+18 SOFTWARE

Presents proudly

"HITME"

Middle East Technical University

Computer Engineering Department

Design Project

Initial Design Report

16/12/2008

CONTENT

CONTENT OF FIGURES

1 INTRODUCTION

1.1 Project Definition and Goals

The project is a kind of laser tag game which is played with a gun in any arranged room. Main goal of the game is to hit opponent members in the game area. Players can play this game by creating their own teams or individually. All players have their own guns which will be a designed circuit which has infrared led as firing component. At the gun there will be a lens which is located in front of the infrared LED. This lens is used to collect infra red light beams into a bunch of linear infrared light beam. Infrared light can be captured from far distances using lens. This infrared light which comes from players' guns will be captured by a sensor placed on the players' bodies.

Also there will be health indicators on the players' bodies. In our design there are 10 LEDs used in Health indicators. These LEDs are composed of 3 different colors which are red, yellow and green to show low, medium and high health respectively. The PICs on the players decide whether the LEDs turn on or off, when the health of player decreases or increases.

Some bonus devices are located throughout the game area to make this game funnier. Players can shoot these bonuses to increase their health or number of bullets on their guns. Another special bonus which locks the opponent members' guns within some seconds is decided to increase the importance of bonuses.

A graphical user interface will be designed to control the game at the server side. User who manages the game can initialize game settings and modes with this GUI. Health, scores and teams of the players can be seen in this GUI at the run time after the game starts. GUI gets these information from players and bonuses with an RF devices located onto the players, bonuses and server. There is an option to look at the high scores of the players. To decide high scores, this program saves all game data when the game is ended. User can stop the game at any time during the play and then restart the game with the same properties of players and bonuses. With all these features of GUI, user can manage the game easily.

1.2 Purpose of Document

The purpose of this report is to show our initial design concepts about project. In this report, we gave details of the project according to requirements explained in the requirement analysis report. This report implies initial design of the following parts of the project.

- software components (server module, PIC module and bonus module)
- hardware components (RF device, PIC)
- complementary components (optics module, IR receiver module)

2 DESIGN CONSTRAINTS

2.1 Resource Constraints

We need datasheets of the devices that we will be using, and manual of the software development environment that we will use for coding. These documents will be supplied by our teaching assistant and also we will use internet resources whenever we need extra information. Sometimes it may difficult to find the resources that are really useful for us, and this might limit our development.

2.2 Power Constraints

Since we are designing real platform game and the players will be carrying their own guns and sensors on their jackets, the power resource must be light enough not to disturb the player. The power supply must also be strong enough to feed the components.

2.3 Time Constraints

The deadline of our project is June and also we should provide a prototype at the end of this semester. Therefore, especially for an embedded project, this is the most important constraint. To use our time efficiently we must follow our schedule strictly.

2.4 Ergonomic Constraints

Since we will be using a device (ZigBee) which is new for us, we may have some problems with integrating it with the other devices we will be using.

2.5 Manufacturing Constraints

We will design the guns and the jackets for players on our own, and manufacturing these will be hard for us. To be more specific, designing and producing the gun with our circuit in it and also the jacket with sensors is a big deal.

2.6 Performance Constraints

As the number of players increases, there may be a huge traffic of interrupts that must be responded correctly. In such cases some of the interrupts may be lost if coding is not good enough. To avoid that kind of scenarios, our coding must be so efficient and well designed.

2.7 Experience of Members

Lack of experience of the team members on coding for an embedded device is one of the restrictions. Sometimes we may have difficulties with managing unexpected problems and unforeseen details of the project.

3 GAME CONTENT

As described in the project definition part, the subject of this game is to "hit" other players.

The main game elements are players and bonuses(and as coordinator "Server").

Players will have special electronical equipments in the game. An infrared gun to shoot, few infrared sensors to get shot by other player and health indicator LEDs for displaying health info. By these equipments , the player can shoot another player or a bonus(decribed below), or can get shot and see his/her health status by the help of health indicator LEDs. Also the bullets for the guns are limited , and if player is out of bullets, he is out of the game.

Bonuses are electronical devices. They can get shot by a players gun , giving him/her the specified bonus. They have activation times , whenever that time comes, the bonus will be activated and indicator LEDs will be displayed informing the type of the bonus. These bonuses will be deactivated if a user shoots or the time interval ends.The types of these bonuses are : health bonus, bullet bonus , lock opponent's guns bonus.

All these shooting stuff and other processes will be coordinated by a server. The communication between the server , players and the bonuses will be all wireless, the player will not be aware of this process.

This game is score based. When a player shoots another one , points will be added to his/her field.

The game can be team based or individual. If teams are defined , player will try to hit opponent teams' players. If not , shoting another player will be enough for getting points.

There are two game modes defined. One is score limited. In score limited game, a score is defined in the beginning of the game and if a player (or a team, depending on team based or individual mode) reaches to that score, game will end. The other mode is time limited, if the defined time is up , the game will end and the scores will be calculated.

4 SYSTEM ARCHITECTURE

4.1 Overview

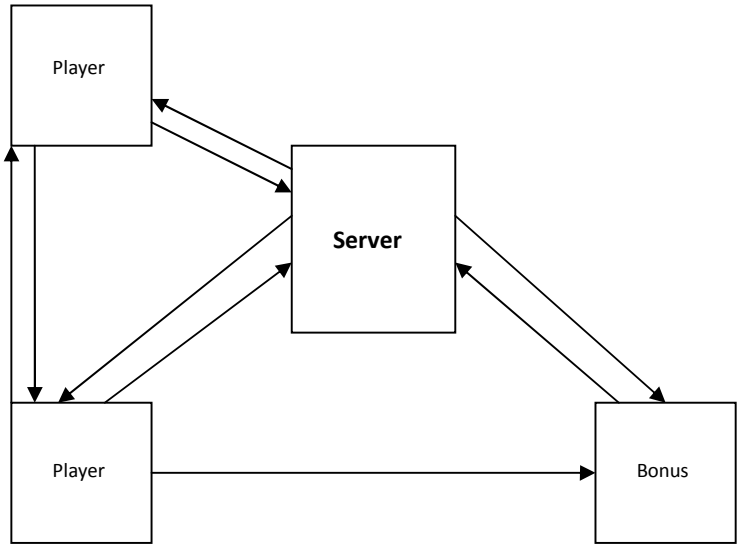

Figure 1: General Overview of the system

Our system consists of 3 main elements which are server, bonus and player. The module based structure of these elements is given below.

- Server

This consists of bidirectional RF communication module (server to bonus, player and player, bonus to server) and a PC.

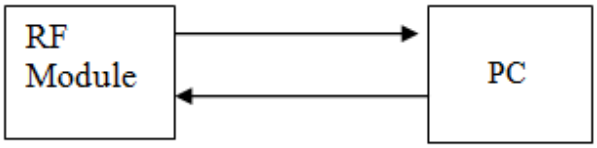

Figure 2: Server Device

- Player

This consists of bidirectional RF communication module (server to player and player to server), bidirectional IR communication module (player to player) and a microcontroller module.

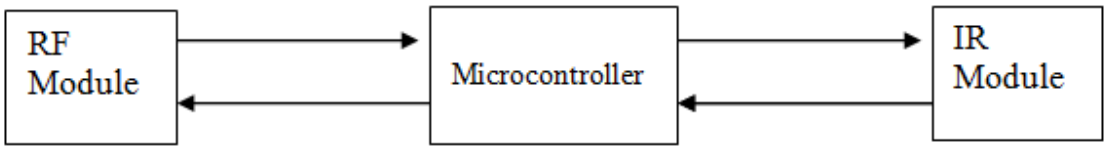

Figure 3: Player Device

- Bonus

This consists of bidirectional RF communication module (server to bonus and bonus to server), unidirectional IR communication module (only player to bonus) and a microcontroller module.

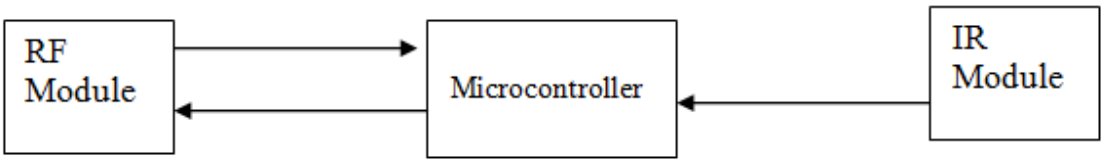

Figure 4: Bonus Device

4.2 Architectural Design

To develop our model, we have decomposed the system into three main modules, which are hardware module, software modules and complementary module.

4.2.1 System Hardware Modules

An overview of the hardware system is as seen below.

