+18 SOFTWARE

Presents proudly

"HitMe"

Middle East Technical University

Computer Engineering Department

Design Project

Requirement Analysis Report

1 Contents

Table of Figures

1 Introduction

1.1 Project Title

Our project title is “HitMe”.

1.2 Problem Definition and Goal

The project is a kind of laser tag game which is played with a gun in a specially designed room. Main goal of the game is to kill other team’s members. In the gun, there will be a designed circuit which has infrared led as firing component. This infrared light which came from players’ guns will be captured by a sensor placed on the players’ bodies. Also there will be health indicators on the players’ bodies which are different colors of LEDs. The LEDs will turn on or off, when the health of player decreases or increases. Some specifications of the game are as follows:

-Decrease the health of the player that is shot.

-Lock the gun of the dead man.

-Decrease the bullets.

-Wait for a while when out of bullets.

-Place the bonus packs.

We chose this project because we think that creating a game will be lots of fun and at the end of the term we are planning to play it.

2 Team Organization

2.1 Team Structure

Our project team has “Democratic de-centralized” structure. We are planning to give the decisions about our project altogether. Each of us has nearly the same experience about tools and concepts. This made us to make the team structure like this.

2.2 Member Roles

Gamze, our contact person and timekeeper, has the capabilities of social discipline and obsessive character. Thus, she will motivate our team in a good way.

Onur will lead us in the dead ends that occurred in this process. He will tidy our works and make them presentable.

Çağrı is our “this will not work” person. As a result of this, he will create alternative ideas or force people to do it.

Zeynep, “Miss. smile”, she is calm and optimistic. She can also use these capabilities to record our process and discussions.

2.3 Process Model

In our project we will use waterfall design model (*classic life cycle*). There are several reasons to choose this model for our project.

First of all, our project is a mix of hardware design and software concepts. Since the project is mainly based on hardware design, once the hardware part is designed, it is nearly impossible to change. This forces us to identify our needs and specifications carefully at the

beginning. Because of this, we discussed the main issues with authorities and made a lot of researches about the possible hardware components that we can use in design. After all of these, we reached final analysis of our project which will not be changed at the end of analysis period.

The hardware design is unchangeable as we said before. The reason why we cannot change the hardware design is the difficulty of changing the main circuit design. After deciding on basic properties of the circuit and implementing it, we cannot make additions or eliminations on it. That makes our design part as strict as analysis.

Waterfall design model is an irreversible and sequential model which is the most appropriate model for us.

