



Gürkan KARAHAN

Onur Can ÜNER

Serkan TUNA

Kader BELLİ

SOFTWARE REQUIREMENTS SPECIFICATION

HEDEF4YILDIZ

Table of Contents

| | |
|--|----|
| Table of Figures | 3 |
| 1. Introduction | 4 |
| 1.1 Problem definition | 4 |
| 1.2 Purpose | 4 |
| 1.3 Scope | 4 |
| 1.4. Definitions, acronyms and abbreviations | 4 |
| 1.5 References | 5 |
| 1.6 Overview | 5 |
| 2. Overall Description | 5 |
| 2.1 Product Perspective | 6 |
| 2.1.1 User Interfaces | 6 |
| 2.1.2 Hardware Interfaces | 7 |
| 2.1.3 Software Interfaces | 7 |
| 2.1.4 Communication Interfaces | 7 |
| 2.1.5 Memory | 7 |
| 2.1.6 Operations | 8 |
| 2.1.7 Site Adaptation Requirements | 8 |
| 2.2 Product Functions | 8 |
| 2.3 User Characteristics | 10 |
| 2.4 Constraints | 10 |
| 2.5 Assumptions and Dependencies | 10 |
| 3. Specific Requirements | 11 |
| 3.1 Interface Requirements | 11 |
| 3.1.1 Game Website | 11 |
| 3.1.2 Welcome Screen | 12 |
| 3.1.3 New Player Screen | 12 |
| 3.1.4 Main Menu | 12 |
| 3.1.5 Game Settings | 12 |
| 3.1.6 Game Instructions | 12 |
| 3.1.7 User Profile | 12 |
| 3.1.8 Character Profile | 13 |
| 3.1.9 Training Mode | 13 |
| 3.1.10 Match Mode | 13 |
| 3.1.11 Tournament Mode | 14 |

| | |
|---|----|
| 3.2 Functional Requirements | 14 |
| 3.2.1 Functional Requirement 1 | 14 |
| 3.2.2 Functional Requirement 2 | 14 |
| 3.2.3 Functional Requirement 3 | 14 |
| 3.2.4 Functional Requirement 4 | 14 |
| 3.2.5 Functional Requirement 5 | 14 |
| 3.2.6 Functional Requirement 6 | 15 |
| 3.2.7 Functional Requirement 7 | 15 |
| 3.2.8 Functional Requirement 8 | 15 |
| 3.2.9 Functional Requirement 9 | 15 |
| 3.2.10 Functional Requirement 10 | 15 |
| 3.2.11 Functional Requirement 11 | 15 |
| 3.2.12 Functional Requirement 12 | 15 |
| 3.2.13 Functional Requirement 13 | 15 |
| 3.2.14 Functional Requirement 14 | 15 |
| 3.2.15 Functional Requirement 15 | 15 |
| 3.2.16 Functional Requirement 16 | 16 |
| 3.3 Non-functional Requirements | 16 |
| 3.3.1 Performance Requirements | 16 |
| 3.3.2 Design Constraints | 16 |
| 3.3.3 Software System Attributes | 16 |
| 3.4 Logical Database Requirements | 17 |
| 4. Data Model and Description | 18 |
| 4.1 Data Description | 18 |
| 4.1.1 Data objects | 18 |
| 4.1.2 Data dictionary | 19 |
| 5. Behavioral Model and Description | 19 |
| 5.1 Description for software behavior | 19 |
| 5.2 State Transition Diagrams | 20 |
| 6. Planning | 20 |
| 6.1. Team Structure | 20 |
| 6.2. Estimation | 21 |
| 6.3. Process Model | 21 |
| 7. Conclusion | 22 |

Table of Figures

| | |
|--|----|
| Figure 1 – Context Diagram | 6 |
| Figure 2 – Use Case Diagram for User | 8 |
| Figure 3 – Use Case Diagram for Player..... | 9 |
| Figure 4 – Use Case Diagram for Server | 10 |
| Figure 5 – Training Mode User Interface..... | 13 |
| Figure 6 – Match Mode User Interface | 14 |
| Figure 7 - Revenge Entity Relationship Diagram | 17 |
| Figure 8 - Revenge Class Diagram..... | 19 |
| Figure 9 – REVENGE State Chart Diagram | 20 |

1. Introduction

This document is the first official report of our term project. Our team have spent most of the time for analyzing and learning the technologies that will be used in our project so some parts (such as interface requirements) will be introduced very briefly in this report.

This section is introduced to give initial information for the users of SRS and starts with introduction section which includes problem definition, purpose of the project, scope of the project, definitions acronyms and abbreviations, references and overview subheadings.

1.1 Problem definition

Most of the people spends great amount of time to play computer games on PC and it is obviously not deniable that this situation affects the health of those people in a bad way. First of all they spend hours and hours sitting in front of the display without moving and this leads to obesity. In addition to that health problem, it also causes more problems as posture disorders, muscle and backbone problems. We can count more but the most important problem is blood circulation and this problem can even lead to the death of people. Our team wants to solve this problem by changing the people's perspective on computer games in a good way by making it possible to play our game without damaging their health.

1.2 Purpose

Purpose of this document is giving the detailed information about our project by specifying the internal and external requirements, constraints, assumptions and dependencies and introducing initial plan that will be followed by our team.

The problem that our team wants to solve is removing the bad effects of computer games on people's health by making it possible to play our game with actual movements instead of the sitting and playing with some hardware such as keyboard or joystick.

1.3 Scope

We are aiming to release two connected products: game application and website which provides registering to our game for user. Our game will support different hardware options for playing, especially we are coming with new concept that our game will support Kinect hardware on PC. REVENGE will be a game which can be played on network from different areas. Players can buy the characters that come from the history of game and they can improve the characters they bought. There will be 3 different modes and 2 different user controllers. These modes are: training mode, match mode and tournament mode. And the controller options are: keyboard and Kinect.

1.4. Definitions, acronyms and abbreviations

| | | |
|------------------------|---|---|
| Unity3D | : | The game engine that will be used to develop our game. |
| Kinect | : | It is the motion sensing input device that is created by Microsoft Team to realize human movements without controller. That hardware will be the option for the players to play our game. |
| OpenNI | : | It is the open source SDK that will be used to interact with the natural interaction device that is Kinect in our project. |
| PrimeSense NITE | : | NITE is middleware that is found by PrimeSense. NITE provides modules for OpenNI to handle gesture and skeleton tracking. |
| Mbit | : | It is the unit of computer storage or data transportation and it is equal |

to 10^6 (1000000) bit.

