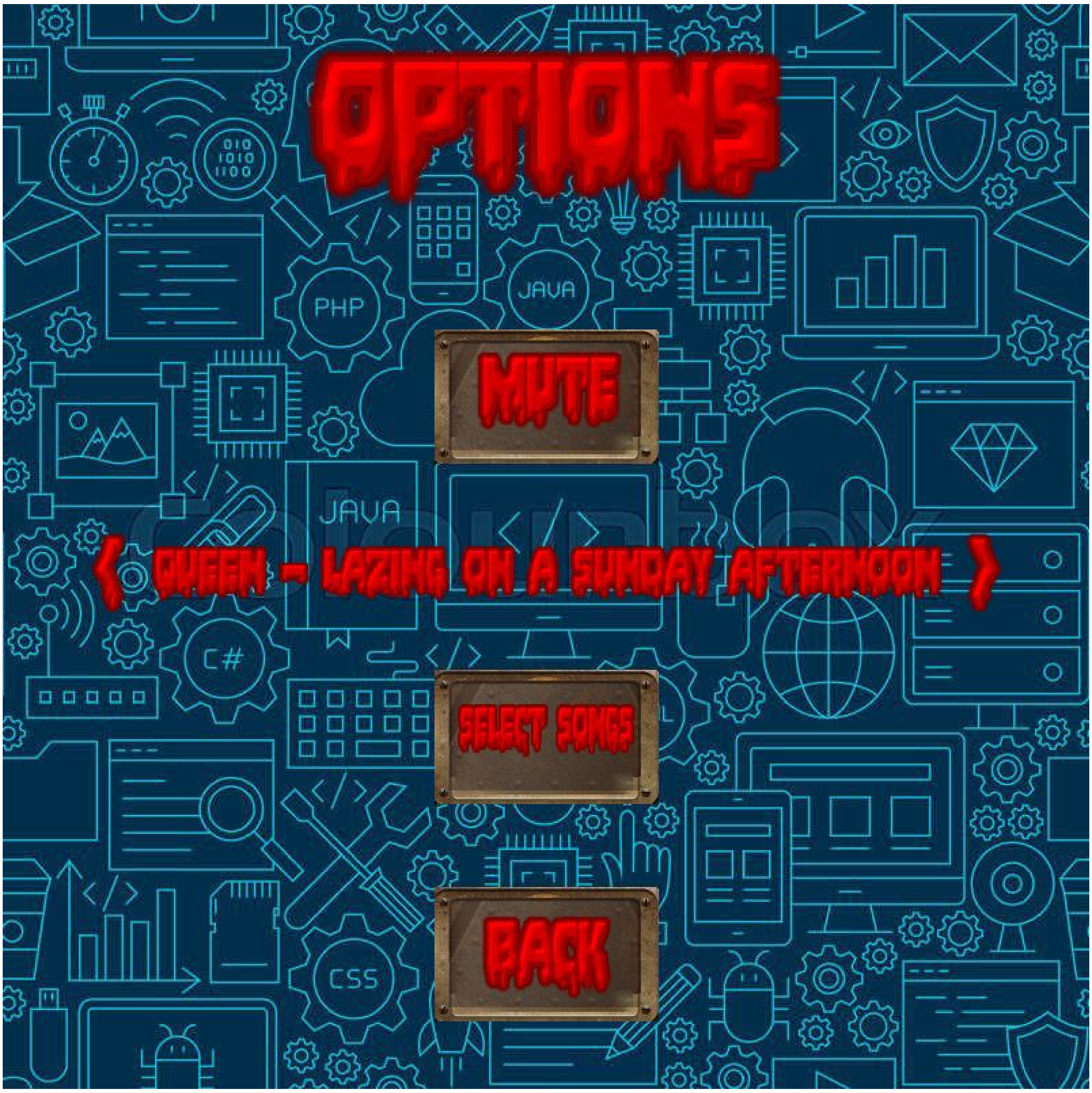


Figure 11: Options Menu [8][9][10]

This menu will be displayed when user clicks options in the main menu or in the pause menu.

The user has three choices as Back, Mute Sound, Select Songs.

Mute Sound will be mute sounds of the game.

Song Select will be allow user to choice a song to play while he/she plays the game.

Back will return user to screen he/she came from.

