CS 319 - Object-Oriented Software Engineering
Analysis Report

Project Name: Survival in Bilkent

Group 2-M

Pelin Elbin Günay - 21402149

Kübra Nur Güzel - 21400946

Alper Şahıstan - 21501207

Semih Teker - 21300964

Table of Contents

1. Introduction

We decided to design a top to down 2D shooter game which is called “Survival in Bilkent”. In this game, the player makes an effort to stay alive in a restricted area. There are a variety of enemies such as bugs, assignments, quizzes, labs, midterms, and finals. These enemies will try to kill the player which is a CS student at Bilkent University by either shooting him or crashing into him or spawning other enemy units that will attack the player. The aim of the student is to defeat these enemies by shooting codes and pieces of his/her will to study. The game has 4 levels which represent the 4 years of the major.

The game will be a desktop application and will be controlled by a mouse and W-A-S-D keys on the keyboard.

This report contains an overview of the game, description of the basic gameplay. It also contains functional requirements, non-functional requirements, system models including use case model, dynamic models, object and class model, and user interface-navigational paths and screen mock-ups.

2. Overview

After launching the game player encounters game menu which has “Start Game”, “Options” and “Quit” buttons. The game commences when the player chooses “Start Game” option. “Options” will take the player to the Options menu where he will be able to modify sound settings. “Quit” option terminates the game.

When the game commences player will encounter various types of enemies that will try to kill him by shooting or crashing into it. The player’s objective is to survive the level without running out of time (the player will only have time instead of health which will decrement as the time passes and additionally decrements when taken damage) by shooting the enemy units while collecting miscellaneous power-ups, keys, chests, and coins. At the end of each

level, the player will face stronger enemy units which are called “Finals”. Top layer to proceed, the player does not need to defeat every Final but a minimum number (depending on the level or circumstances) of Finals must be defeated. Yet, the undefeated Finals will come back again in the next levels Finals Phase in addition to that levels Finals. After the level is completed by the player game will enter an upgrade phase in which player will be able to upgrade his stats by spending coins on standard shop items or open chests by using a key and a chest which was collected in the previous level. The game will have a total of 4 levels and 4 upgrade phases.

2.1. Controls

The player can move around with W-A-S-D keys on the keyboard: W for going up, S for going down, A for left and D for right movement. Shooting and aiming will be done by using the mouse. Clicking on the left click shoots bullets around. Pointing the mouse on the screen will change the aim of shooting. Clicking the right mouse button will activate the deployable power-up and clicking the left mouse button will deploy the power up.

2.2. Levels

As mentioned before the game will contain 4 levels with increasing difficulty. (Smarter enemy AI, higher enemy health, enemies that hit harder) Each level will bring at least one different enemy type to the game mechanism. The player will try to defeat standard enemy types without running out of time. If the player does defeat the standard enemies the Finals will arrive, pushing game difficulty to higher. When the required minimum number of Finals are defeated player will earn the right to proceed to next level. The level can be completed in two ways; the first one being the defeating all the Final enemies which guarantee that there will be extra

Final enemies in the next level, the second one being defeating minimum required number of enemies which will result as undefeated Final enemies returning in the next levels Finals Phase.

2.3. Enemy Types

There will be various types of enemies that player will come across during the game. These enemies will be harmonious with our theme.

Bug: The most common and weak enemy type which will simply crash into the player to damage him. Yet, this crash will damage it as well. It cannot shoot bullets; it is only capable of crashing.

Assignments: Basically, tougher versions of bugs that will hit harder and sustain more damage.

Quiz: Quizzes can spawn very close to the player. It has average damage, can move around and has low health. It can also shoot bullets.

Lab: A slow-moving enemy type that will spawn bugs and assignments continuously unless it is destroyed. It has high health, it cannot shoot bullets.

Midterm: This enemy type is quite rare in comparison to others. It has more health than Labs and applies more damage by shooting bullets. It can move around. It is the most dangerous enemy type excluding Finals.

Finals: Finals come as a pack, unlike Midterms. Yet, they spawn when there are no other enemies left for that level. They have more health and apply more damage by shooting. They will have a shield additional to their health, which will regenerate unless they are kept shot at.

2.4. Credits, Upgrades, Keys, and Chests

The game will contain types of chests and keys. Keys will spawn in high-risk locations in-game arena and valuable chests will be dropped by the high-risk enemies when they are killed. There will be 4 types of chests.

Credits (Coins): Money that can be spent in the shop. Dropped by all enemies.

Freshmen Chest: Least valuable chest usually dropped by weaker enemies. Requires 1 key to open. It has 90% chance of giving a standard tier item, 7% chance of giving a rare tier item and 3% chance of giving an ultra-rare tier item.

Sophomore Chest: Usually dropped after the 1st level is completed. Requires 2 keys to open. It has 50% chance of giving a standard tier item, 30% chance of giving a rare tier item and 20% chance of giving an ultra-rare tier item.

Junior Chest: Usually dropped by the stronger enemies after 1st level. Requires 3 Keys to open. It has 30% chance of giving a standard tier item, 35% chance of giving a rare tier item, 25% chance of giving an ultra-rare tier item, 10% chance of giving a "hacker" tier item.

Senior Chest: Usually dropped by the stronger enemies after 2nd level. Requires 3 Keys to open. It has 18% chance of giving a standard tier item, 25% chance of giving a rare tier item, 32% chance of giving an ultra-rare tier item, 25% chance of giving a "hacker" tier item.

Keys: Appears in the game arena for 30 seconds at random moments (usually close to stationary or slow enemies). They can also be bought from the shop.

Standard Items: Gives player simple and small perks. Some of them can also be bought from the shop. Such as Increase in fire rate, speed, bullet, speed, time…

Rare Items: Gives player simple but better perks. They cannot be bought from the shop.

Ultra-rare Items: Gives player greater perks or they affect the overall mechanics of the game. Despite the features like standard items, it has an effect on game mechanics. Such as certain enemy types with less health, slower speed, less amount; more power-ups, keys or chests.

Hacker Items: Gives the best perks or rather affect the game in favor of the player. Yet the perks and effects of these items are so great that they all have a trade-off. Such as for reducing the minimum amount of finals to pass the level, it reduces player's fire rate…

2.5. Power-Ups

The game will offer a variety of power-ups that will appear in the arena in random moments and at random frequency. These power-ups will enhance the player temporarily. If a power-up is deployable after the pick up the player must right click to activate and left click again to deploy the power up.

Time: Time power-ups will give player additional time.

Slow-Time: Slows everything in the game by 20% for 10 seconds.

Bouncy – Bullets: Bullets bounce off (3 times per bullet) from the borders of the game are instead of simply going out.

Bullet-Blast: Sends out a circular group of bullet originating from the player.

Damage upgrade: Player bullets hit harder for 5 seconds.