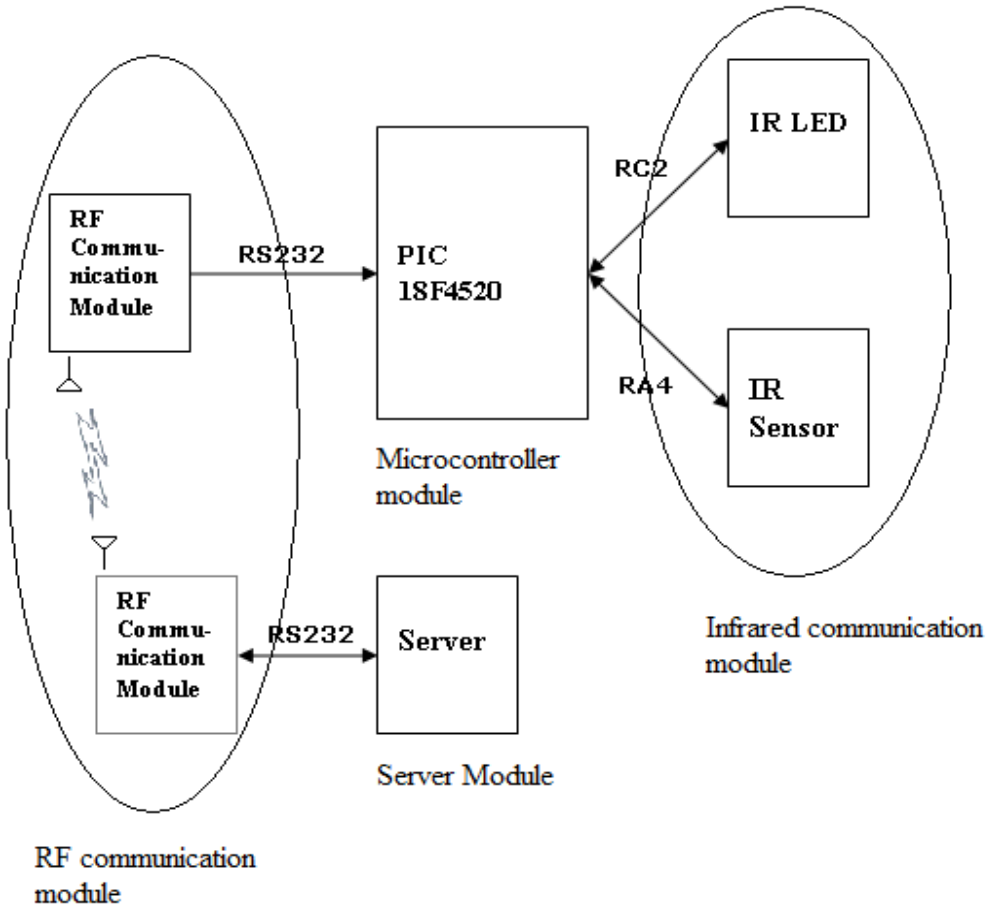

Figure 5: Overview of the Hardware System

4.2.1.1 Microcontroller Module

The main part of the hardware system is the microcontroller module. As microcontroller, we will use PIC18F4520. The main components of this module are

1- Microcontroller
2- Health Indicator LEDs via digital outputs(Bonus indicator in bonuses)
3- RF communication via RS232
4- IR LED via RC2(not in bonuses)

5- IR Sensor via RA4

6- Pushbutton via RB0(not in bonuses)

7- Sound Indicator via digital output(not in bonuses)

The overview of the microcontroller module is given in the figure above, “Module view of the system”.

The RS232 communication will be used for RF module interaction (will be described in detail later).

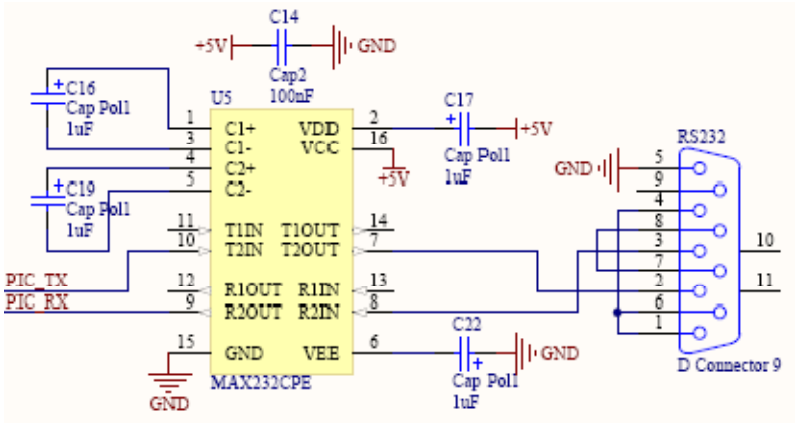

Figure 6: RS232 Communication of PIC184520 [1]

The infrared LED and sensor connection is made via RC2 and RA4. By RA4 connection of the sensor, the system will be able to use the TMR0 interrupt for infrared data receive. IR data transmit part will be all digital. IR communication devices will be described in “Infrared Communication Module”

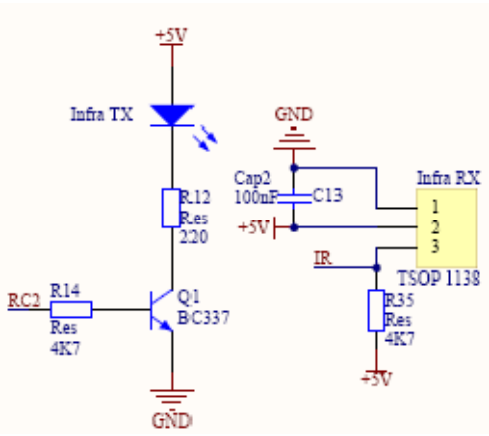

Figure 7: Infrared TX/RX of PIC184520 [2]

The Health Indicator LEDs will be used for informing the user about health information. The above 3 LEDs will be green, middle 4 LEDs will be yellow and bottom 3 will be red. Also, in bonus elements, these LEDs will be used for informing the user about the type of the bonus. This hardware will improve the playability of the game. 10 of the unused digital ports are being used for this LED displays.

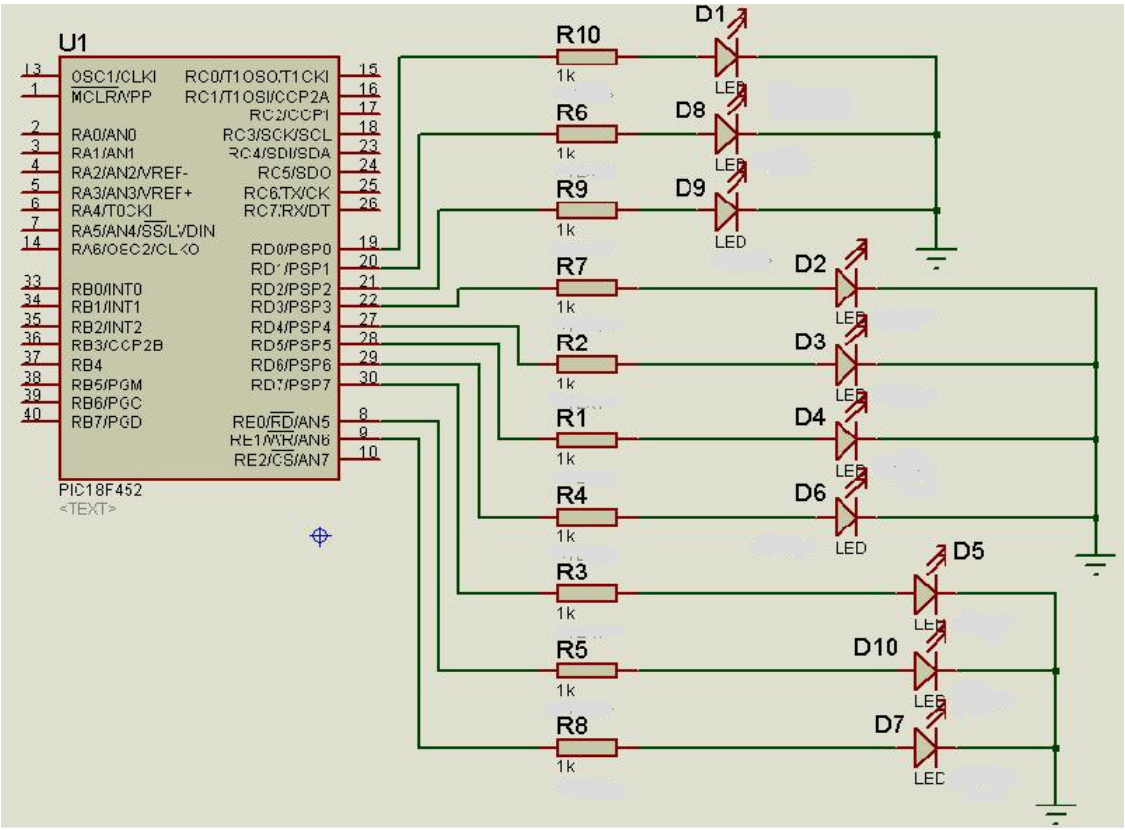

Figure 8: Initial Design of Health Indicator LEDs (image by Proteus designer)

One pushbutton will be added to the system for activating the IR LED. When the user presses to the button, the IR LED will start giving light at 38 kHz for a fixed amount of time (this will be arranged with the software).

A sound indicator is attached to this health indicator and will “beep” when health decreases below some critical limit. This is important, because the user should be informed

about her/his health condition .This beeper will be attached to one of the digital outputs of the PIC184520.

4.2.1.2 Infrared Communication Module

There are 2 main components in Infrared communication module.

1. TSOP1138 38 kHz receiver.
2. TSALxxxx IR LED, 5VDC, 850-950 nm wave length (this is not included in "Bonus" element.)

TSOP1138 IR receiver will be used in RA4 port in PIC184520. Together with the software, this module will be able to receive fixed-length data packages in 38 kHz frequency (which is produced by the other players' IR LED (namely "GUN")).

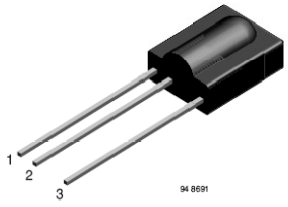

Figure 9: TSOP1138 [3]

This LED is connected to the PIC by RC2 port.