DBMS : Data Base Management System is simply program that is designed to manipulate the collection of data in database (DB).

DB : Database is organized and structured collection of data that will be manipulated by some database management system (DBMS).

MySQL : It is a database management system (DBMS) that is planned to be used in our project.

1.5 References

The resources listed below were the important sources of information for our initial research.

- IEEE Standard Documents:
IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications
- Unity3D Game Engine:
<http://unity3d.com/>
- Kinect Driver and SDK:
<http://zigfu.com/>
- Official Kinect Driver and SDK:
<http://www.microsoft.com/en-us/kinectforwindows/develop/learn.aspx>
- Asynchronous event-driven network application framework
<http://netty.io/>
- Massive Multiplayer Server SDK:
<http://www.smartfoxserver.com/>

1.6 Overview

Overall description part is explained in section 2 which gives general information about project and explains what the general factors that affect our product and requirements of our products are. There are 4 subsections are included in part 2 which are product perspective, product functions, constraints, assumptions and dependencies.

At section 3 specific requirements with sufficient details for designers and testers are explained. There are 3 main subheadings included in that section that are interface requirements, functional requirements and nonfunctional requirements.

Data model and description is explained in section 4.

In section 5 the behavioral model of the software is presented and this is supported with state diagrams.

In section 6 the information about developer team can be reached. This section explains the structure of team, which processing model is used, estimations and tentative schedule briefly.

Conclusion part is in section 7.

Supporting information can be found in section 8 which makes the SRS easier the use. Structured information that are table of contents, indexes and appendixes are provided in this section.

2. Overall Description

This section of the Software Requirements Specifications report describe general factors that affect the product and its requirements by providing a background.

2.1 Product Perspective

The product is a multiplayer online game designed to be played via Kinect sensor. It can be categorized as a Massively Multiplayer Online Game (MMOG), which is one of the most popular game types today. It is planned to be distributed as a free-download game over the game website. The game is not a component of another larger system; in other words, it is an independent product. Being a self-contained system, the general structure of the system consists of a user, whose movements are sensed via a Kinect sensor, the user interface that displays inputs of the sensor on the computer screen, and a server to maintain communication between the user and the game database. The system is visualized by the context diagram in Figure 1.

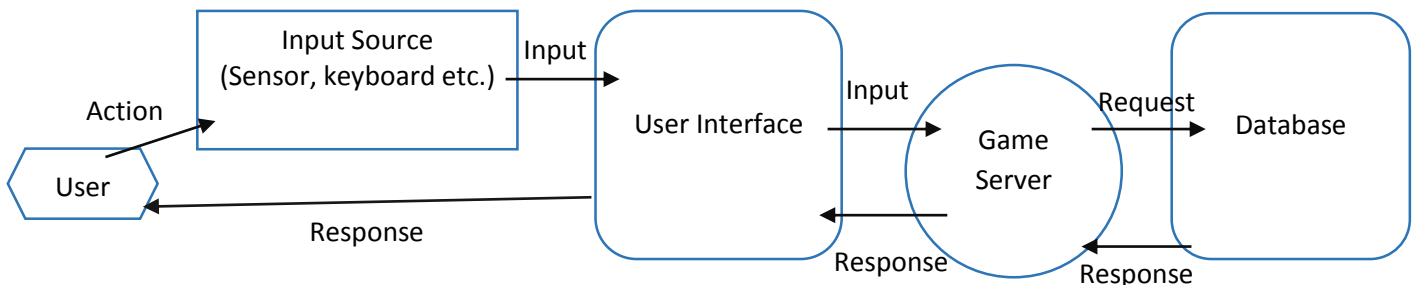


Figure 1 – Context Diagram

2.1.1 User Interfaces

User interfaces are of two main types: Game website interfaces and application interfaces. Game website will be designed with the main concern of attracting new visitors' attention, and making them excited about the game. To this end, it contains attractive visuals, funny fonts and easy-to-use interface, so that users can easily find what they look for. Similarly, the application window will be designed concerning usability.

Characteristics of user interfaces are described below, briefly. Detailed explanations supported by visuals are available in section 3.1.

- The game website is where the game is promoted and distributed. It is opened on a web browser and has several menus, which are described later in this document.
- The application interface is opened after the user downloads and installs the game to his/her computer.
- The application is displayed on a full screen window by default, but the user has the option to turn the full screen display on and off.
- Application starts with a log in screen. User cannot reach content of the application without logging in.
- After the user's first login to the application, the *new player scene* is opened to make the user familiar with main controls on the window and to specify short instructions about the game. Details of the *new player scene* are described in Section 3.1.3.
- User's passing between different menus is performed through buttons in *main menu*.
- There is a pull-down menu on each screen of the application interface which contains options to forward the user to the *main menu* and game settings, or to log out of the game.
- There are two profile menus: *User profile* and *character profile*. On *user profile* menu, the user manages his/her game account and follow his/her status in the game. On *character profile* menu, the user observes characteristics and status of his/her characters in the game, and makes character selection for the next game.
- There are menus for game instructions and settings.

- There are specific interfaces for the three game mode, namely *training mode*, *match mode*, *tournament mode*.
- In case of invalid entries, short warnings are displayed on the screen, and the user is guided for the correct action.
- All interfaces are in English; however, as new versions of the game becomes available, language support is to be added.

2.1.2 Hardware Interfaces

The hardware interface is composed of Kinect sensors, personal computers and game servers. Communication between Kinect sensors and computers is achieved by using a Kinect SDK. Communication between computers and servers is performed through a network connection.

2.1.3 Software Interfaces

Interaction of the product with external software products is described below.

- *Unity3D* game engine is used to design application interfaces and game environment. The preferred version of the software is Unity 4.5.x. The reasons why Unity 3D is used for development of the product are that it facilitates programmers' control over graphics and design issues, and it is a long standing software which comes with many sources and manual documents.
- The game is developed in Windows operating system. However, it will be designed to be compatible for all operating systems.
- *MySQL* is used as the database management system. A database management system is required to hold and manage information on users and matches. MySQL is preferred since it is a long standing open source database.
- *PrimeSense NITE* and *OpenNI* SDK are used to achieve communication between Kinect sensor and the development environment, i.e. Unity. These tools has built-in functions to get inputs from Kinect as joint coordinates, which facilitates the sensor perception issues a lot.