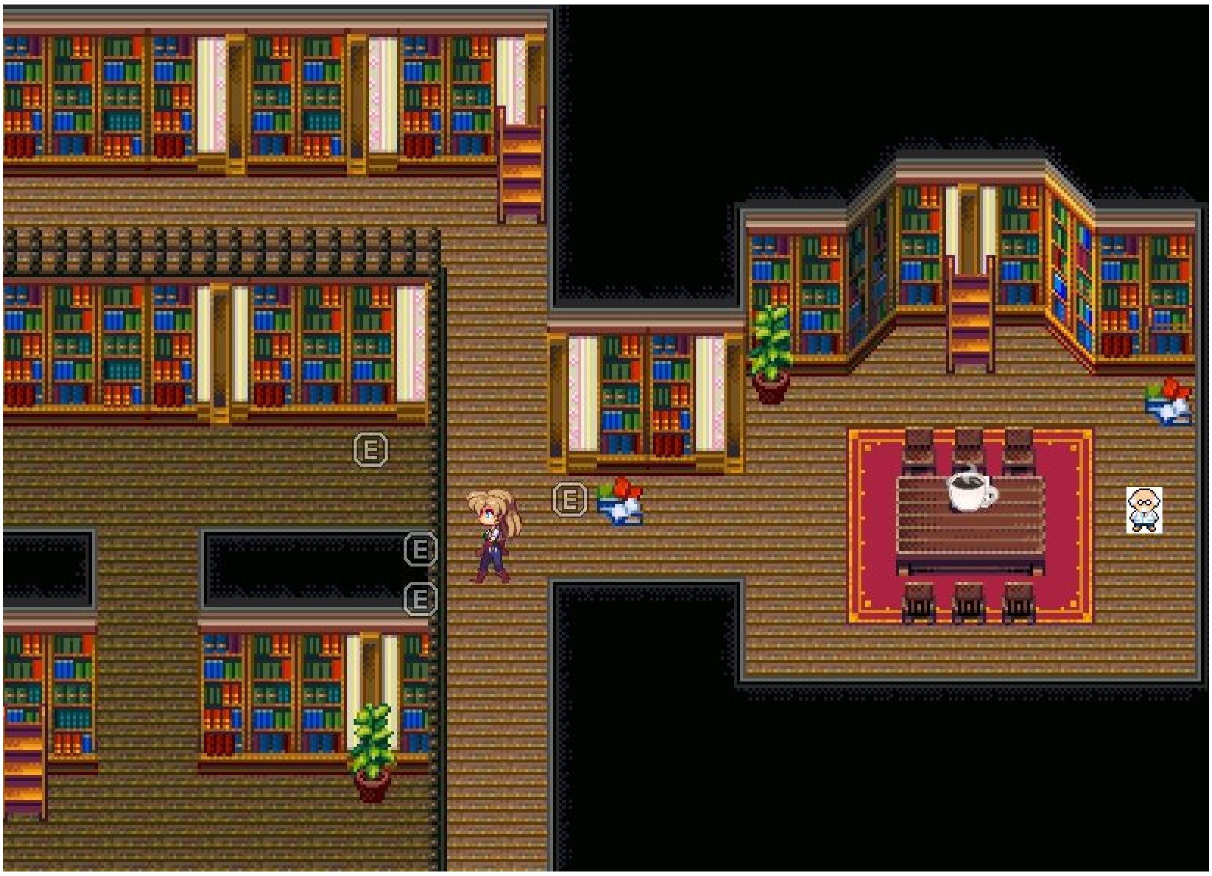


Figure 12: Main Menu [1][5][11][12]

In the game screen player can see available collectable items and known enemies. As seen in this figure there is a teacher enemy in the right of the screen and a coffee object left to teacher which player can go and collect.