Again with the software, IR LED light will be sent to other players' receivers (38 kHz). The light will represent fixed-length data

Figure 10: TSAL1138 [4]

4.2.1.3 RF Module

As can be seen from the figure above, the communication between PIC18f4520 and Server is established by RF module. As RF communication device, we will use Xbee ZNet 2.5.

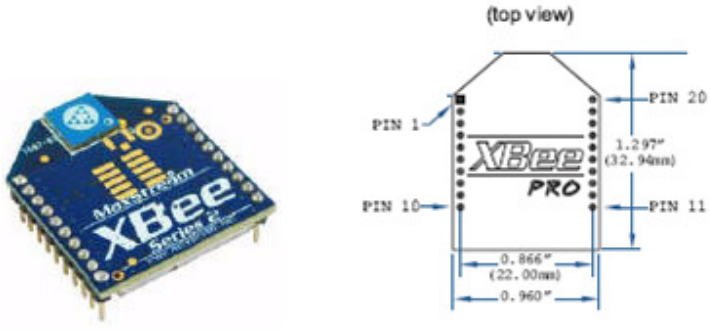

Figure 11: XBee PRO ZNet 2.5 Module [5]

This device can give the data to a host device through a logic-level asynchronous serial port. Through its serial port, the module can communicate with any logic and voltage compatible UART; or through a level translator to any serial device [6]. We are planning to make the data flow via serial communication ports. This would make the system easier to develop and debug. The communication logic of the device can be seen on the figure below

4.2.1.4 Server Module

The server module is the “Coordinator” of the system. Basically, the server consists of a single PC with a serial port and a RF module connected to it with that serial port. As described above, RF module will be an XBee pro 2.5. This PC will have the server side software running, and will send and receive the proper information.

4.2.2 System Software Modules

Software module will be composed of 3 main modules. These are Server Module, Player PIC Module, and bonus PIC module. Each module will be processing some operations by themselves. The modules will also have 3 main parts to process. 2 of the parts, which are sending data using RF module and getting data from the RF module, will provide the data transfer between the modules. The other part is processing the data as

internal operations. In the player and bonus PIC modules, there will be also IR processes which are sending data and receiving data via IR. The functions of the parts of each module are explained below in detail.

4.2.2.1 Server Module

Sending Data With RF Module	
ActivateTheGame ()	Activates the game at the beginning of the game. Activation means initializing the guns with the bullets and giving every player health degrees.
ActivateBonus (bonusType, bonus_ID, interval)	Activates the bonus with the given type (which can be health bonus, bullet bonus, etc.) and bonus_ID. Activation will be on up to the given interval.
GiveBonusToPlayer (playerID, bonus_type)	Sends the bonus of the given type to the player who got it(with the given player_ID).
DeactivateTheGuns ()	Turns off the guns when the game is over.
ReactivateTheGuns ()	Reactivates the guns with its previous values.

Getting Data via RF Module	
WhoShotWhom (what_shot_id, who_shoot_id)	Analyzes whether the player shoots a bonus or a player. Calls GiveScore(who_shoot_ID) and Decrease_Health(what_shot_ID) method if a player is shot. Calls LockTheGuns() method if the player shoots a bonus and the type of the bonus is locking guns of the other team's guns, and GiveBonusToPlayer(who_shoot_ID, bonus_type) method to give the bonus to the player.

Internal Methods	
DescreaseHealth (player_ID)	Decreases the health degree of the player with the given player_ID.
GiveScore (player_ID)	Give points to the player with the given player_ID to be assigned in the score table.
PauseGame()	Pauses the game with calling DeactivateGuns() method.
EndGame()	Calls SaveGameData() and UpdateHighScores() methods to finalize the game.
ContinueGame()	After some pause, calls ReactivateGuns() method to continue the game from the point it was paused.

4.2.2.2 Player PIC Module

Sending Data With RF Module	
WhoShotMe (player_ID1,player_ID2)	Sends the ids of player that shoots(player_ID1) and player's own id(player_ID2). This method also calls DecreaseMyHealth() function to decrease health value.

Getting Data via RF Module	
GetRFDataFromServer (dataPackage)	Analyses the data package and calls the appropriate function. If the data is about bonuses , GetBonus(bonusType) function is called. If the data is about activation of the gun, it calls one of the functions; ActivateMyGun(), DeactivateMyGun(), ReactivateMyGun().

Getting Data via RF Module	
GetRFDataFromServer (dataPackage)	Analyses the data package and calls the appropriate function. If the data is about bonuses , GetBonus(bonusType) function is called. If the data is about activation of the gun, it calls one of the functions; ActivateMyGun(), DeactivateMyGun(), ReactivateMyGun().

Internal Methods	
DescreaseMyHealth ()	Decreases the health degree of the player.
IncreaseMyHealth ()	Increases the health degree of the player.
DescreaseMyAmmo ()	Decreases the ammo of the player.
IncreaseMyAmmo ()	Increases the ammo of the player.
LockMyGun()	Locks the gun of the player for a predefined time period.
GetBonus(bonusType)	Calls the appropriate function according to bonusType which are IncreaseMyHealth(), IncreaseMyAmmo(), LockMyGun().
ActivateMyGun ()	Activates the player's gun when the game is started.
DeactivateMyGun ()	Turns off the player's gun.
ReactivateMyGun ()	Reactivates the player's gun with its previous values.
Getting Data Via IR	Data reception will be implemented with interrupts on RA4 pin. Using the interrupt, the data package will be caught and analyzed so that WhoShotMe() function is called with the ids of the players that shoots and is shot.
Sending Data Via IR	Data sending will be implemented with interrupts on RC2 pin. When the player shoots, the player's own id will be sent. Also DecreaseMyAmmo() function will be called.

4.2.2.3 Bonus PIC Module

Sending Data With RF Module	
WhoShotMe (player_ID,bonus_ID)	Sends the id of player that shoots(player_ID) and bonus_ID to server. This method also calls DeactivateMe() function.

Getting Data With RF Module	
ActivateMe(bonusType,interval)	Activates itself with the given bonus type , and also sets its activation interval.

Internal Methods	
DeactivateMe()	Deactivates itself when a player shoots it.
Getting Data Via IR	Data reception will be implemented with interrupts on RA4 pin. Using the interrupt, the data package will be caught and analyzed so that WhoShotMe() function is called with the ids of the players that shoots and is shot.

4.2.3 System Complementary Modules

In our project, some complementary materials will be used. Some of them are mandatory and some of them are optional for increasing the efficiency. Those are:

- Optics module for the IR Communication
- IR receiver module.

4.2.3.1 Optics Module for IR communication

The IR LED that is used in our IR communication has some limits. It cannot give the light as beams. So, some complementary material should be used to collect the light. We have decided to use convex lens.

The most proper convex lens and IR LED couple is found after some research. That is:

Infrared LED: TSAL 6100

LEDs beam angle: 20 Deg

Lens diameter: ~4.8 cm

Lens focal Length: ~10.8 cm

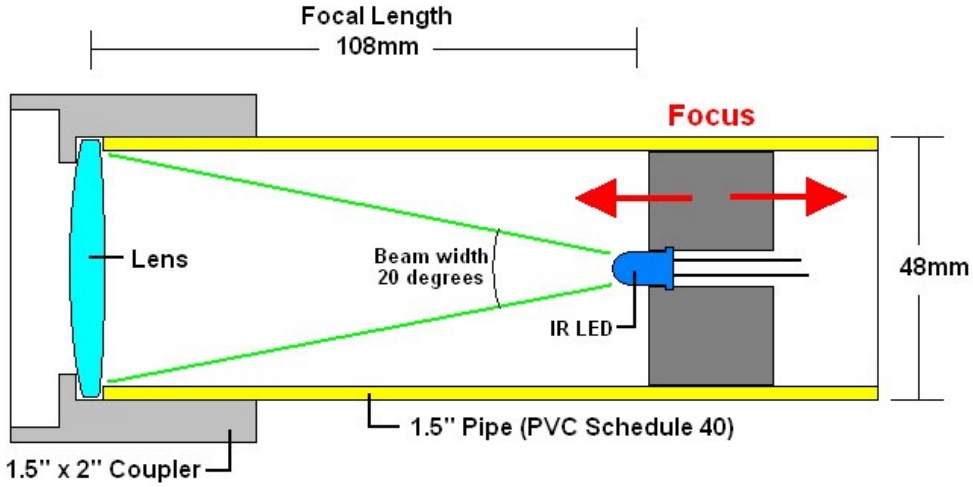

Figure 12: Usage of the convex Lens [7]

This lens will be in a gun-shaped material together with the IR LED. Also, in this gun-shaped material there will be a pushbutton, which will be used as a “trigger”. This gun will have a look like a real pistol.

4.2.3.2 IR Receiver Module

IR sensor will capture the light beams. But if only a single sensor is used directly, it will not be efficient. For efficiency, a material for the sensor should be found. Also, a system for the IR sensor will be mandatory for efficiency and reliability.

As IR sensor, TSOP1138 38kHz is used (which is described above). For the final version of the system, 4 or 5 sensors will be used.

The sensors will be connected serial. This will make the electrical design part easy to develop.