2.1.4 Communication Interfaces

Communication between the client application and the server is provided by UDP (User Datagram Protocol). The reason why UDP is selected as the Internet protocol is it's having better time performance than other protocols, which is the key point for a multiplayer online game. However, UDP does not guarantee synchronized and lossless communication between the two ports, which may affect the flow of the game, and cause unexpected behaviors of characters. To handle this problem, the same packages are to be transferred more than once in order to avoid data loss on the protocol. In addition, each package contains a data structure which holds information specifying transfer order of the package. Client uses the data structure to process packages in correct order.

2.1.5 Memory

The game requires a minimum of 4 GB primary memory to be played at an acceptable performance. At an 8 GB of primary memory, the user can expect optimal performance. The secondary memory requirements are not specified, for now. It varies according to quality of game models, which are not accurately determined, yet.

2.1.6 Operations

The game has three play modes: Training mode, match mode and tournament mode. There is the *Settings* operation, where the user arranges visual and audio related properties of the interface. In addition, there are operations which are performed on the game website, rather than the application. Website operations are rarer than the operations which are performed on the application interface. All operations are assumed to be interactive.

Mode of operations are described in *User Interfaces* section (Section 2.1.1), and brief descriptions of operations are available in *Product Functions* section (Section 2.2).

2.1.7 Site Adaptation Requirements

In order for the game to be played as expected, a Kinect sensor is expected to be connected to user's computer. Otherwise, user has the option of playing with keyboard, as well. The operating system should support Unity game platform. Additionally required properties will come and set with the installation package of the game.

2.2 Product Functions

While describing product functionality, we define three actors for the product: User, Player and Server. User is the person who has not logged in to the game, yet. Once a User logs in, it is defined as a Player, who is actually involved in the application, and capable of playing the game, making decisions and settings. Server is the actual game server, which achieves communication between users and database.

Product functions are described in this section, briefly. Step by step explanation of each use case is available in *Functional Requirements* section (Section 3.2).

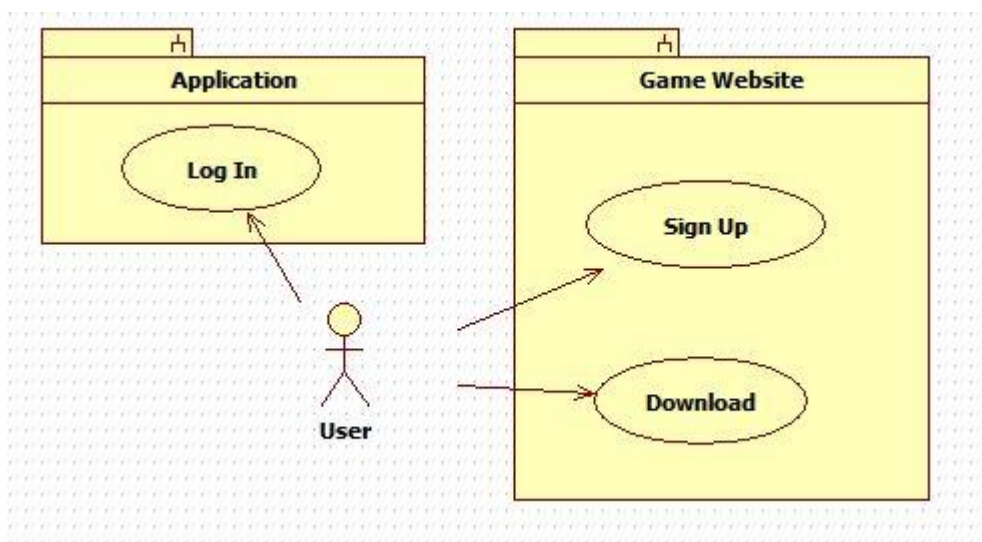


Figure 2 – Use Case Diagram for User

| Actor | Use Case | Description |
|-------|----------|--|
| User | Sign Up | An unregistered User is registered to the game. |
| User | Download | The User downloads the game to his/her computer by following the instructions provided in the website. |
| User | Log In | A registered User connects to the game server by entering his/her user information. |