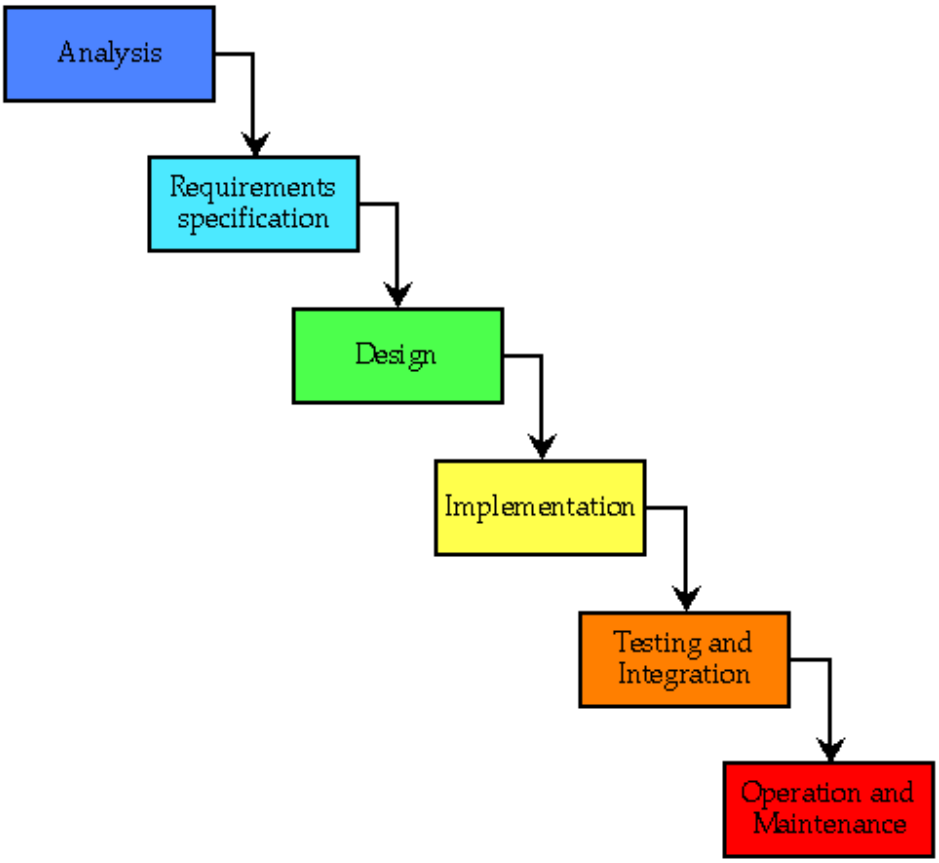

Figure 1: Waterfall

3 Research

3.1 Literature Survey

3.1.1 IR

Before literature research, we were planning to communicate via “laser pointer”. Using a camera and image processing was another idea to construct our idea.

Using “laser” will have only been possible if we used photo-resistor as capturer. But in this point some external constraints occurred. The photo-resistor cannot capture the laser under daylight. On the other hand, an appropriate laser will harm the user’s skin. Also, laser

pointer will not let us to send and receive data packages. That is a critical point in our development.

As a second way, we have thought that we can use a camera. First, a camera on the gun would capture the image at the direction of the gun when the user pulled the trigger and then, we would process that image to check whether a successful shot occurs or not. After some research, we have reached a conclusion that we cannot make it in an efficient way because the RF communication and PIC's capabilities are not enough for these kinds of complex algorithms.

After all, we have started to research for some alternative ways. At the end, we have found an indoor/outdoor game like our project in which infrared technologies are used. With this technology we can improve our project because IR allowed us to send data between player's guns and player's bodies (*)[1].

We had two big problems about IR technology. First, IR LEDs does not produce a linear light beam, so a shot will affect more than one player. Second, we will not be able to find a compact device which will satisfy data transfer needs. These problems have directed us to new literature researches, which will be described below.

3.1.2 Convex Lens

We have realized that this game is currently available in some places in U.S. and Europe after some researches and also we have found an open source version of this project. We have been looking for a way to make the IR beam linear, and we have seen that the creators of this game used an IR Module with a convex lens to focus the IR beam as in the figure below (*).[2]

[1] http://en.wikipedia.org/wiki/Laser_tag
[2] http://www.lasertagparts.com/mtoptics.htm

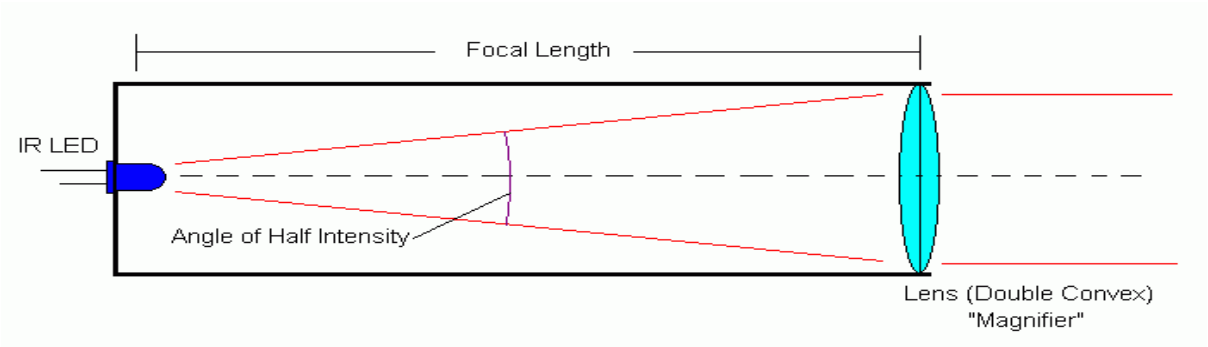

Figure 2 : Typical Lens Tube Design

3.1.3 IR Communication Protocol

Our IR communication protocol will be very simple and efficient for little amounts of data. As seen on the figure, an IR LED will produce infrared light for fixed amounts of time. That will be 1ms for bit '0', 2ms for bit'1' and 2ms break between any bits (*).[3]