3. Requirement Specification

3.1. Functional Requirements

3.1.1. Play Game

Survival in Bilkent is a 2D top to down shooter game. In the beginning of the game, player types his/her nickname which will be displayed under the player object during the game. The player can shoot by using the mouse and move by pressing W-A-S-D keys on the keyboard. Enemies have limited health which is dependent on their type. Player kills the enemies by shooting bullets. Bullets that hit the enemies decrease the enemy's current health by a damage value. The duration of each level are limited (due to player's time stat). In order to stay alive, the player has to kill a certain amount of enemies until the end of the level without running out of time. After defeating simple enemies player must face the Finals which are a pack of bosses. The player does not have to defeat them all just a minimum number that is specified. Yet un-defeated Finals will come back to the next level. Each hit dealt by the enemy causes the player to lose additional time. The player can use W for going up, S for going down, A for left and D for right movement. Shooting and aiming will be done by using the mouse. Clicking the left mouse button shoots bullets towards the mouse position. At the end of the levels, the player will enter an upgrade phase where he can purchase standard items, see his/her item inventory, collected chests, coins, and keys.

3.1.2. In-Game User Interface

In game interface will display;

- Remaining time in the right edge of the screen.
- The currently picked up power-up in the bottom right corner.
- Current score in the top right corner
- Current level on top edge
- A number of chests and keys in the top left corner.

3.1.3. Pause Game

During the game, the player can pause the game whenever he/she wants and then he/she can continue the game whenever he/she wants to continue. When the player presses the ESC, the game is paused and pause menu will appear offering three choices “Continue” which will unpause the game, “Settings” which will go to settings menu, “Quit” which ends the game terminating the current game session.

3.2. Settings Option

The player can mute and un-mute the sound of the game by using the settings option.

3.3. Non-Functional Requirements

3.3.1. User-Friendly Interface and Mechanics

The game can be played easily. In other words, the mechanics are generic and self-explanatory that seen in many 2-D shooter games.

3.3.2. Game Performance

The game must run smoothly even though there are many objects on the screen. Its average FPS must be around 30 since if it is larger than 30 it will look like a fast-forwarded video if it is less than 30 it will not be a playable game.

3.3.3. Extensibility

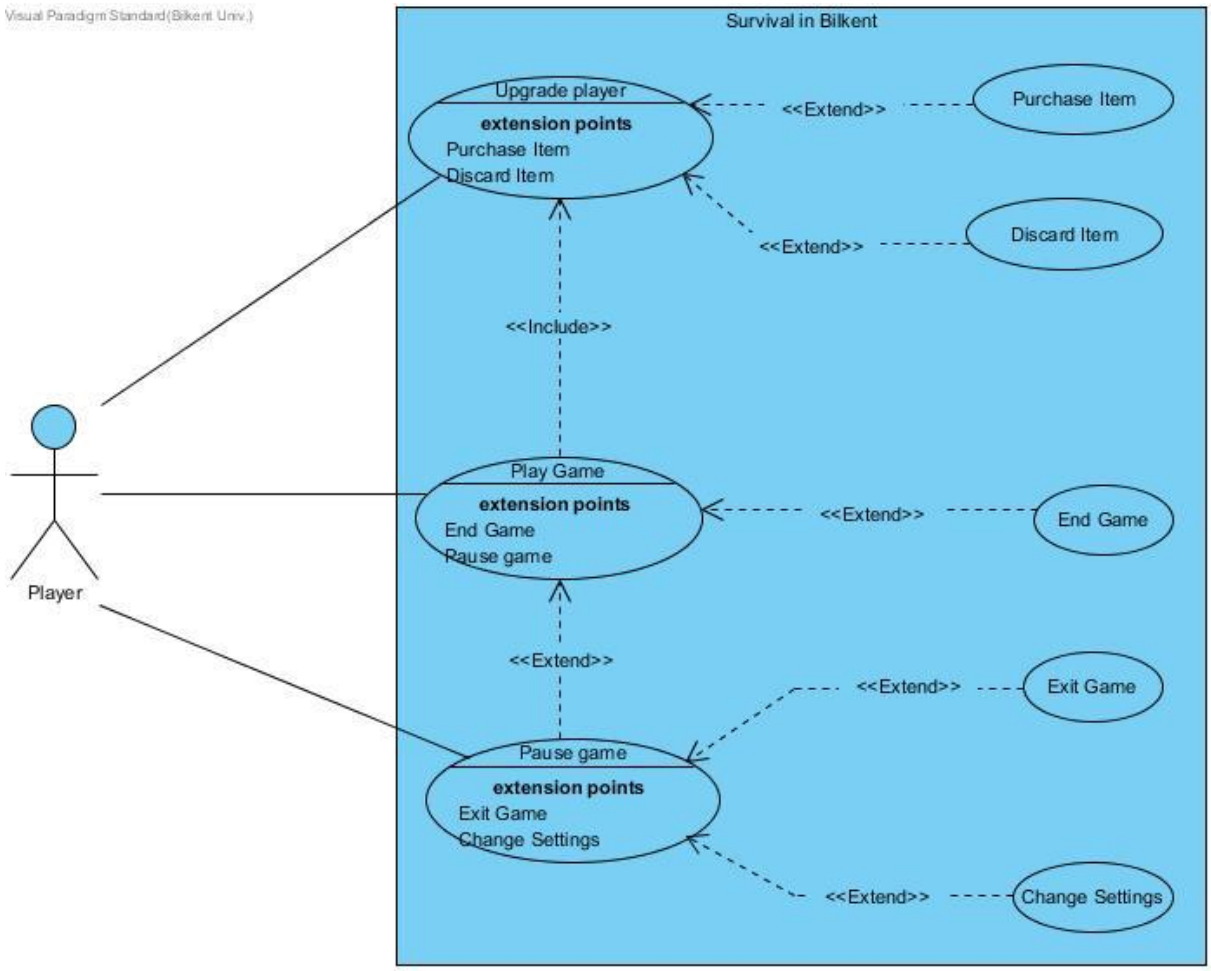
In software engineering, reusability and extendibility are two important properties of a project. Survival in Bilkent will be able to extend and re-used for future projects. Especially we will be adding new upgrade items to the game in future iterations.

3.4. Pseudo Requirements

The game will be implemented in Java. Since it will run on the Java Virtual Machine, the system is highly portable due to a number of machines which has Java. It has no other portability considerations. Graphics will be implemented using Java's swing and AWT libraries. (Java Version 8 Update 144)

4. System Models

4.1. Use Case Model



4.1.1. Use Case Descriptions

Use Case #1

Use case name: Play Game

Participating actors: Player

Entry condition: Player has opened the game and he is on the title screen.

Exit condition:
- Player has completed the all 4 levels successfully, OR
- Player has lost his health (time) in any level, OR
- Player has chosen to exit the game via the pause menu.

Main Flow of Events:

1. Player launches the game
2. Player enters his in-game-name.
3. Upgrade screen appears.
4. The system prepares the level.
5. Player completes all the levels (Upgrade phase between levels).
6. The system displays the score of the player.
7. Player returns to the title screen.

Alternative Flow of Event:

1. Player loses his health (time) in any of the levels. (Game Over screen appears.)
2. Player exits using exit option in the pause menu.

Use Case #2

Use case name: Game Over

Participating actors: Player

Entry condition: Player is already playing the game. His/her time bar becomes empty to represent that the time is up.

Exit condition: Player returns to title screen.

Main Flow of Events:

1. Player consumes his/her time by spending time or taking enemy damage.
2. Game over screen will appear and player sees his/her score.
3. Player returns to title screen.