Table 1 – Use Case Descriptions for User

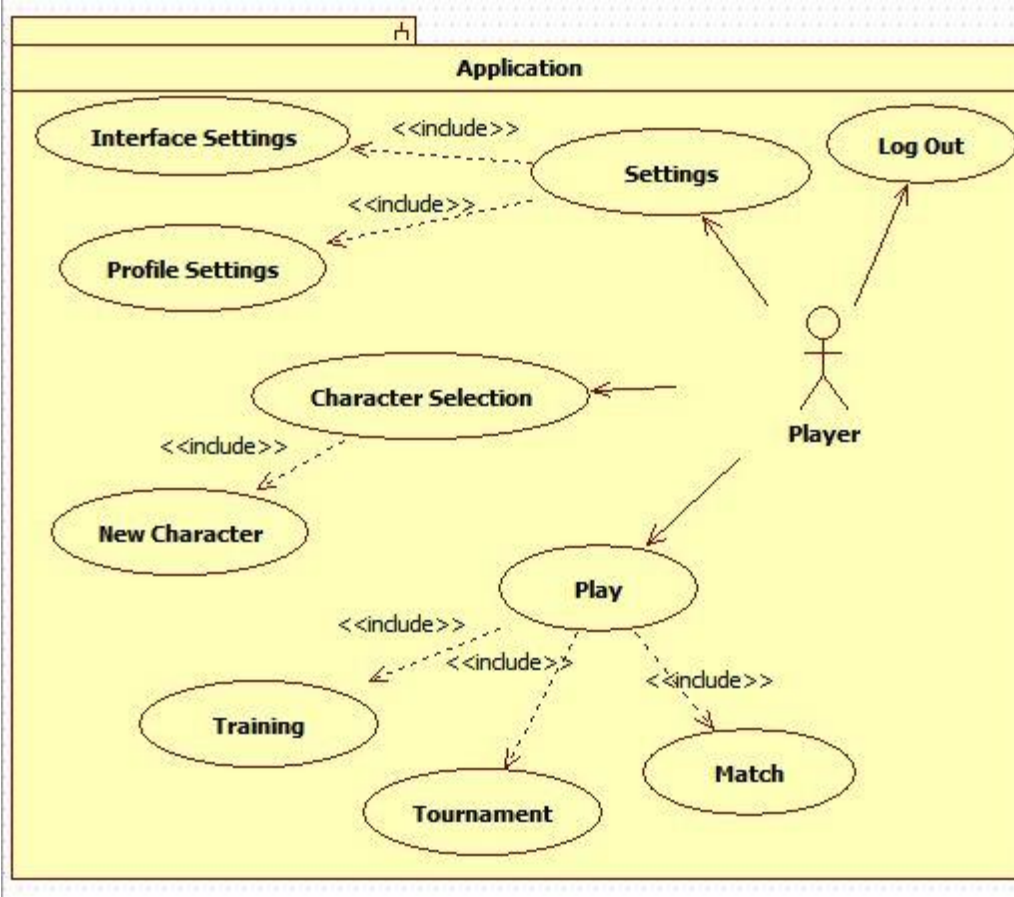


Figure 3 – Use Case Diagram for Player

| Actor | Use Case | Description |
|--------|---------------------|--|
| Player | Settings | The Player edits application’s display settings, and his/her account settings. |
| Player | Character Selection | The Player chooses one character among his/her characters in the game. The selected character is used in the following game play. The Player can also purchase new characters. |
| Player | Play | The Player chooses a game mode and starts playing with the last selected character. |
| Player | Log Out | The Player logs the system out, and is disconnected from the server. |

Table 2 – Use Case Descriptions for Player

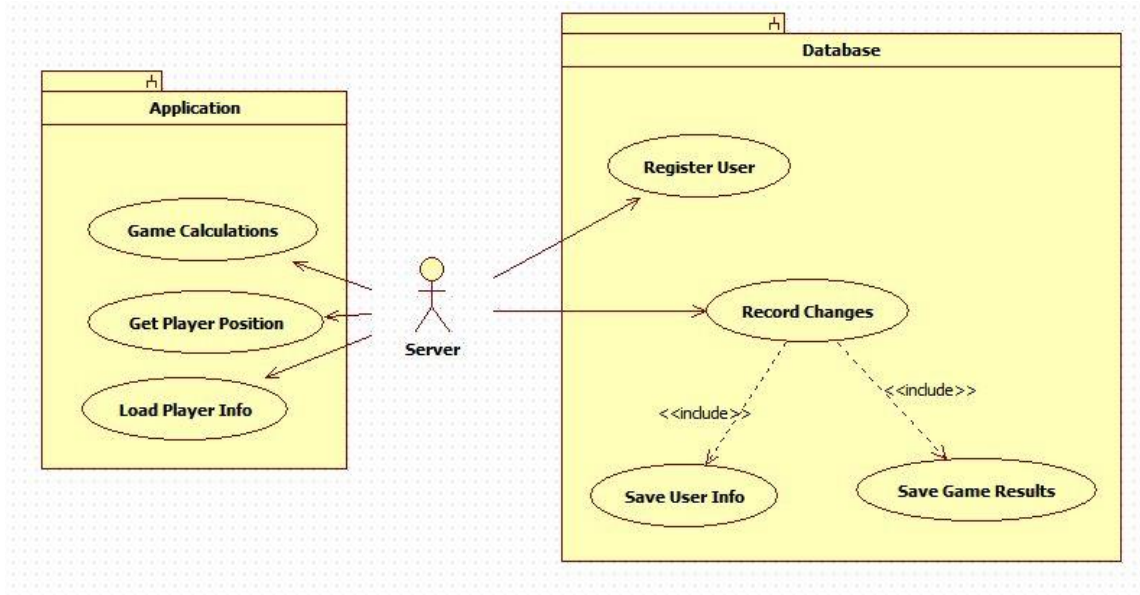


Figure 4 – Use Case Diagram for Server

| Actor | Use Case | Description |
|--------|---------------------|---|
| Server | Register User | Once a User performs Sign Up operation, the Server checks the entered information and registers the new User to the user database |
| Server | Record Changes | The Server updates user and match databases according to results of games, and changes a User made on his/her profile. |
| Server | Load Player Info | The Server loads Player's information from the database to be displayed on the screen. |
| Server | Get Player Position | The Server reads Player's positions from the ports, to use it in calculations and transfer it to other clients. |
| Server | Game Calculations | The Server calculates Players' and characters' experience points and levels. |

Table 3 – Use Case Descriptions for Server

2.3 User Characteristics

Our target group is required to be familiar with network based multiplayer games. Our game is a game of violent. That is the reason why our player should be aware that this is only game. Thus our players are required to be range of age limits to not be affected by our game's violent content.

2.4 Constraints

There is a single item that will limit the developer's options are: To play game on cross platform with Kinect, to be used Kinect SDK must be open source or derived from open source Kinect SDK which can provide use of Kinect on cross platform.

2.5 Assumptions and Dependencies

We assume that the users of REVENGE know how to set up and use their Kinects. Since REVENGE will be compatible with all operating systems, Kinect should be compatible to all of them too. In order to achieve this, we need an open source library which can work on all the computers. These libraries are openNI and primeSenseNITE and these must be installed on users' computers.

3. Specific Requirements

This section contains all software requirements with sufficient details. The specified requirements are especially useful for designers to design the product and testers to test that the product satisfies those requirements.

3.1 Interface Requirements

The product has 11 main user interfaces, which can be listed as follows:

- Game website
- Welcome and login screen
- New player screen
- Main menu
- Game Settings
- Instructions
- User profile
- Character profile
- Training mode
- Match mode
- Tournament mode

Details related to each interface are described below.

3.1.1 Game Website

The game will be freely distributed through a website, like the most popular games of the same type. The website starts with a welcome screen, which is covered with attractive visuals and screenshots related to the game, in order to attract new visitors' attention at first sight. Colored, large buttons to start the game are available at the center of the welcome page. New versions, updates and extensions are also announced here as short titles for visitors who are already familiar with the game. Detailed news can be reached by clicking on the titles, and following the links. The welcome screen has buttons to forward the visitors to pages for downloading the game, displaying multimedia, viewing game information and reading news.

The "Download" page starts with a registration panel. Once a visitor is registered to the game, download links for different operating systems, together with their system requirements are displayed. The user selects an appropriate download option, and download process starts. This page provide services for User's operations with names "Sign Up" and "Download".

On the "Multimedia" page, photos and videos are exhibited to illustrate interesting and attractive features of the game.

On the "Game Information" page, prominent features, modes and basic rules of the game are explained. Explanations are supported by screenshots, videos and links, when necessary.

Game versions, updates and new features are announced on the welcome screen, as stated above. The newest 5 to 10 news are displayed on the welcome screen, and the news body can be displayed by clicking on the title. In addition, visitors can view all the news related to the game on the "News" page. Buttons for sharing the news on social media platforms are available for each news.