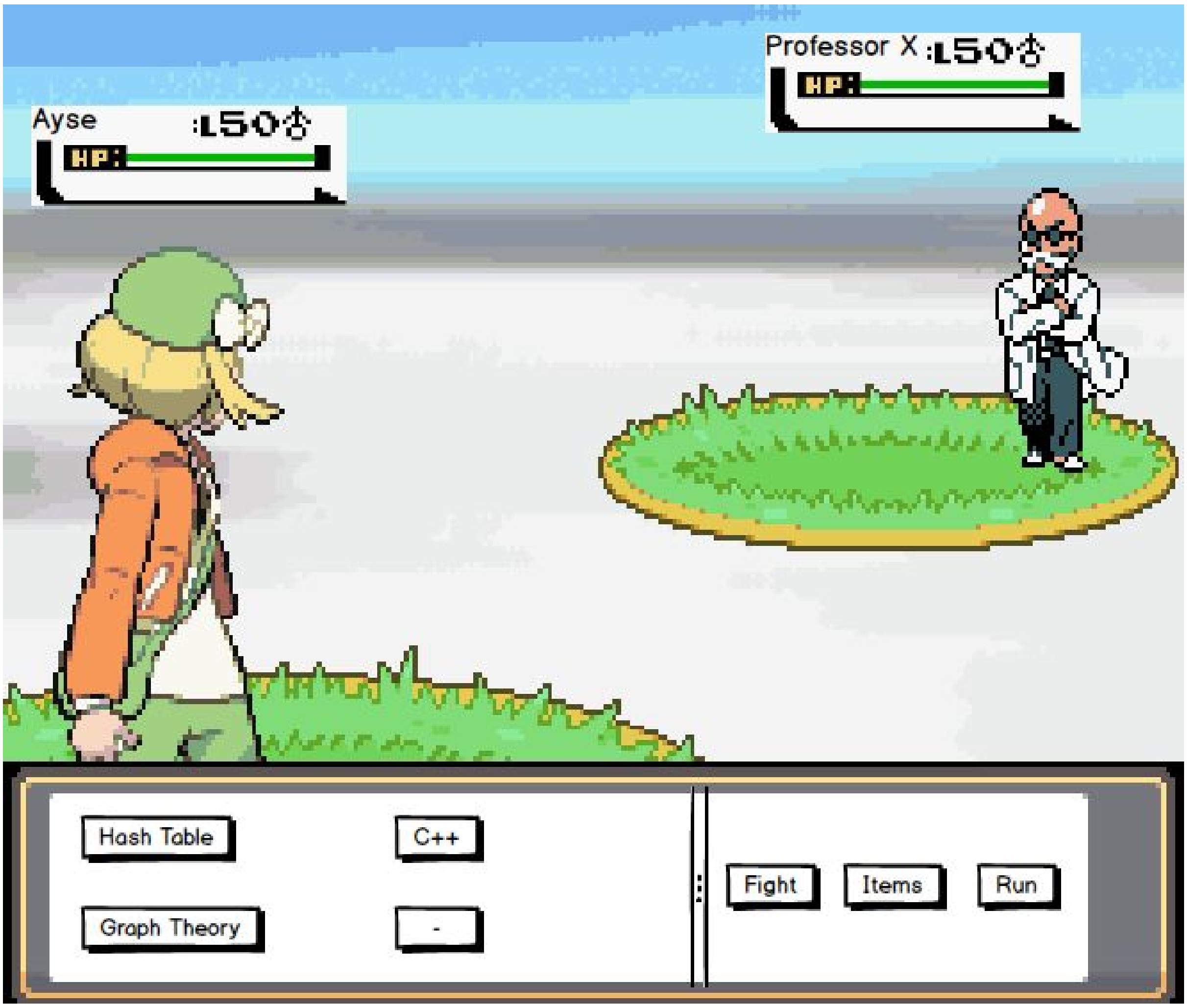
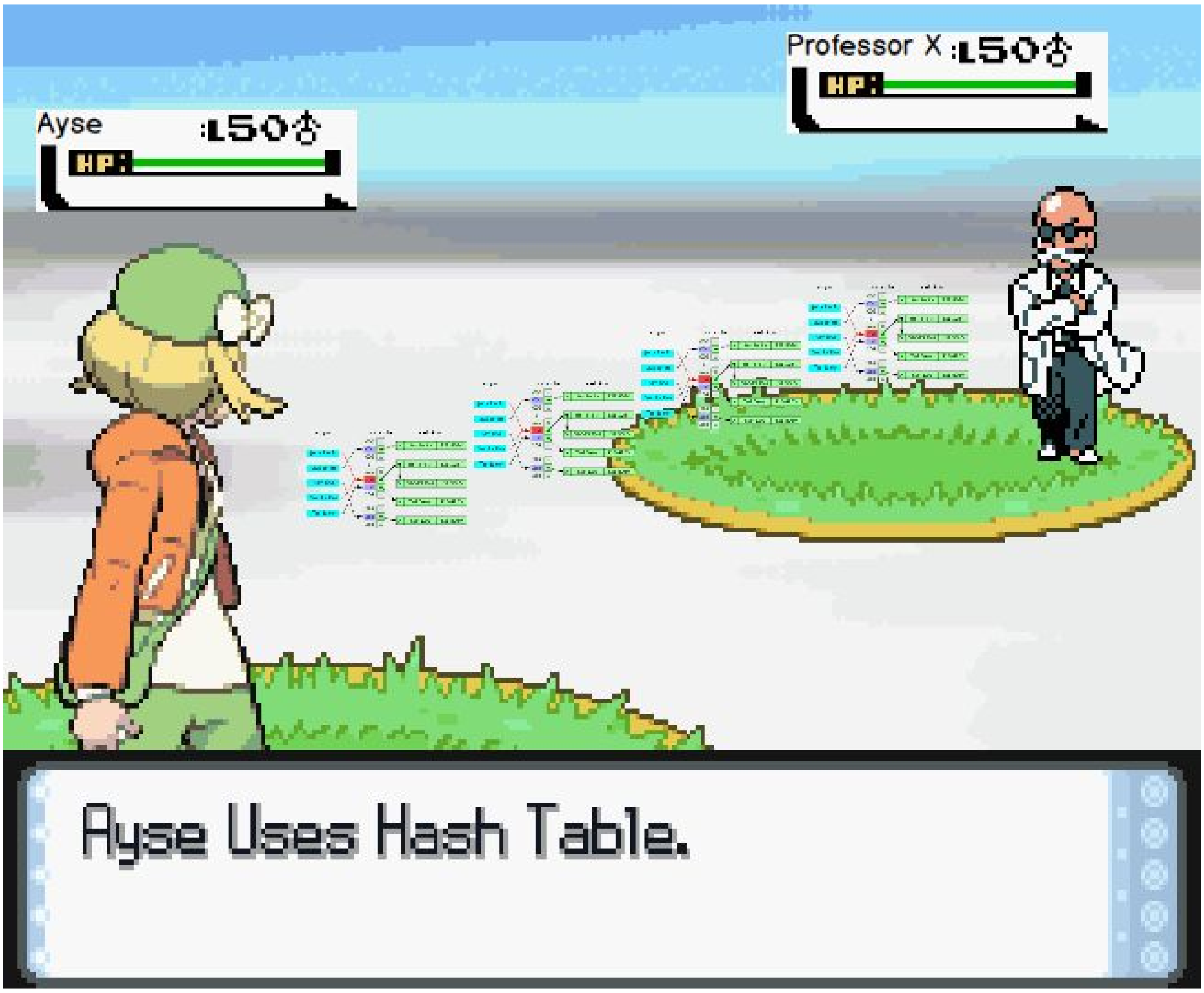