Use Case #3

Use case name: Pause Game

Participating actors: Player

Entry condition: Player is already playing the game. Player presses the Esc character to pause the game.

Exit condition: Player either return to game or exits the game

Main Flow of Events:

1. Player presses the pause button on the keyboard.
2. The game is paused, and the pause menu window displays over the now frozen game screen.
3. Player returns to the game if he/she chooses to continue the game.

Alternative Flow of Event:

1. Player exits using exit option in the pause menu without any changes.OR
2. Player goes to the settings menu using settings button on the pause menu.

Use Case #4

Use case name: Change settings

Participating actors: Player

Entry condition: Player is already paused the current game. Player chooses the "settings" button on the pause menu.

Exit condition: Player either return to game or exits the game.

Main Flow of Events:

1. Player presses the Esc button on the keyboard.
2. The pause menu window displays.
3. The player chooses the "settings" option.
4. Player adjusts the options that suit to him/her.
5. The system saves changes.
6. Player returns to the game if he/she chooses to continue the game

Use Case #5

Use case name: Upgrade Player

Participating actors: Player

Entry condition: Player has already launched the game and he either started a new game or just completed a level (except level 4).

Exit condition: Player chooses the "Done" button in the right bottom corner of the Upgrade screen.

Main Flow of Events:

1. Before each level of the game, upgrade screen appears.
2. The Chests (freshmen, sophomore, junior, and senior) may be opened by using keys. Chests will give an item when opened.
3. Player can use their credits (initial or collected) to purchase new items(only some of the Standard tier items and keys) to the character's item inventory.
4. Items (rare, ultra-rare, hacker) can either be discarded to make room for better items (player has a limited amount of slots) or can just be placed in the inventory. **OR**

5. Player can go back and use his purchased keys to open another chest which was already in their inventory.
6. Player returns to the game screen by clicking “Done”.

Alternative Flow of Event:

1. Player may exit the game.

Use Case #6

Use case name: Purchase item

Participating actors: Player

Entry condition: Player has already launched the game and he is on the shop screen.

Exit condition: Player returns upgrade screen.

Main Flow of Events:

1. Before each level of the game, upgrade screen appears.
2. Player clicks on shop button on the upgrade screen and shop screen appears.
3. To buy something player uses his/her credits as money.
4. Player can buy keys (to open chests), and standard items.
5. After shopping the purchased products can be used for the upgrade character.
6. Player returns to the upgrade screen.

Use Case #7

Use case name: Discard item

Participating actors: Player

Entry condition: Player has already launched the game and he is on the shop screen.

Exit condition: Player returns upgrade screen.

Main Flow of Events:

1. Before each level of the game, upgrade screen appears.
2. Player clicks on shop button on the upgrade screen and shop screen appears.
3. To make room for the new items player discards his/her own items by clicking right click of the mouse.
4. Player returns to the upgrade screen.

Use Case #8

Use case name: Exit game

Participating actors: Player

Entry condition: Player is already playing the game. Player presses the pause button on the keyboard to exit the game.

Exit condition: System terminates the game.

Main Flow of Events:

1. Player press pause button on the keyboard.
2. The pause menu window displays.
3. Player chooses the "Exit" option.
4. Another window appears asking the player if he/she is sure or not.
5. Player exits from the game if he/she chooses "yes".

Alternative Flow of Event:

1. Player can choose "No" and return to the pause menu.

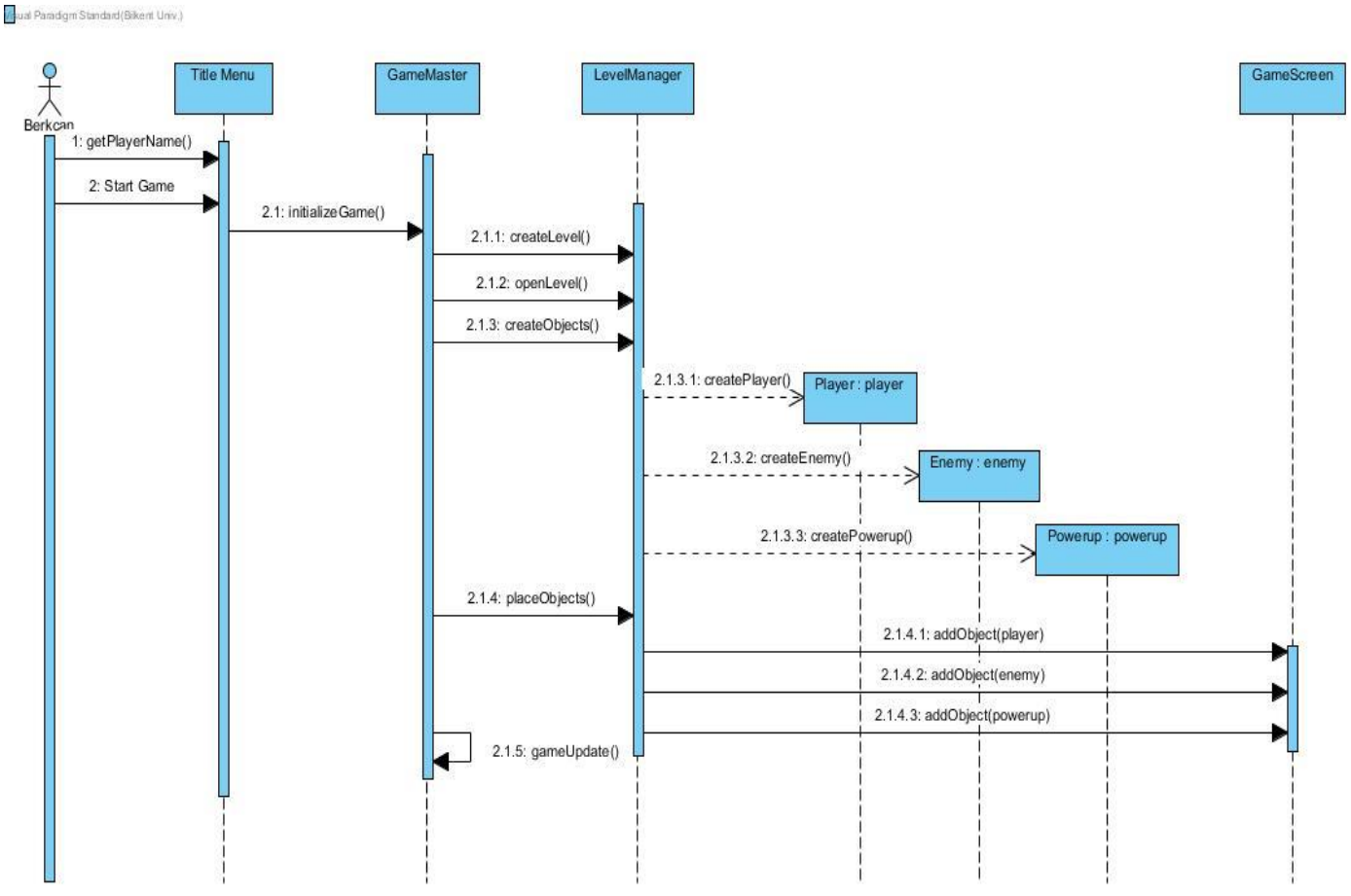
4.2. Dynamic Models

This section describes the Survival in Bilkent game in detail by illustrating some of the crucial situations as scenarios in sequence diagrams. Player and the enemies are the core parts of this game thusly their behaviors will be provided in state diagram form. Also, the activities of the system (game master) will be stated in the activity diagram.