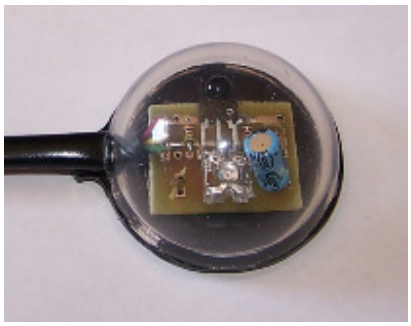

Figure 13: Representative photo of the sensor system [8]

5 MODELING

5.1 Functional Modeling

5.1.1 Data Flow Diagrams

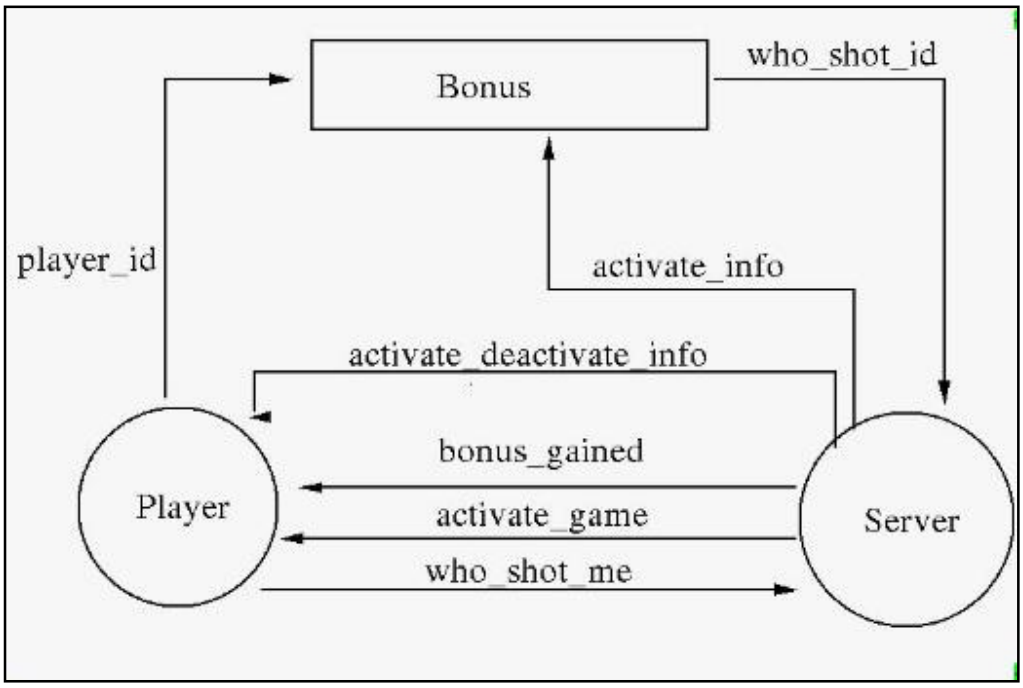

Figure 14: Level 0

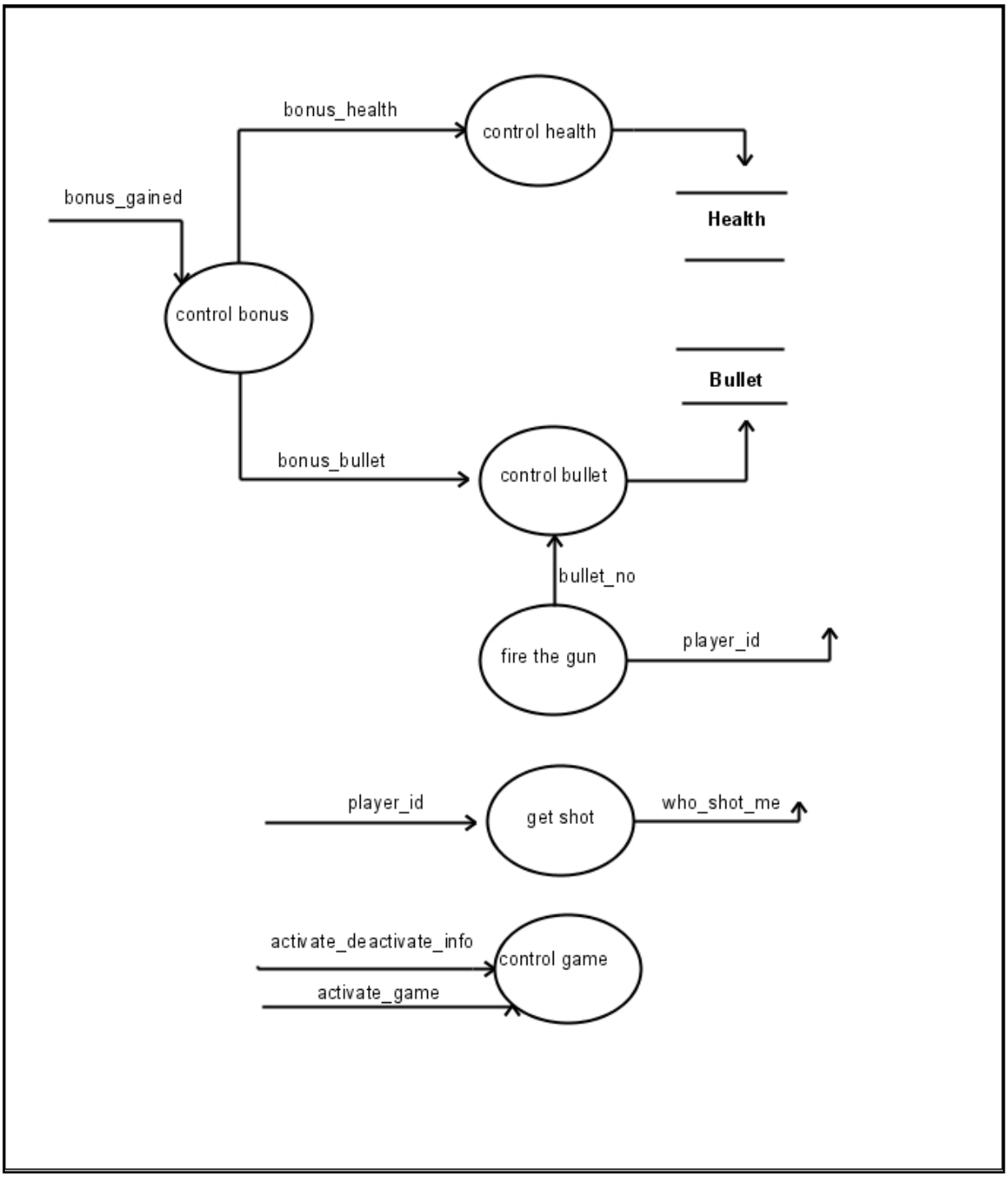

Figure 15: Level 1 Player

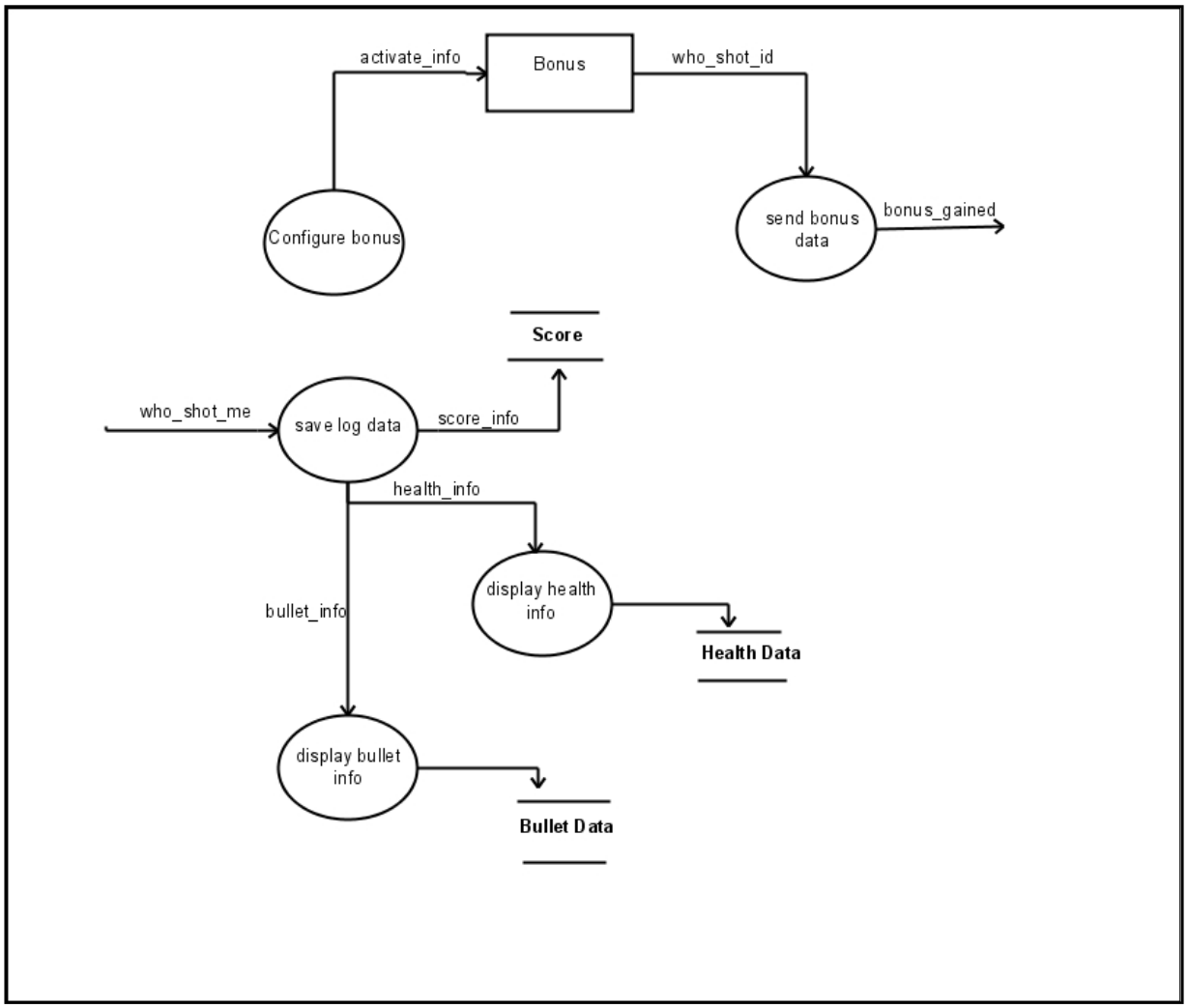

Figure 16: Level 1 PC

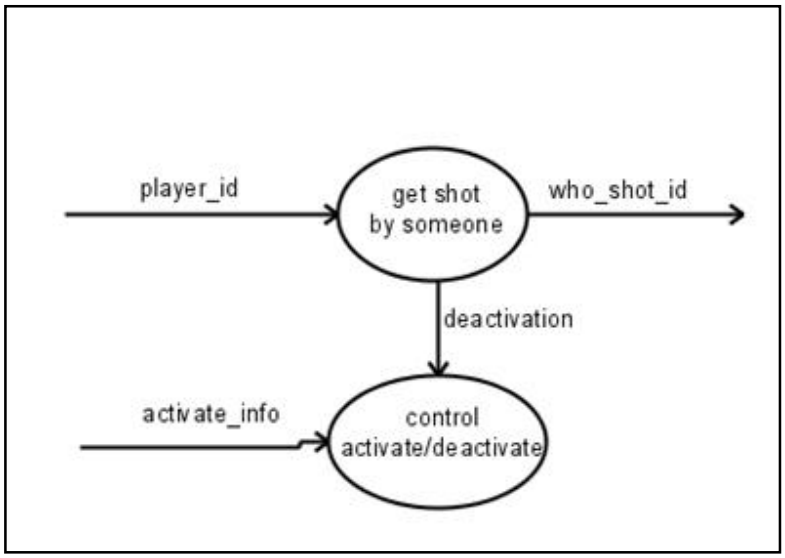

Figure 17: Level 1 Bonus

5.1.2 Data Dictionary

Name	Bonus pack
Where used?	Level 0 – External Entity
Description	The components which will be placed at somewhere in play area and activated by Server. It includes IR receiver and RF kit.

Name	activate_info
Where used?	Level 0 , Level 1-Server Output from 'configure bonus', Input to 'Bonus pack '
Description	Server sends some information using the RF kit to make the bonus package activated at some time of the game. Also the type of the bonus is in the activate_info.

Name	player_id
Where used?	Level 0 , Level 1-Player Output from 'fire gun', Input to 'get shot'
Description	Every time the player fires the gun, the gun is sending the id of the player.

Name	who_shot_id
Where used?	Level 0 , Level 1-Server Output from ‘Bonus pack’, Input to ‘send bonus data’
Description	When the bonus package is shot by some player, the IR receiver captures the player_id of the shooter, which is the who_shot_id. This id then goes to Server to understand who gained the bonus.

Name	bonus_gained
Where used?	Level 0 , Level 1-Player, Level 1-Server Output from ‘send bonus data’, Input to ‘control bonus’
Description	Server the bonus information (its type) to the player who gained it.

Name	who_shot_me
Where used?	Level 0 , Level 1-Player, Level 1-Server Output from ‘get shot’, Input to ‘save log data’