3.1.2 Welcome Screen

Once the game is downloaded and installed to user's computer, after user clicks the game icon, the welcome screen is displayed on a full screen window. The welcome screen is covered by images of game characters, and environmental objects, as well as the name of the game. There are three text boxes for user name, password and security code, and a "Sign in" button.

This screen is where the User performs "Log In" action.

3.1.3 New Player Screen

Just after a player's first signing in to the game, a new player screen is opened. This screen introduces the player the window layout, main buttons and short instructions by giving him/her simple tasks to complete. The player has the options of skipping this screen by clicking the "Skip" button. Otherwise, after completing a short task, the player clicks the "Next" button to display and perform the next task. Once the instructions are completed, the player is rewarded with some experience points, and forwarded to the main menu.

3.1.4 Main Menu

Once a user signs in with a correct user name and password, main game menu is displayed. The player's selected character, which is the character selected at the last session, is displayed at one side of the page. The menu has buttons to forward the user to screens "Settings", "Instructions", "User Profile", "Characters", "Training", "Fight Now" and "Tournament", whose details are described in following sections. A button to log out of the game is also available.

3.1.5 Game Settings

Under the settings screen, there are measures to set voice, effects, brightness, contrast and resolution of the game and an option to turn on and off the full screen display. In addition, the user can specify a game controller in this screen; in other words, the user can choose which hardware he/she will use to control the game character, keyboard, Kinect or joystick etc.

This screen is where the Player performs "Interface Settings" action.

3.1.6 Game Instructions

In this screen, game controls are displayed by default. There are six submenu buttons at the top of the screen to forward the player to screens which explain details about rules of different game modes, which are training, matching and tournament mode, and a screen which gives general information about how to play the game.

3.1.7 User Profile

In the user profile screen, user can display his/her experience point, match statistics, and ratings over other players. User can also edit password and e-mail information on this screen.

This screen is where the Player performs "Profile Settings" action, and when profile settings are changed, the Server performs "Save User Info" action.

3.1.8 Character Profile

On the character profile screen, the default character is the character which the player has chosen at the last game session. On this screen, specific properties of the selected game character is displayed, such as its name, power, level, experience point, durability level, and items that the user purchased for it. User can change the selected character by choosing another character on the selection pane. In addition, users are capable of owning new characters. If the user selects “<< new character >>” option in the selection pane, he/she can purchase new characters if his/her experience points and level allows.

This screen is where Player performs “Character Selection” and “New Character” actions.

3.1.9 Training Mode

In the training mode, there is the game character exercising with a punching bag in order to improve his/her abilities. Character properties are displayed at the top of the screen and improvements are visualized as the training goes on. An example environment for the training mode is visualized in Figure 5.

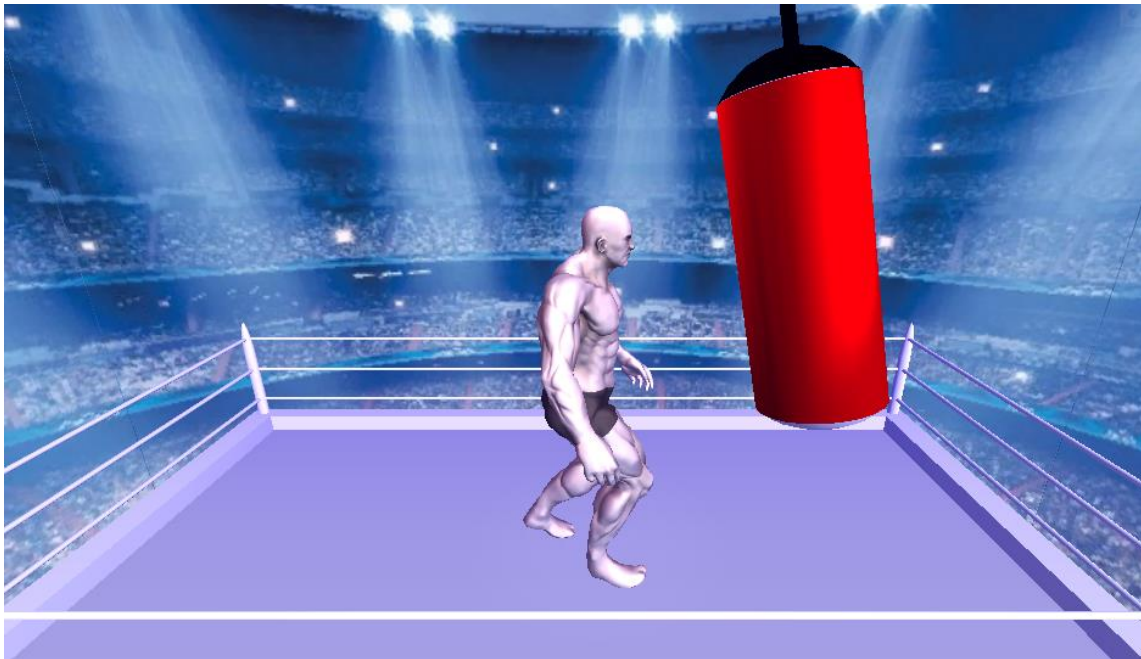


Figure 5 – Training Mode User Interface

3.1.10 Match Mode

In the match mode, there are two game characters fighting against each other. Character properties are displayed at the two sides of the screen and any improvements or losses are visualized as the match goes on. An example environment for the match mode is available in Figure 6.

3.1.11 Tournament Mode



Figure 6 – Match Mode User Interface

There will be periodic tournaments in the game. When a tournament is announced, desired players are registered to the tournament if their experience points and levels allow. When a player plays in the tournament mode, the match continues in a similar environment like the match mode. At the end of the match, player's rating after the result of the match is displayed. In the tournament mode, a player can display his/her results of previous matches and schedule of next matches in the tournament. Tournament statistics and leader board of previous tournaments can also be displayed in this mode.

3.2 Functional Requirements

This section describes major software functions, along with data flow and requirements of the function.

3.2.1 Functional Requirement 1

User can sign up to the system through the game website.

3.2.2 Functional Requirement 2

User uses the user name and password pair which he/she specifies during registration to log in to REVENGE. Whether the user name matches the password is checked on the server side.

3.2.3 Functional Requirement 3

If it is the User's first login to the game, the *New Player Screen* is opened, and the User is given a short tutorial on controls and basic rules in the game.

3.2.4 Functional Requirement 4

If it is not the User's first login, the User is forwarded to *Main Menu*.

3.2.5 Functional Requirement 5

If a Kinect sensor is detected on user's computer, the Game Controller automatically chooses Kinect controller as input source. Otherwise, keyboard controller is the default controller. User can change his/her choice of controller through the *Settings* menu.