Figure 13: Main Menu [13]

This is the battle screen of the game where player can select to fight use items or run away. Left to main choices of the player there are different kinds of attack moves player can use.



Ayse Uses Hash Table.

Figure 14: Main Menu [13][14]

This is an example screen of after player selects “Hash table” attack move on the previous screen. After the attack animation ends, the health bar of the “Professor X” will be decreased accordingly.

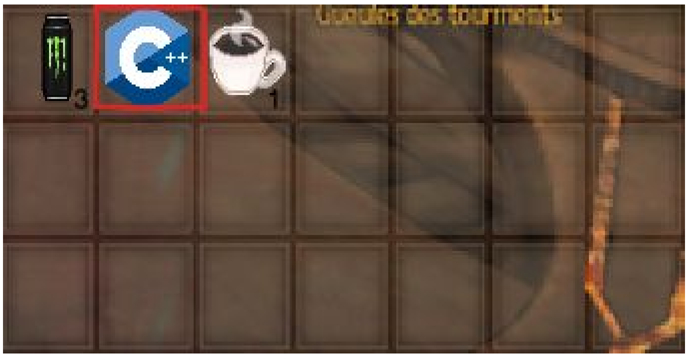


Figure 15: Main Menu [1][2][15][16]

This screen is representation of users inventory. The numbers “1” and “3” represents number of objects from a kind in players inventory. In this example player has three “Energy

Drink”s and one “Coffee”. The red square on C++ image indicates, that enhancement is equipped and player’s stats calculated accordingly.

6.References

[1]https://amsa95.deviantart.com/art/WC-Coffee-595368783

[2]https://aprocitied.deviantart.com/art/I-drink-Monsters-139802665

[3]https://novpixel.deviantart.com/art/FOOD-PIXEL-ART-ICONS-605906984

[4]https://sneezeupyournostrils.deviantart.com/art/Stack-Of-Paper-399063171

[5]https://coheedark-industries.deviantart.com/art/Tiny-Proffessor-38918485

[6]https://empresskyrav.deviantart.com/art/notebook-paper-1893575

[7]https://img00.deviantart.net/ca91/i/2004/196/1/f/computer.gif

[8]https://d2gg9evh47fn9z.cloudfront.net/800px_COLOURBOX23310792.jpg

[9]http://www6.flamingtext.com/

[10]http://talkingmoose.ca/wp-content/uploads/2011/03/steampunkApr28c.png

[11]https://el-sato.deviantart.com/art/Commission-female-player-walk-619759326

[12]https://alcancevitoria.files.wordpress.com/2009/12/interface1-758507.png

[13]http://www.pokemonbattlecreator.com/

[14]https://upload.wikimedia.org/wikipedia/commons/thumb/d/d0/Hash_table_5_0_1_1_1_1_1_LL.svg/450px-Hash_table_5_0_1_1_1_1_1_LL.svg.png

[15]https://upload.wikimedia.org/wikipedia/commons/thumb/1/18/ISO_C%2B%2B_Logo.svg/1200px-ISO_C%2B%2B_Logo.svg.png

[16]https://i.redditmedia.com/ygbVdbAY68eCpVt18sUq_5fFX9TMk1t5jC8_OMgIJOk.png?w=1024&s=4b246394e22ba19797be233f6432f3b7

[17]https://opengameart.org/content/city-background-repetitive-2