Description	This id will be sent by the player kit to the Server when the player is shot by someone to understand who shot the player.

Name	Health Data
Where used?	Level 1-Server Data Storage
Description	Health data of the players which will be stored in the server.

Name	Bullet Data
Where used?	Level 1-Server Data Storage
Description	Bullet data of the players which will be stored in the server.

Name	Score Data
Where used?	Level 1-Server , Data Storage
Description	Scores of the players which will be stored in the server.

Name	Health
Where used?	Level 1-Player , Data Storage
Description	Health degree of the player stored in player kit (in PIC).

Name	Bullet
Where used?	Level 1-Player , Data Storage
Description	Bullet number of the player stored in player kit (in PIC).

Name	bonus_health
Where used?	Level 1-Player Output from ‘control bonus’, Input to ‘control health’
Description	The info to give the bonus to the player when the player gained the health bonus.

Name	bonus_bullet
Where used?	Level 1-Player Output from ‘control bonus’, Input to ‘control bullet’
Description	The info to give the bonus to the player when the player gained the bullet bonus.

Name	bullet_no
Where used?	Level 1-Player Output from ‘fire gun’, Input to ‘control bullet’
Description	When the player fires the gun, the bullet number in PIC will be controlled by the player kit and stored bullet number will be recalculated.

Name	deactivation
Where used?	Level 1-Bonus Output from ‘get shot by someone’, Input to ‘control activate/deactivate’
Description	When bonus is shot by some player, it sends a deactivation message to control activate/deactivate in order to be deactivated.