3.2.6 Functional Requirement 6

Revenge has three game modes: Training mode, match mode and tournament mode.

In the *Training* mode, user exercises with a punching bag, in order to improve his/her characters and gain experience on game controls. The *Training* mode runs completely on the client side. At the end of each play, improvements are recorded by the server.

The *Match* mode should always be available to users at all hours of the day, and the game should run on the server side.

In *Tournament* mode, a series of matches in which winners get more points than regular matches are played. A tournament becomes available for users periodically, and the game runs on the server side.

3.2.7 Functional Requirement 7

After selection of the character, the user chooses a game mode. If the selected mode is not *Training* mode, the user is taken to game lounge where he/she waits for a competitor whose experience points and level are close to the user's, so that the match will be a fair play.

3.2.8 Functional Requirement 8

Users should be matched in such a way that their game modes and controller selections are the same. For example, a user who will play in *Match* mode via Kinect should be matched to another user who will again play in *Match* mode via Kinect.

3.2.9 Functional Requirement 9

All information of users who play the match are held by the server during a match. Some required information is displayed on the game screen. All calculations related to characters and the game are performed on the server side, and updated values of remaining energy, remaining time, and character positions are transferred to clients.

3.2.10 Functional Requirement 10

Selected game character scores experience points according to the match result.

3.2.11 Functional Requirement 11

Experience points are used to level up a character or making in-game purchases.

3.2.12 Functional Requirement 12

Users can change appearance and physical properties of characters by in-game purchases. A purchased item is specific to the character for which it is purchased, and it cannot be used for another character.

3.2.13 Functional Requirement 13

Items purchased for characters are recorded on databases. Users can use a purchased item with no time limit.

3.2.14 Functional Requirement 14

Match statistics, including number of matches won, number of matches lost, experience points scored and lost, are recorded on databases for each user, they should never be deleted. Users are matched to each other considering those statistics.

3.2.15 Functional Requirement 15

Users should click to "Exit" button on *Main Menu* to exit from the game.

3.2.16 Functional Requirement 16

All updates, purchases and result of matches a user performs during the session are reflected to databases, and records are reloaded by server to the client side on the user's next login in such a way that the user cannot manipulate them.

3.3 Non-functional Requirements

3.3.1 Performance Requirements

- REVENGE should serve 100 users concurrently, this also mean that it must stand while there are concurrent 50 fights.
- REVENGE can serve 10000 users per day without any problem
- REVENGE should be available for at least 5000 fights per day.
- REVENGE should serve 5 million users per month.
- REVENGE users must have at least 3MB of internet connection speed.

3.3.2 Design Constraints

REVENGE should be developed by using IEEE standards. The code should be understandable and some standards should be applied to

- Class should obey PascalCase standards.
- Fields of classes should obey camelCase standards.
- Class methods should obey PascalCase standards.
- Local variables should obey camelCase standards.
- Interfaces should have "I" in the first letter of their names.
- Interfaces should obey PascalCase standards.
- All of the names of variables, functions, classes must be in English.

3.3.3 Software System Attributes

3.3.3.1 Reliability

- Users cannot change their characters specifications in their computers. These values will be stored in our database and will only be changed in our server.
- Users also cannot change their account information, since all the related information is kept in our database.
- In order to avoid cheating in fights, some algorithms will be applied.
- Users cannot send fake positions and cannot persuade us that these values are true.
- Mean time between failures (MTBF) should be at least 12 hours.

3.3.3.2 Availability

We assume that if REVENGE fails, the problem will be solved in 3 hours and the system will be back again. REVENGE will be available unless there is a crash or it is an update time.

3.3.3.3 Security

- REVENGE will require an email activation to complete the registration.
- Instead of keeping the original password of users, we will keep their encrypted passwords.
- For those who forgot their passwords, a reset option will be provided.

- Cheating will be prevented with keeping all information (from users or game related) and doing all necessary calculations in server side.

3.3.3.4 Maintainability

- Requirement and change management should be used in development phase.
- Configuration items should be changed under version control system and linked to change request to provide traceability.
- Complexity should be minimized by using object oriented design, modules, interfaces.
- All design artifacts should be well documented.

3.3.3.5 Portability

- REVENGE will be downloadable from our website.
- REVENGE will be compatible to all platforms.

3.4 Logical Database Requirements

Below diagram is the entity relationship diagram of the database that Host DBMS will use.