4.2.1. Sequence Diagrams

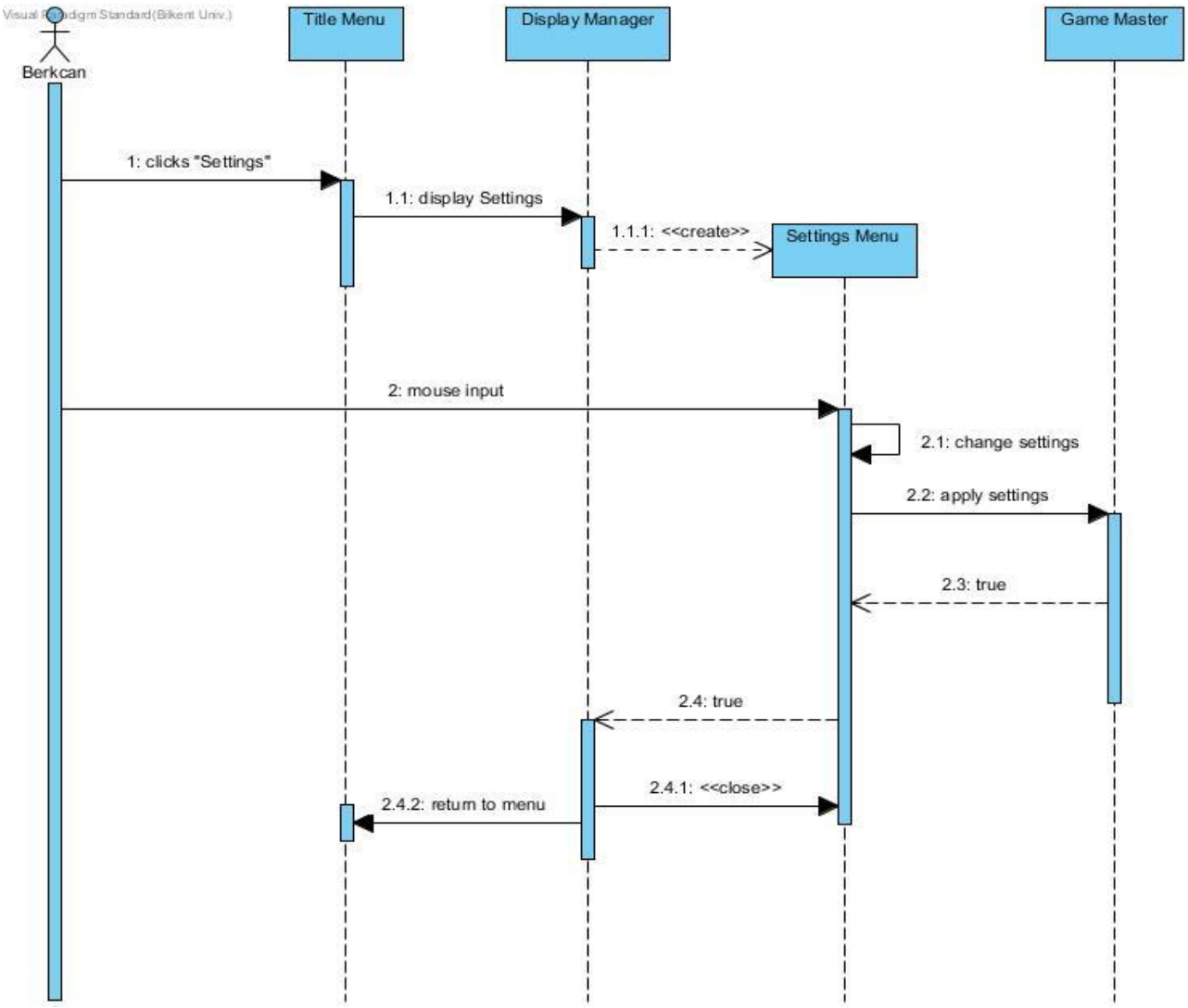
4.2.1.1. Start Game

Scenario: Player Berkcan is bored and after entering his name in the text box starts the game by pressing the “Start” button on the title screen. After the first upgrade phase which lets Berkcan choose his initial start items system constructs the level by reading a text file which specifies the contents of the current level. He sees the main Game Screen.