5.2 Behavioral Modeling

5.2.1 State Diagrams

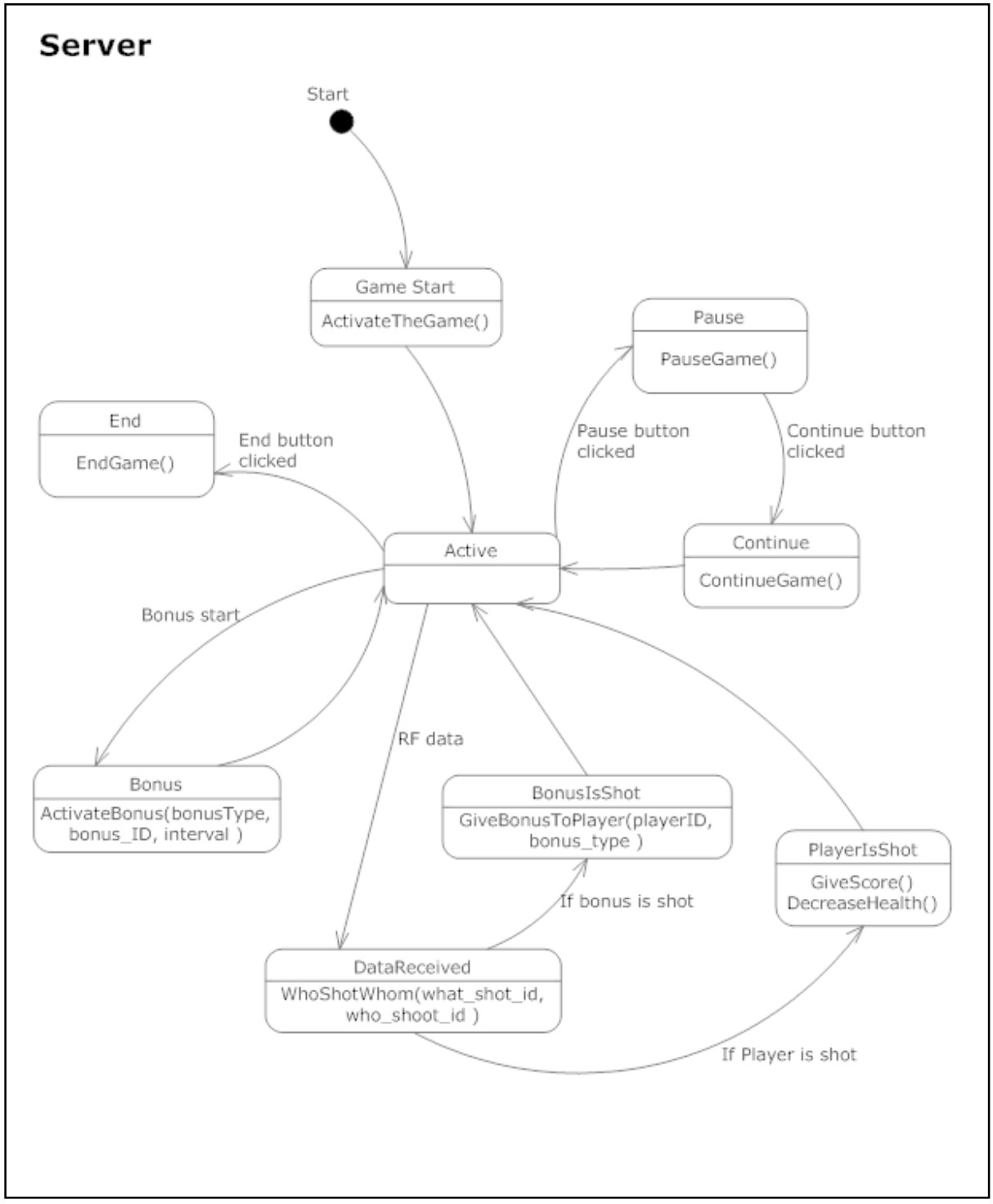

Figure 18: Server State Diagram

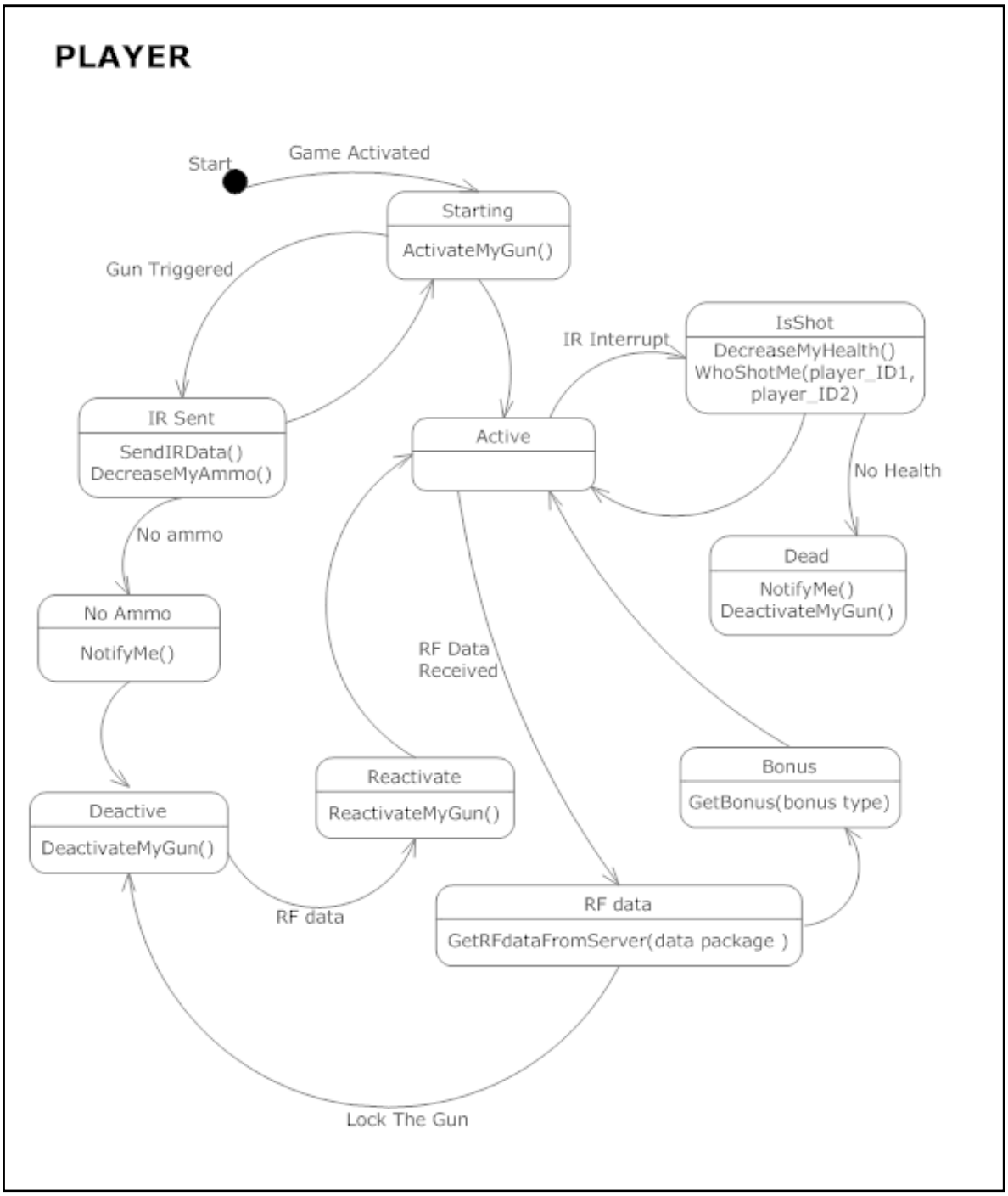

Figure 19: Player State Diagram

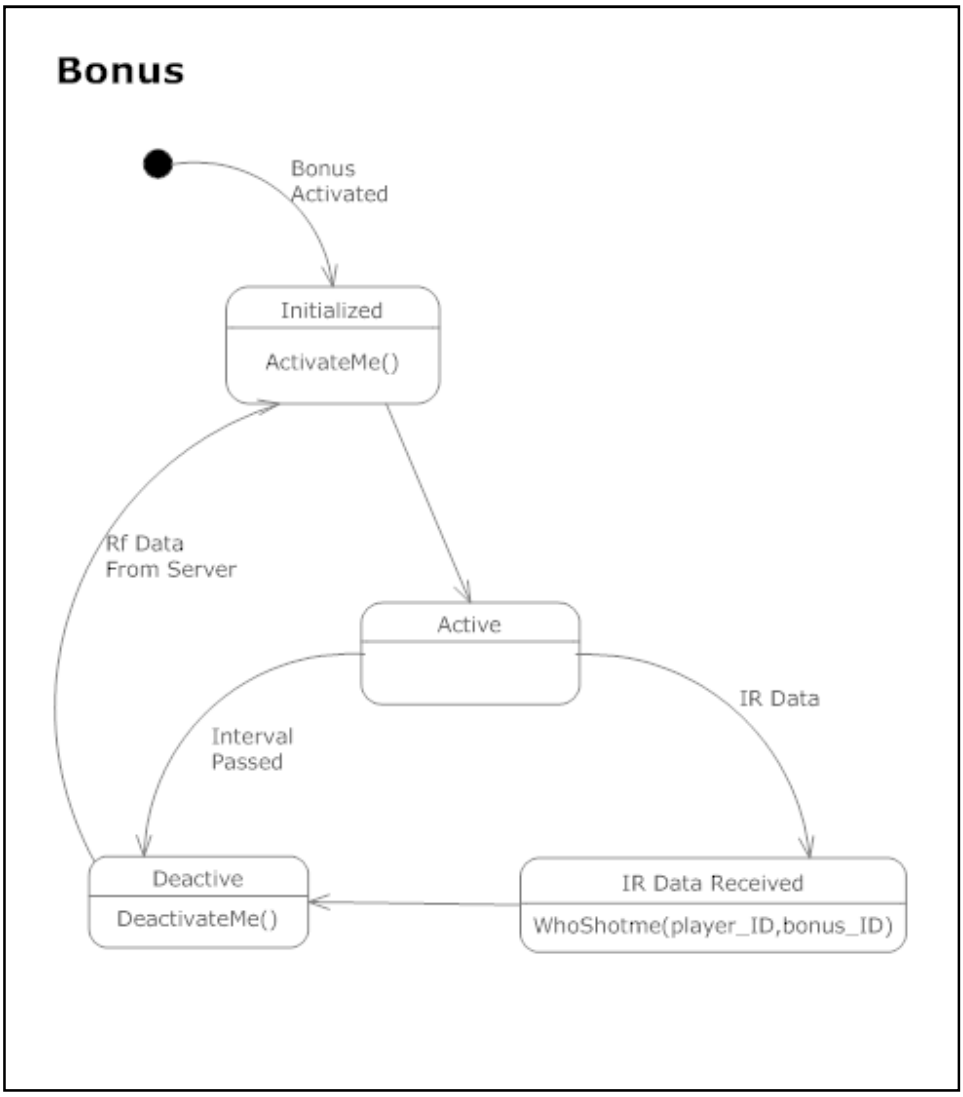

Figure 20: Bonus State Diagram

6 GUI DESIGN

We designed a GUI for initializing the game, choosing the mode of the game and deciding other options and specifications. In our design this GUI will be used by the person who is responsible for managing the game in front of the server.

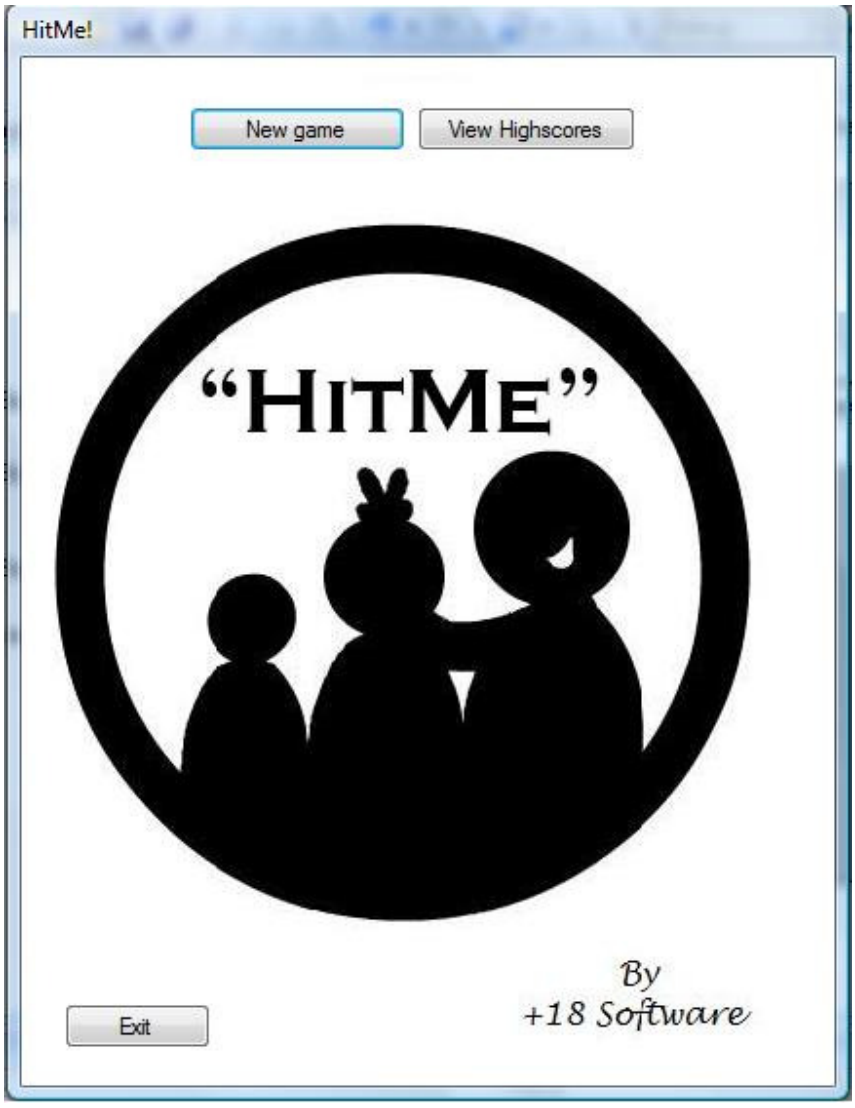

Figure 21: Hitme Main Window

The main screen of our game can be seen above. The user can see the high scores by clicking "View Highscores" button. The scores of the people will be sorted in our XML file.