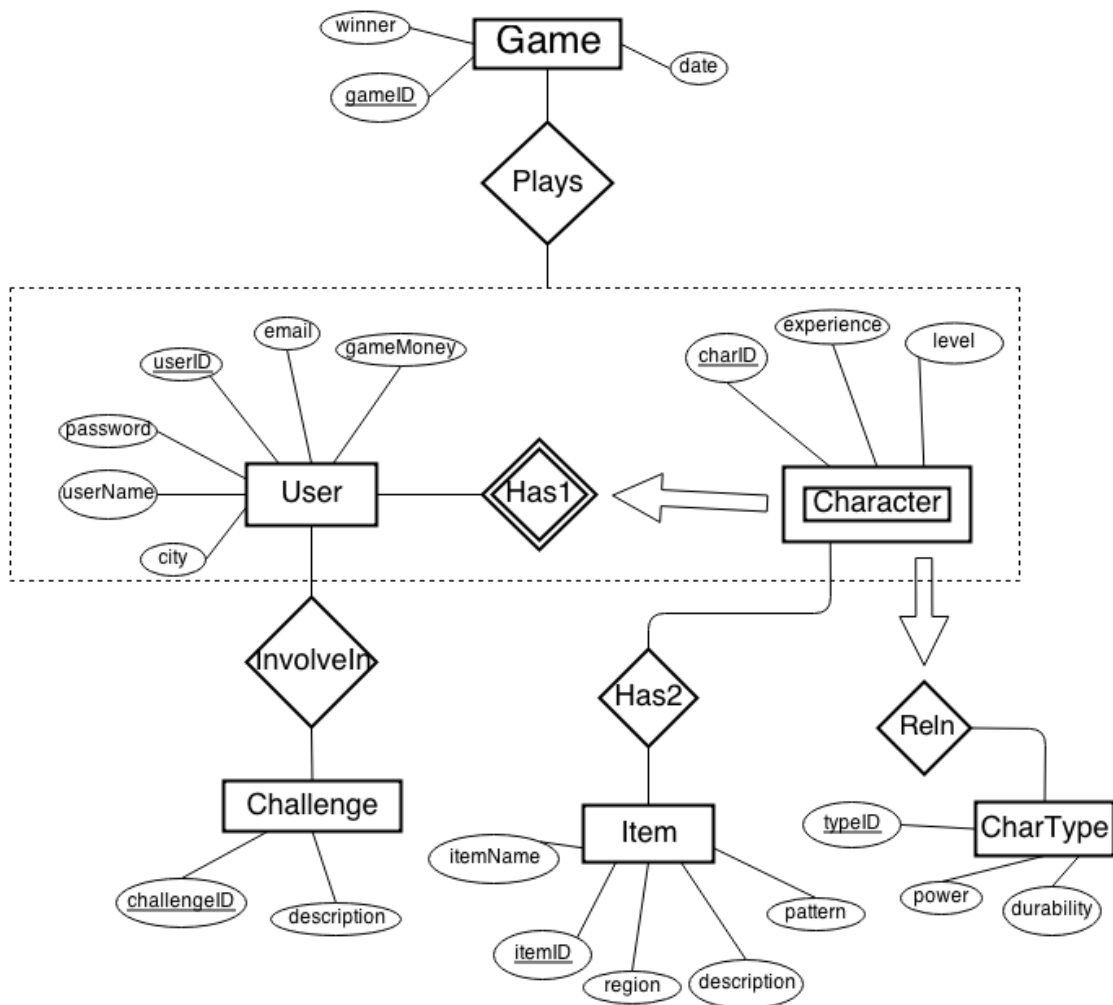


Figure 7 - Revenge Entity Relationship Diagram

4. Data Model and Description

This section includes information about data descriptions, their relations and diagrams related to the software.

4.1 Data Description

There will be different game modes such as training, league and quick fight modes. In the first mode-training mode- as much as a player practices, his/her character's power and durability increases. In the league, users will fight with another user and these games will be a base on determining the level and experience user's character. Level can be described as in its real meaning and experience can be described as some points which must be gained for the player in order to be promoted to the higher level. This is not a fixed number, it can change one level to another in order to maintain the challenge. In quick fight mode, a user also improves his/her characters' skills like power and durability. Also in all modes, we may give some game money to the users.

This game will consist a store and there will be items for characters. An item will have a price which can be game money or real money depending on the quality of that item.

Finally there will be challenges so that users can participate and gain some money, experience points etc.

4.1.1 Data objects

As mentioned in the previous section the following are the definitions of the objects.

User: Each user can have at most 2 characters and from this button they can see the characters' specifications which are their levels, items, power, durability, intelligence, history like participated fights etc.

Character: This is a game player which is special to its user and includes user's level and experience etc.

Character Type: These types forms the characters and gives them some "initial" features such as power and durability. These can be improved by the user later.

Game: Game will include 2 or more player and their positions

Position: The player's current place in the environment. These values will be used for detecting collisions and dynamic environment of the game.

Item: Items are wearable object in game and can be bought with game money or cash and increases the power or durability of the character.

Challenges: A user can participate in several challenges. These challenges can be based on different game modes and different time intervals. For example a 2 hours of practice challenge is valid for training mode and 1 day interval.

A short explanation is written on each bidirectional association in the Figure 8.

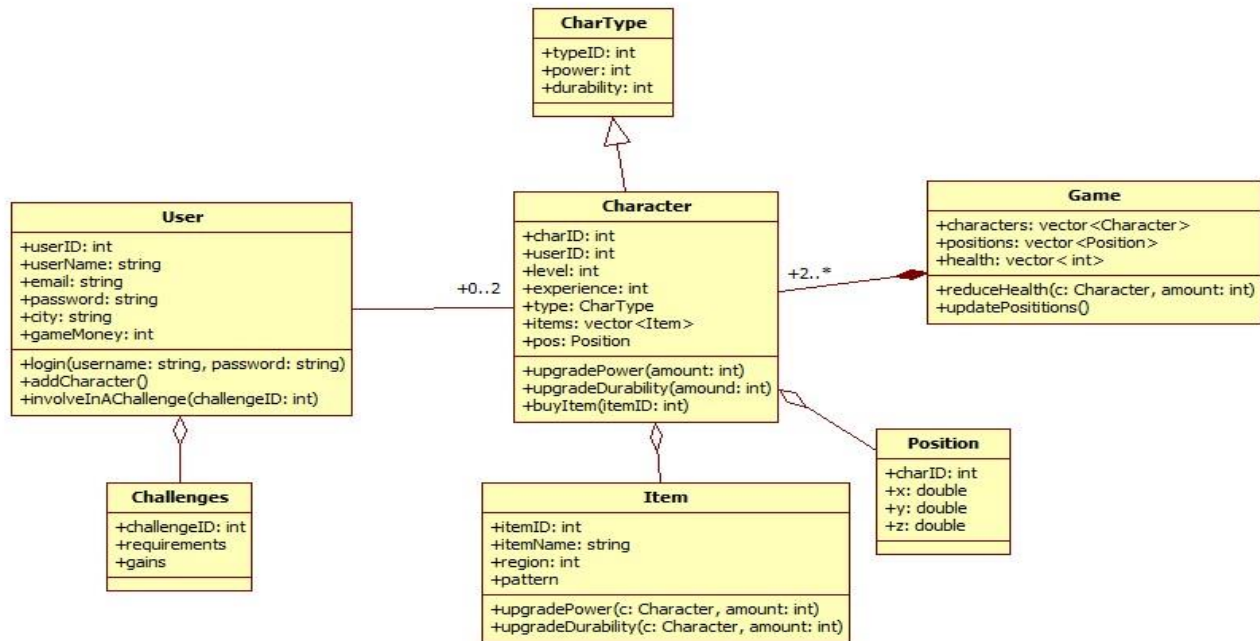


Figure 8 - Revenge Class Diagram

4.1.2 Data dictionary

Related definitions can be found in section 4.1.1

5. Behavioral Model and Description

This section presents a description of the behavior of the software.

5.1 Description for software behavior

Our game will consists of several game scenes. They are;

Mode Selection: In this scene the user chooses the playing mode which he/she wants to play or he/she can choose to do shopping or involve in challenges.

Store: Users can see details of their items here and they can buy new items but these items can be bought by game money, which can be gained in fights, or cash (of course these items will be much more featured). Also the items have their own region to be fit.