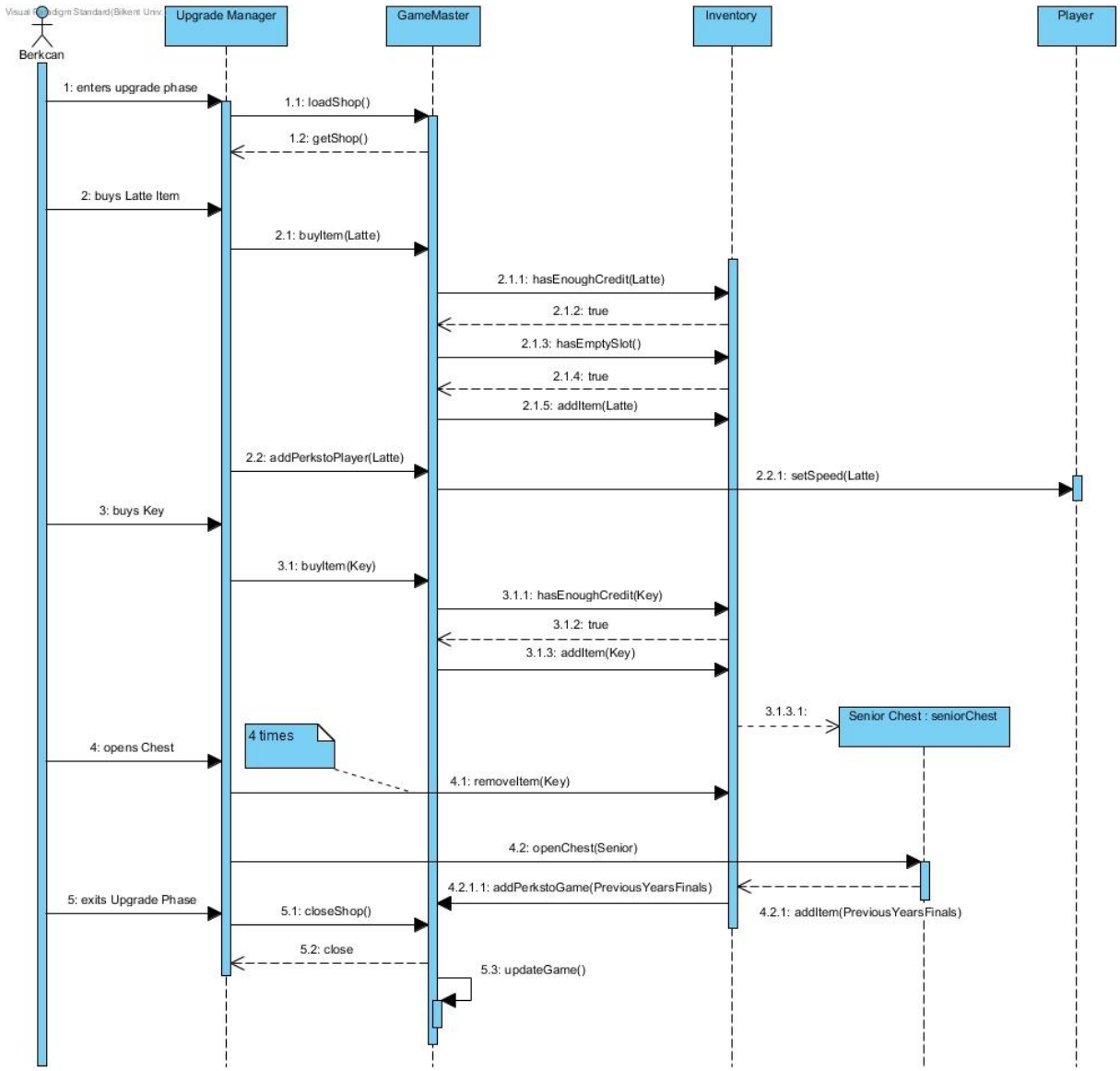
4.2.1.2. Change Settings

Scenario: Player Berkcan decides to change settings of the game before he starts playing. He clicks “Settings” from the main menu and game opens up the Settings Menu. Berkcan changes the settings he wishes to and closes the Settings Menu. That returns him to the main Title Menu.



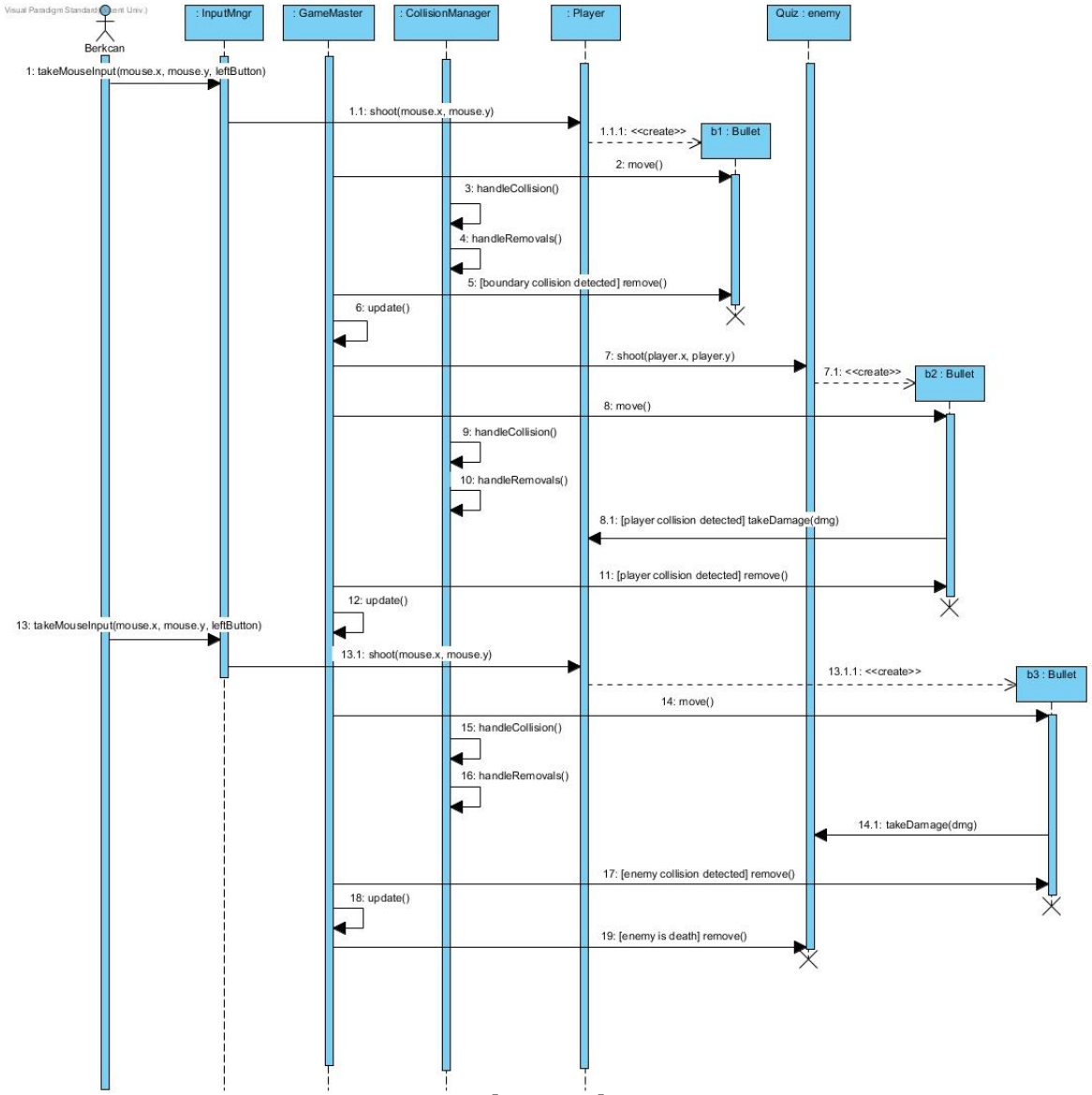
4.2.1.3. Player Upgrade

Scenario: Player Berkcan completes the 2nd level the player enters the upgrade phase. After that Berkcan has enough credit to buy a “Latte” item which increases his player speed. Then, he sees that he has 3 keys so he buys another one from the shop and he tries to open the Senior Chest that he obtained from the previous level. The chest opens and gives him a “Hacker Type” item “Previous Year’s Finals” which reduces the health of “Finals” by 35%. “Previous Year’s Finals” will be added to his inventory. After his transactions are finalized Berkcan clicks “done”. The next level will start accordingly to his new items.



4.2.1.4. Player Shooting

Scenario: Player Berkcan has already in the game. Player Berkcan shoots two bullets to the enemy whose type is quiz. One of the bullets hits the quiz enemy and the other one misses the enemy. At the same time, the quiz enemy also shoots Berkcan. The enemy's bullet hits Berkcan and disappears (is removed from the bullet list). After that Berkcan takes damage and his health decreases. The first bullet of Berkcan which did not hit the quiz enemy keeps going until the border of the game and disappears when it hits the border (is removed from the bullet list). The second bullet hits quiz enemy and then the enemy takes damage and gets killed. After that, the killed enemy is removed from the enemy list.