This XML file will be updated after every game and high score table will be holding up to 100 players. When the user presses the “Exit” button the program terminates. Clicking “New game” button will direct the user to the “Game Mode Selection” window.

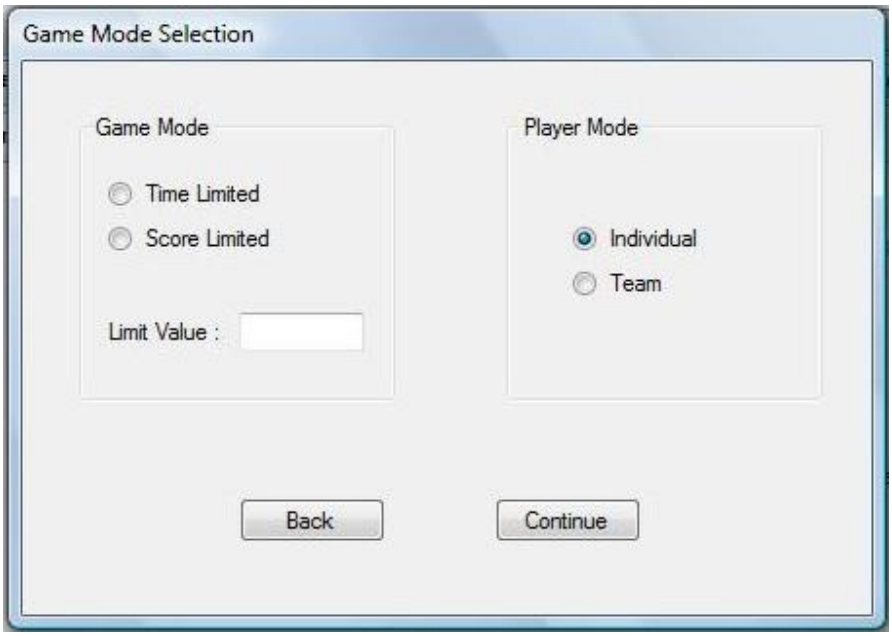

Figure 22: Game Mode Selection Window

In this window user decides the mode of the game which can be either “Time Limited” or “Score Limited”. In each case the user will enter the Limit Value. The game will be ended when this limit value is reached during the game. Besides user should select either “Team Mode” or “Individual Mode” .User should click “Continue” button after selecting the Mode.

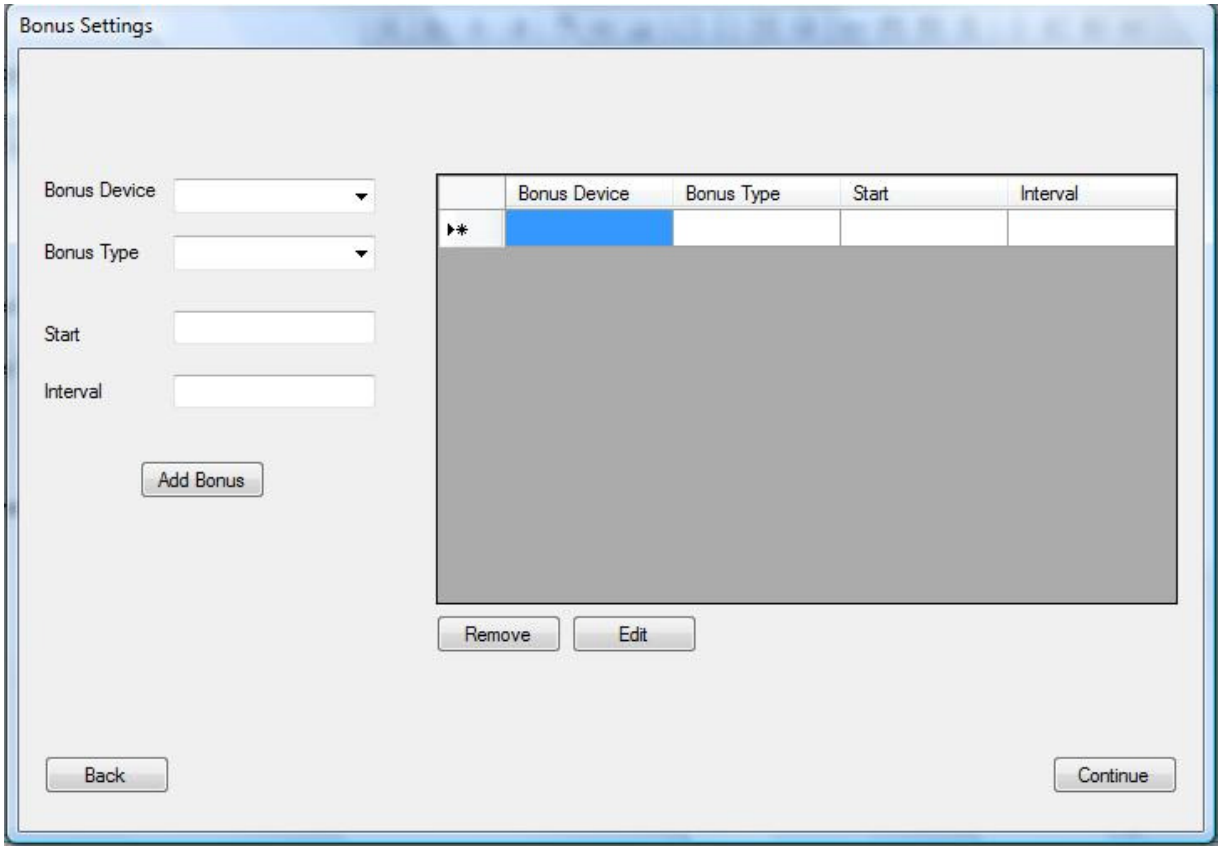

Figure 23: Bonus Settings Window

Before starting the game, the user must adjust the settings of the bonuses. In the game play area there will be some bonus devices located at different places. All of these bonus devices have a unique id and user can select one of these by “Bonus Device” combo box. User must select the type of the bonus of the selected device by “Bonus Type” combo box. These bonuses have two attributes which are start and interval. The start attribute decides the activation time of the bonus and this bonus will appear active within interval time which is typed by the user in “Interval” text label. It is possible to remove and edit the data entry selecting the row and then clicking one of these buttons. Clicking “Continue” button directs the user to the “Player Settings” window.

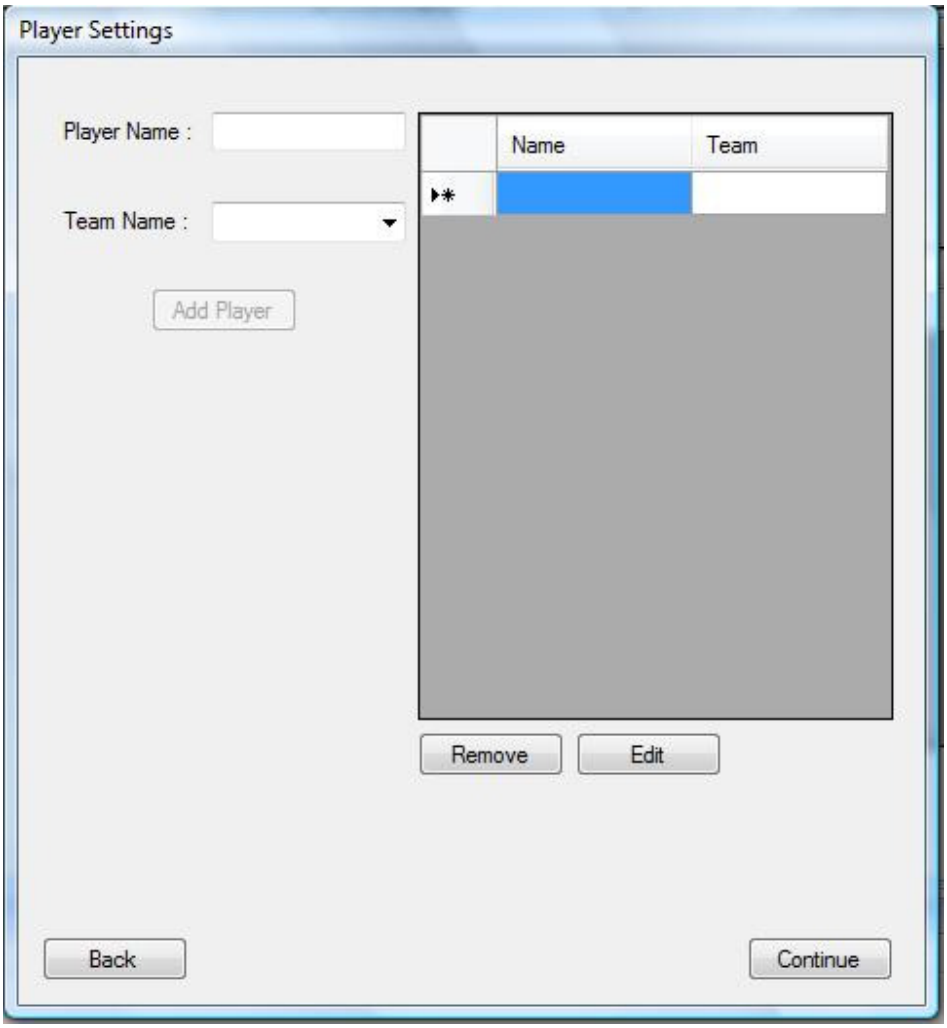

Figure 24: Player Settings Window

In this window user will create players by entering their names and selecting their teams. If Game mode is decided as “Individual mode” in the “Game Mode Selection” window, “Team Name” combo box will be locked and cannot be used by user. On the other hand, if “Team Mode” is selected, user should select a team name from predefined team names. After these steps, user should click “Add Player” button to add created player to the table. User can remove or edit data entries by selecting the related entry and clicking “Remove” or “Edit” buttons. Clicking “Continue” button directs the user to the “Now the Hitme Time!” window.