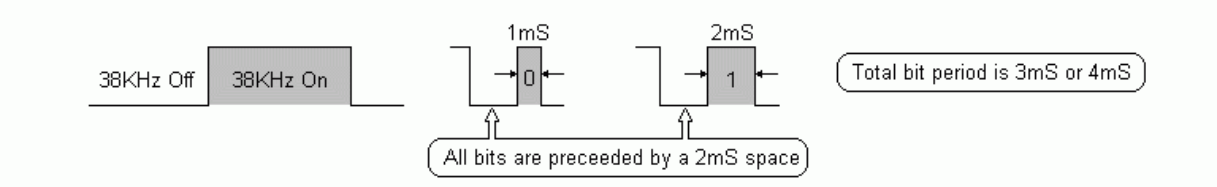

Figure 3 : IR Protocol

3.2 Market Research

In our local market research we have seen that there exists no such application. But In global market, there are several companies which designed this application as a game or a military simulation project. The most popular companies that designed this application as a

[3] http://www.lasertagparts.com/ltto.htm

game are Laser Tag and Miles Tag (*)[4]. We could not find an example for this application as a military simulation because of confidentiality. But we heard that Companies that works for Military Services developed this kind of applications before Laser Tag and Miles Tag.

When we decided to do this project, we wanted to discuss Laser Tag with people who played it in US or European countries. They told us their experiences about the game. The subject of the game attracts people very much. We see that in our country such games would be popular, because of our people's character. Depending on our local research and talking to ordinary people around, we have collected some ideas and opinions which feed our design.

To sum up, our aim is to be the monopoly of the market in our country.

4 Project Requirements

4.1 System Requirements

4.1.1 Software Requirements

4.1.1.1Server Side Software Requirements

Server side software is planned to control the initialization of the game, game flow (will be described later) and finalization. In addition, various information about the players and game play will be stored by this software.

Windows XP or later operating system has been chosen to develop and run this application.

We will develop this application on Microsoft Visual Studio 2008.

The server side software consists of GUI and will handle serial port events. So we decided to use C# as programming language at .NET framework. Because this programming kit is easy to implement, has a lot of GUI extensions and a reliable serial port library. Also we have such experience in this platform.

[4] http://www.lasertagparts.com/mtdesign.htm

For logging and storing game information, we are planning to use XML because XML is portable and reliable data storage.

4.1.1.2Device Side Software Requirements

PICC will be used as compiler for our device's software. We will develop our PIC programs in Windows environment since it is easy to debug.

PIC is responsible for the main part of our project and because of that the software which will be embedded to the PIC is very critical. Software on the PIC must be coded very efficiently and carefully. PIC is the bridge between IR and RF communication. Information of Player and gun is also stored and processed by PIC. In special cases such as "Player is dead" or "No bullets in the gun" PIC will forward this information to the server to further processes. Besides these, another job of the PIC is to refresh the LEDs of health indicator with the health information stored in the PIC.

4.1.2 Hardware Requirements

4.1.2.1RF Communication Hardware Requirements

RF communication kit which our sponsor (ANKIRA Electronic) will provide us to use in our project.

Only requirement with that kit is serial communication. We will give the data to RF via RS232 protocol. This data will come from IR communication part.

4.1.2.2IR Communication Hardware Requirements

In our project, IR transmitter is the gun; IR receiver is the body sensor and bonus. (Actually bonus and body sensor have same design).

Receiver:

38 kHz freq, 1 mA @5VDC, connected to PIC18Fxxxx. This captures the IR data sequence which reaches on to it and it returns an appropriate digital signal. PIC will handle the decode part.

Transmitter:

TSOPxxxx IR LED, 5VDC, 850-950 nm wavelength, connected to PIC18Fxxxx. With PIC18, this will be able to send IR data sequence (5-6 bits). PIC will handle the encode part.

850 – 950 nm IR led is best for our device (because of the optic stuff).

38 kHz will be enough for our IR communication. Because the data those will be transferred will be 5 or 6 bits long.

Development kit with PIC18 is provided from our department, so we are planning to continue our project with it.

4.1.2.3Health Indicators

A piece of LEDs will be used to represent the health of the player. The LEDs will be vertically placed on top of each other. Their colors will turn from green to red according to health information.

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.

4.1.2.4Device Side Physical Materials

Efficiency of IR communication is the most important part of this project, as our project mainly depends on IR.

The IR led does not send the IR light as beams. That's why the accuracy of the gun/sensor pair becomes unacceptable. To prevent this, by our literature survey, we have found that a lens and a pipe can be used.