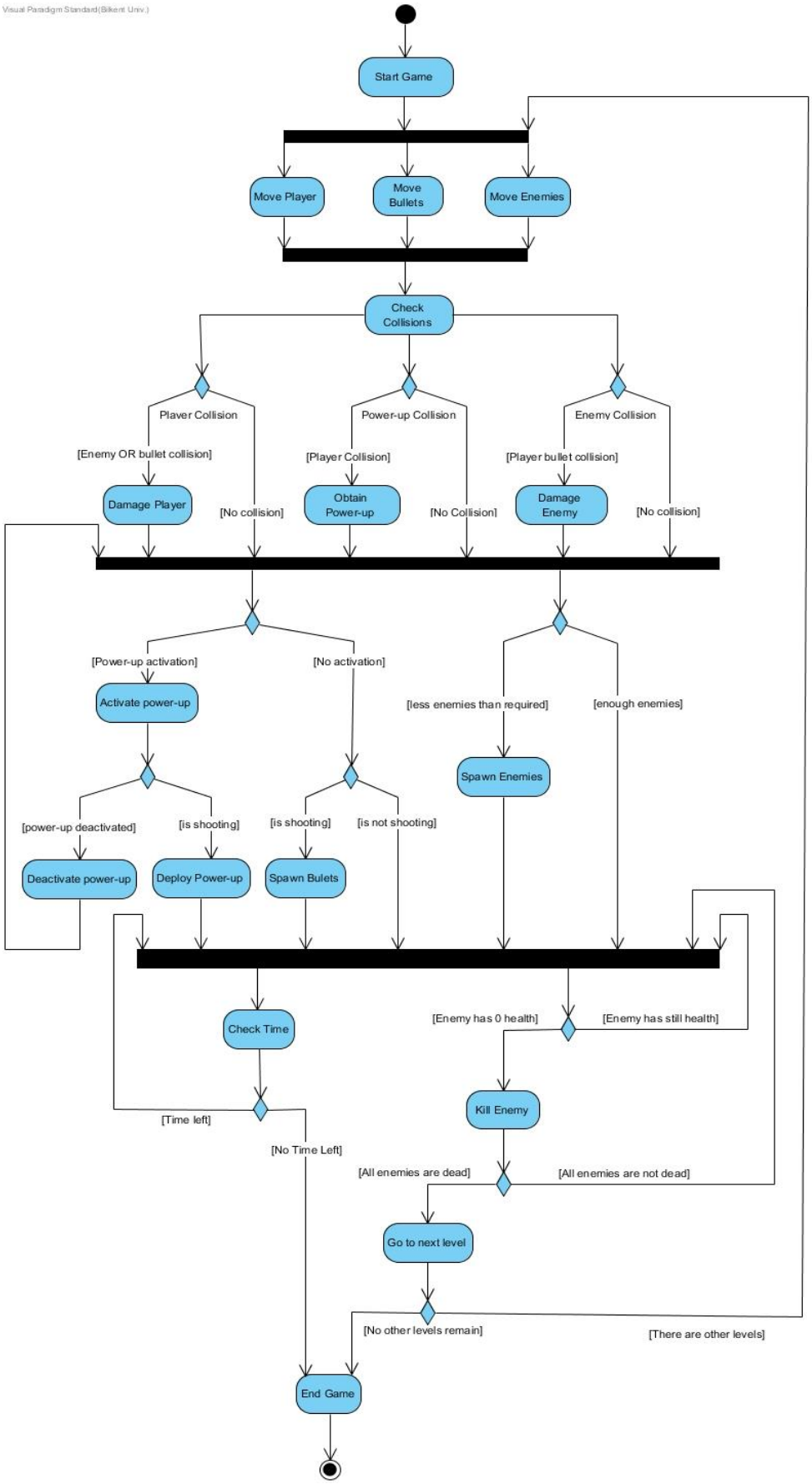
4.3. Activity Diagrams

The game starts when the user clicks on “Start” button on the title screen. When the game is started, the system initializes the game and creates the enemies, player, and their bullets on the game screen.

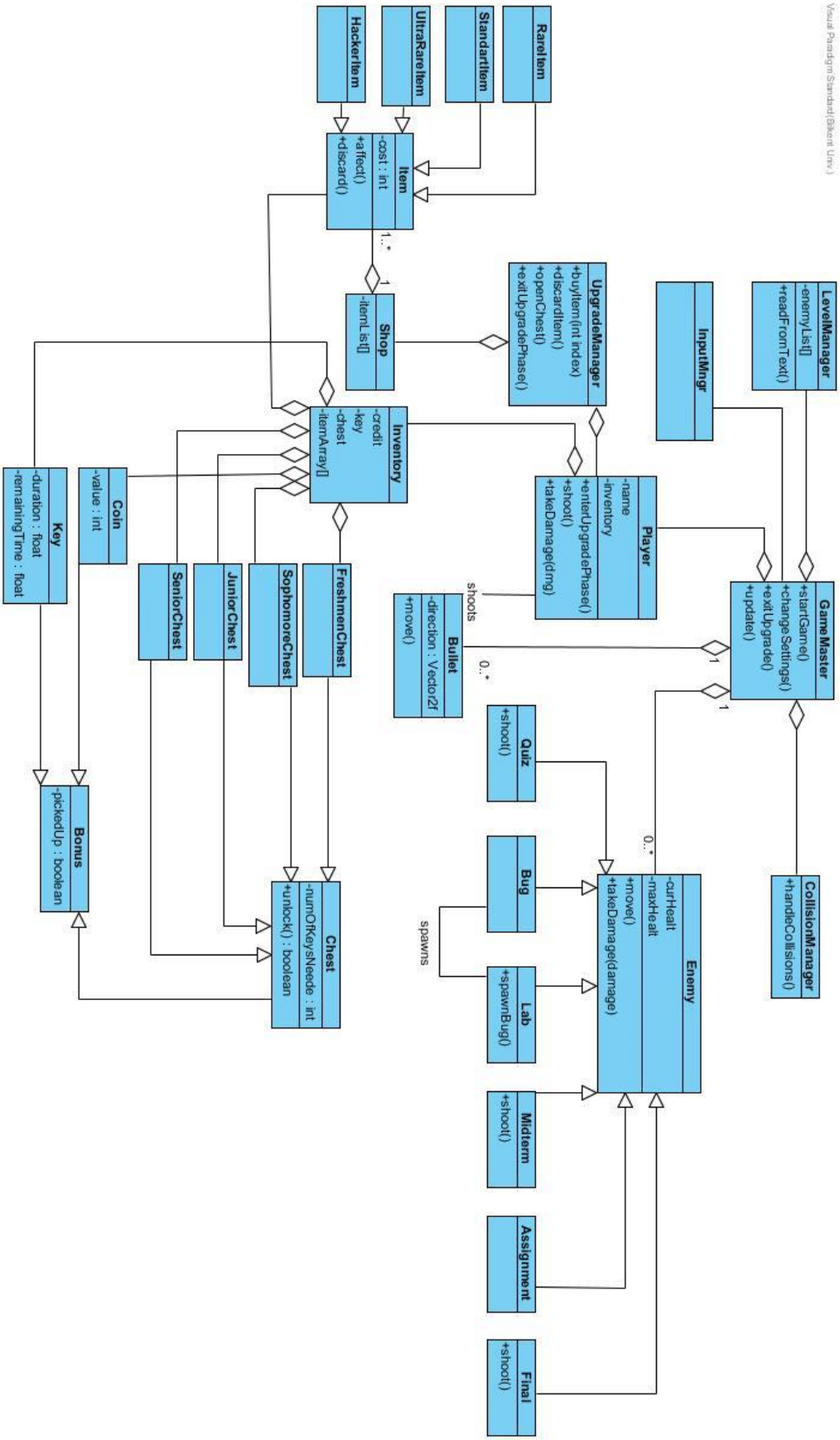
In the next stage, the system controls every collision in the game. If the bullets collide with the player, the player takes damage and resumes play. If the player collides with the power-up he will obtain it to use in the game. If the player collides with the enemy, he/she will damage the enemy. Otherwise, the game continues normally.