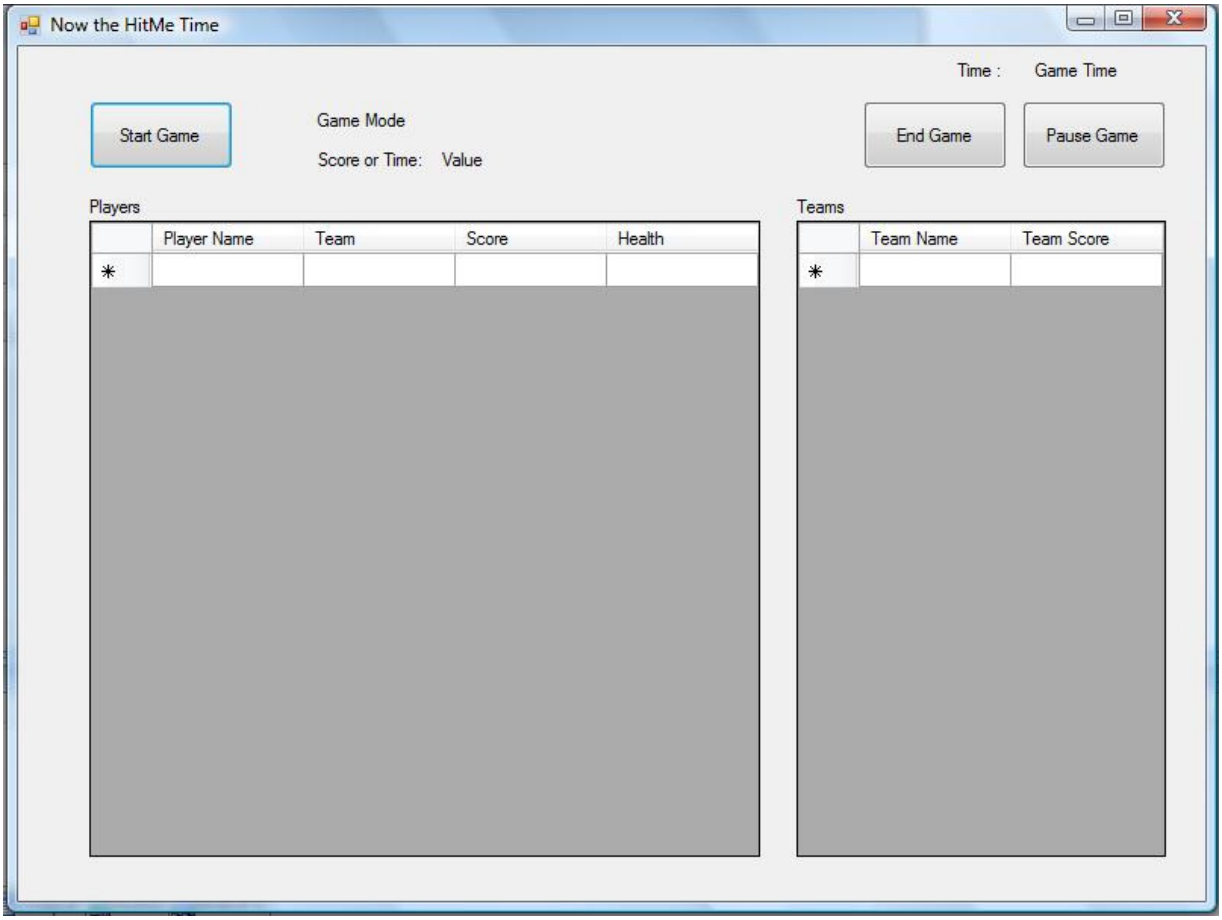

Figure 25: Hitme Game Play Screen

This window is displayed during the game. User can see player names, their team names, score values and health values in “Players” data table. If the game is in “Team Mode”, user can see team names and team scores in “Teams” data table. User can also see the limited value right side of the “Start Game” button. Time of the game can be seen at top right corner. When players are ready to play, user clicks the “Start Game” button to start the game. In any time user is able to end or pause the game by clicking “End Game” button or “Pause Game”. When the user clicks the “End Game” button, all of the game data will be saved to an XML file.

7 TESTING METHODOLOGY

Basically, our system has 4 main parts (figure: Module view of the system). Because of our design methodologies, testing will take place after each part has been developed. Every testable code segment will be tested just in time (we could not do the tests by test frameworks, they will be manually and semi-automated). In addition to that, some systematical tests for communication parts will be performed.

Those are:

- RF communication testing
- IR communication testing

7.1 RF Communication Testing

The RF module that we will use is an easy-to-use and practical device. But in addition to that, using it with a new designed microcontroller system has some risks. Together with untested software, it could be a big problem.

To avoid this, we are planning to perform two dimensional testing on this part. One dimension is the data length and the other dimension is the distance between receiver and transmitter.

Some test codes (functions) will be implemented for these operations. Arranging the arguments of these test functions will give us the ability of extreme testing.

Also, we will see that if our results agree with vendor's specifications or not. This is a good way of measuring the efficiency.

7.2 IR Communication Testing

IR communication is the most detailed and difficult part in our system. Some complementary material (lens, completed optics module.) will be used with the hardware. Testing of IR communication will take much time than developing it.

In addition to software testing, there will be physical tests. The efficiency of optics module will be tested by laboratory experiments. We will be able to make this test by the help of Department of Electrical and Electronics Engineering. The optics laboratory of that department will be in our use (by the help of Assistant Professor Behzat Şahin).

Other tests (software and hardware) will be two dimensional like RF tests. One dimension is data length and the other dimension is distance between transmitter and receiver. Test functions will be written and automated tests will be performed.

8 DEVELOPMENT SCHEDULE

8.1 Completed Parts

By the early days of this semester, we have started to make researches about this project. Due to time restrictions in this project, we have an early start to development. The completed parts of development process are described under 3 main titles:

- Microcontroller Hardware & Software Primitives
- Infrared Communication Primitives
- Complementary Material Design and Prototype

8.1.1 Microcontroller Hardware & Software Primitives

As a development board, we have chosen to use our department educational embedded system development kit. This has accelerated our process because of our experience in this kit.

PIC184520 is chosen to be the processor .The circuit design of our hardware system will be cut off from the embedded system development kit (The pins and circuit elements that will be used are chosen by the help of development kit.). Also, for “Health Indicator”, the hardware has been designed and ready for production.

8.1.2 Infrared Communication Primitives

Infrared communication hardware is ready (present on the development board). The usage of IR LED and sensor has been tested by the help of oscillator.

Simple test code is written for the test of data transfer via IR. This simple code has proven the success of the hardware.

8.1.3 Complementary Material Design and Prototype

The most ambiguous part in our project, optics module has been developed. Also we have made a basic prototype.

This prototype consists of three parts: a carton made tube, a convex lens and a carton made back cover. The tube has a length equal to the focal length of the convex lens. And the convex lens fits perfectly to the front of the tube.

We have placed an IR LED to the back of the tube (to the center of the cross section of the tube), and given constant voltage to IR LED. With the collective effect of the convex lens, the spread light of the IR LED has turned into a straight beam.

Completion of this prototype has been a milestone in our project. The straight beam would let the IR light to reach to receiver's sensor properly.

8.2 Future Work

At this point the first thing to do is implementing an IR protocol and RF protocol which is the milestone of this project. When we implement protocols, players will able to shoot each other and send this data packages to server module via RF device. After this step, we will develop our IR guns with the optics module to send data linearly. When we completed all of these, we will develop graphical user interface at the server module to process all of the data send by the bonuses and players.

If all the modules finish before the time we decide, we will design special guns and armors for players to make the game more attractive.

8.3 Gantt Chart

Figure 26: Gantt Chart

9 REFERENCES

1 . Department of Computer engineering , PIC development Kit user manual

2. Department of Computer engineering , PIC development Kit user manual

3. TSOP1138 ,Datasheet for IR Receiver Modules for Remote Control Systems, Vishay semiconductor Co.

4. TSALxxxx ,Datasheet for GaAs/GaAlAs IR Emitting , Vishay semiconductor Co.

5. http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-2.5-Datasheet.pdf

6. http://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-2.5-Manual.pdf

7. http://www.lasertagparts.com/mtoptics.htm

8. http://www.lasertagparts.com/mtsensors.htm