Challenges: Described in section 4.1.1

Training: When player choses training game mode, player will fight against punch bag. This mode will make it possible to upgrade player's physical abilities (as power, speed, agility etc.). Player will be able to play at training mode whenever he/she wants but there will be a limitation to improve power of player. Player can upgrade his/her character maximum 5 times of a day.

Match (Quick Fight): When player choses this option, the game will assign random player to the player as a rival. The result of a match will increase the experience points of both players according to player's level and winning side. This change at experience depends on the levels of players. What we mean is that, if the player wins the match against another player that has higher level, the experience points of lower level player will increase dramatically.

Tournament (League): First 3 Mondays of every month will be weekly championships. If players want to enter these leagues, they have to make an appointment until 00.00 Sundays. This mode will be a kind of tournament mode which differs at 2 points from regular tournament;

5.2 State Transition Diagrams

This diagram shows the overall behavior of the system.

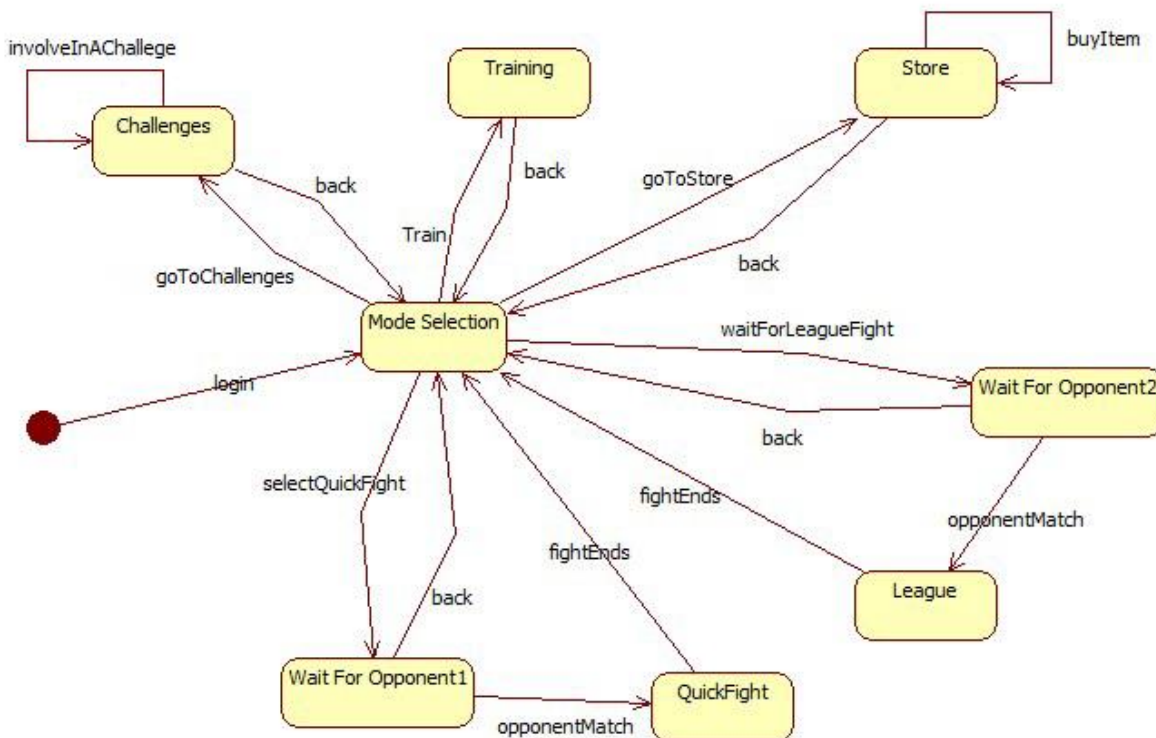


Figure 9 – REVENGE State Chart Diagram

6. Planning

6.1. Team Structure

Our team consists of 4 people who are; Onur Can ÜNER (team leader), Gürkan KARAHAN, Kader BELLİ and Serkan TUNA. All of our members are last year students of METU and our team was created 1 year ago. We are together because all of the members have different styles and characters and this make it possible to look and analyze any kind of problems or ideas from different aspects. Another advantage of our team is every team member likes to work on different subjects and assigning some task to somebody is not a problem.

We are meeting twice a week as a team and in addition to that once a week with our assistant to analyze the progress of our project. We are trying to assign the tasks equally to make contribution of every member in a team equal. Currently Kader and Serkan are working on a graphical part and Onur Can and Gürkan are working

on a Network part of a project. At meetings we are communicating and sharing the knowledge that is earned. It makes it possible to develop our project in parallel.

We are always deciding something together. Every member of a team listens each other so the decisions will becoming more stable and true. We defined code standards when implementing some functions and we are defined the informal report standard to give each other the speed up the implementation process. Everything is going well and we hope to be successful and reach our goals as a team.

6.2. Estimation

The basic tentative plan that will be followed for next 4 week until the demo day of our project is listed below as a table;

| MAIN TASK | ASSIGNED TO | ESTIMATED START DATE | ESTIMATED FINISH DATE |
|---|---|-----------------------------|------------------------------|
| Writing game server | Onur Can ÜNER | 01.12.2014 | 12.12.2014 |
| Graphical issues and improving the reality of training environment | Kader BELLİ | 25.11.2014 | 07.12.2014 |
| Adding required animations to the models and designing and improving the reality of match environment | Serkan TUNA | 25.11.2014 | 07.12.2014 |
| Binding Zigfu SDK to designed environments and defining a data type that will be sent to the server | Gürkan KARAHAN | 01.12.2014 | 12.12.2014 |
| Designing the architecture that includes classes, structures, pattern, of the project | All team members | 18.11.2014 | 15.12.2014 |
| Implementation of code for first prototype | All team members (modules for each member) | 12.12.2014 | 25.12.2014 |
| Designing user interface | All team members (Kader BELLİ) | 16.12.2014 | 25.12.2014 |

As we introduced at proposal form, there will be training mode on our first prototype. We also want to present multiplayer mode (without network optimizations) until the first prototype additionally.

6.3. Process Model

Scrum which is iterative and incremental technique of agile software development, is followed by our team. Up to now we are happy to using this process model because it makes easier to develop our project thanks to its concept is iterative and incremental.

7. Conclusion

There exists Kinect technology which is already used in X-BOX platform. The main reason that we choose this project was there is not a game that uses Kinect technology currently on PC. And there's not such a game which provides multiplayer Kinect gaming with network connection.

As we referred at introduction section the aim that we are trying to reach is changing people's looking aspect to the computer games and making them realize that sometimes it can be (playing our game) good activity for both physically and mentally.