In the next step, system check that whether the player has available power-ups or not and if there are enough enemies in the game or not. If the player has an active power-up, it is checked whether he uses it or not; inactive if it is used, or inactive if it is not. If there is no active power-up, spawning bullet continues. If there are not enough enemies in the game, new enemies are spawned; if there is enough enemy the game continues.

In the last step, the system checks whether the time (life) is not over and whether the enemies die. If there is enough time, the system continues to check the time, if the time is over, the game ends. If the enemy has enough life, the enemy will continue the game; if the enemy’s life is over, it will die. If all enemies are not killed in the game, the game continues; if all enemies are killed, the new level is passed. If there exists a new level, this all process starts again, if all the levels are finished and the new level is not available, the game ends.


Visual Paradigm Standard(Bilkent Univ.)
Start Game
Move Player
Move Bullets
Move Enemies
Check Collisions
Player Collision
[Enemy OR bullet collision]
Damage Player
[No collision]
Power-up Collision
[Player Collision]
Obtain Power-up
[No Collision]
Enemy Collision
[Player bullet collision]
Damage Enemy
[No collision]
[Power-up activation]
Activate power-up
[No activation]
[power-up deactivated]
Deactivate power-up
[is shooting]
Deploy Power-up
[is shooting]
Spawn Bulets
[is not shooting]
[less enemies than required]
Spawn Enemies
[enough enemies]
Check Time
[Time left]
[No Time Left]
[Enemy has 0 health]
Kill Enemy
[Enemy has still health]
[All enemies are dead]
[All enemies are not dead]
Go to next level
[No other levels remain]
[There are other levels]
End Game

4.4. Object and Class Models



5. User Interface

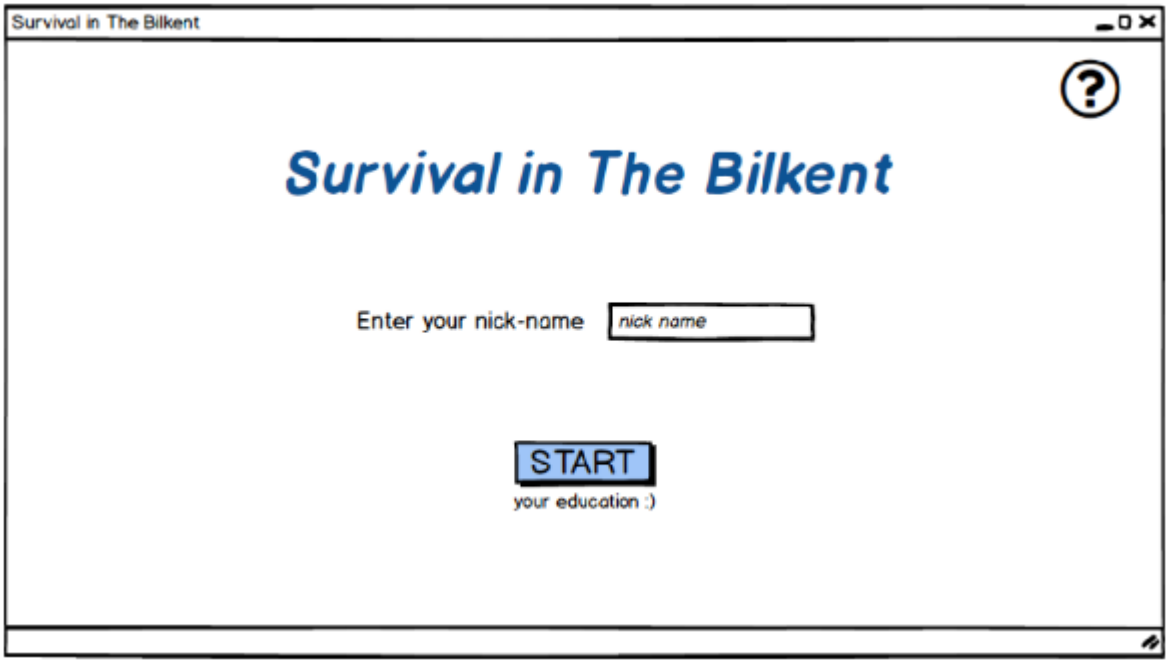
5.1. Main Screen

The main screen includes ‘enter your nickname’ bar, start button and question mark.

‘Enter your nickname’ bar: At the beginning of the game, the user enters his/her name.

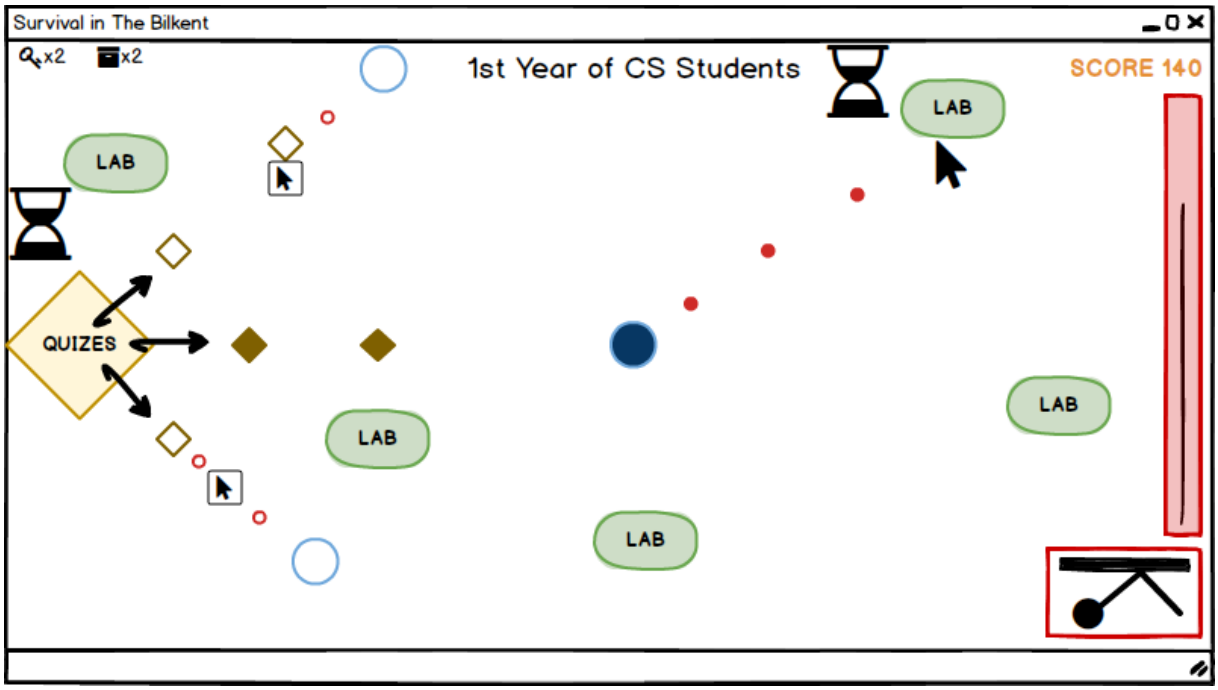
Question mark button: The player can get information that about the game’s rules.

Start button: The player can start the game by clicking this button.



5.2. Game Screen

In the game screen, the player can see his/her remaining time, score, key number, chest number, and power-ups.



5.3. Pause Screen

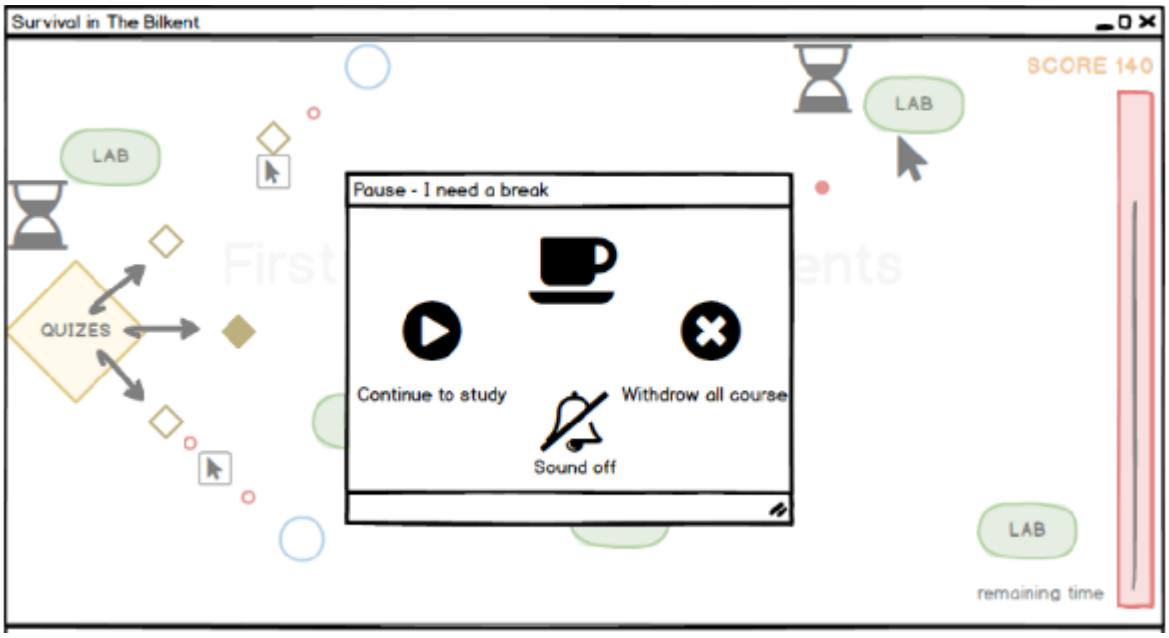
In the pause screen, the player sees a text which is 'Pause- I need a break', 'continue to study' button, 'withdraw all course' button, 'sound off' buttons, and coffee symbol.

'Continue to study' button: After stopping the game, the player can continue to the game by clicking this button.

'Withdraw all courses' button: after stopping the game, the player can close the game.

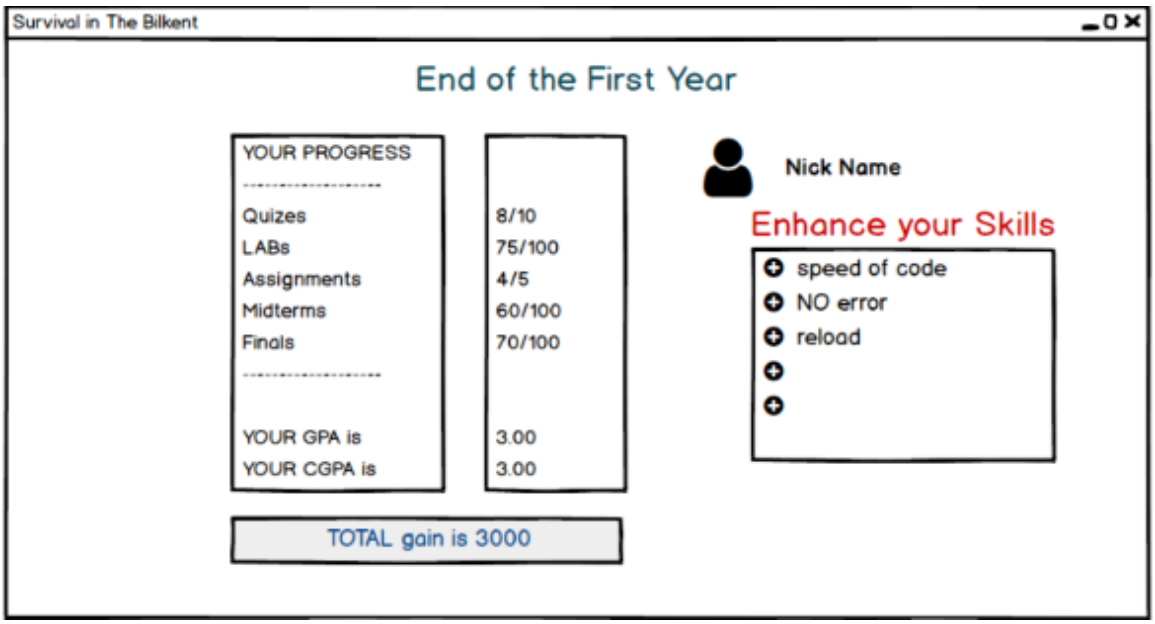
'Sound off' button: The player can turn off the sound of the game.

Coffee symbol: It does not have any function. It is stated in the pause screen as a design purpose.



5.4. Result Screen

In the result screen, the player sees a title which is 'End of the First Year' and an information table called 'Your Progress'. In this information table, quizzes' points are shown out of 10, labs', midterms', and finals' points are shown out of 100, assignments' points are shown out of 5, GPA and CGPA are shown out of 4.00. Total gain is also shown on the result screen. Nickname is shown above 'Enhance your skills' table. In the 'Enhance your skills' table some items are shown such as 'speed of code', 'no error', 'reload'.



6. Conclusion

We prepared an analysis report to implement our project. In this report, we described our project under 5 main subtitles which are the introduction, overview, requirement specification, system model, and user interface.

In the introduction and overview parts, we describe the project briefly.

In requirement specification part, we determined the main requirements of the project both in functional and non-functional.

In system, model part includes 4 parts which are the followings;

1) Use case diagram
2) Dynamic diagrams
3) User interface

In use case diagram, we designate use cases for the main actions of the game.

In the dynamic diagram, we designated the sequence diagram and activity diagram. We found the actions of the game which are found between the game and the player. After the dynamic diagram, we designated the activity diagram which briefly shows the gameplay flow. In the user interface, we designed game screens that are seen when the player plays the game. Consequently, we tried to do the detailed description of our game in order to do better design for the project.

References:

Inspired by the game in the following URL:

http://diep.io/