By basic optics rule, if the light source is in the focal length of an appropriate convex lens, light will be collected. This makes the spread light a collected beam. As seen in the figure, the IR LED will be placed at the focal length of the lens in a pipe.

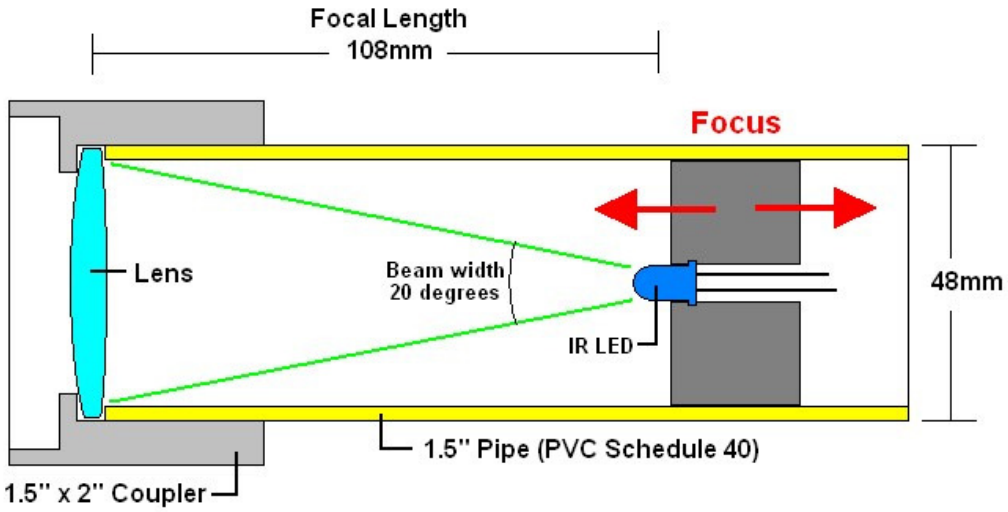

Figure 4 : IR Gun Sample

The reliability of the IR light will increase with this mechanism. Also, the proportion of the power of the LED and the distance of the target will be balanced (like in laser pointer).

The reflection area of the IR beam will be constant (related with the convex lens used in gun). This will be very useful in capturing the IR light. Otherwise, as distance changes, the reflection area will increase or decrease and no stability can be obtained.

The designed part is planned to have an outlook like the figure shown below.

Figure 5 : Final Outlook of the Gun

4.2 Functional Requirements

Device operations

- Gun

-Bonuses

-Body sensors

- Process and display unit

-User interface (server side, logging)

We will have 3 main components which are Player Kit (Gun and Body Sensors), Bonus Kit and Server side in our game design.

In the player kit, there will be a gun with IR Led on it and body sensors that has IR receiver and RF kit. The gun will send it's id to the other player's body when he/she fires it. At the body, this id will be captured by the IR sensor when the player shoots well directed. The id of the player who shoots the other player will be sent to the Server. When somebody is shot, that player's health degree will be decreased. This health information with the id of the player will be sent to the Server whenever the health decreases. We are planning to put a heath indicator to the body of the player to make it visible by other players. When somebody's gun is

out of bullet, Server will be alerted and at some predefined time Server will provide bullets to the gun using the RF kit.

In the bonus kit, there will be IR receiver and RF kit. Bonuses will be activated in some time decided by the Server. Server will activate the bonus kit and send a type to the kit, which defines whether it is a bullet bonus or a health bonus. The players should fire at the bonus kits to get the bonus. The IR receiver will capture the player's id when it is shot, and this id will be sent to the Server to identify who got the bonus.

In the server side, software has several functionalities such as game initialization, game play logging (score information, health information, bonus information etc.), bonus arrangement, and game finalization. Also this logging facility will give the users to see the old game score records

4.3 Non-Functional Requirements

We have three main non-functional but mandatory requirements.

- Low cost

- Portability

- Reality

Our project will consist of too many parts: Body kit, gun, server side software, IR kit, RF kit etc. We will make each of these parts in a way that the project will not cost too much. This is also a result of our market search.

Portability is one of the most important arguments in our project. If the game objects are not portable, there will be no way to present it or it will be very hard to arrange the game items.

The gun targeting and shooting system reality will determine our project's quality. If one targets to air and shoots, no one should get shot!

4.4 User Requirements

4.4.1 Use Case Diagram

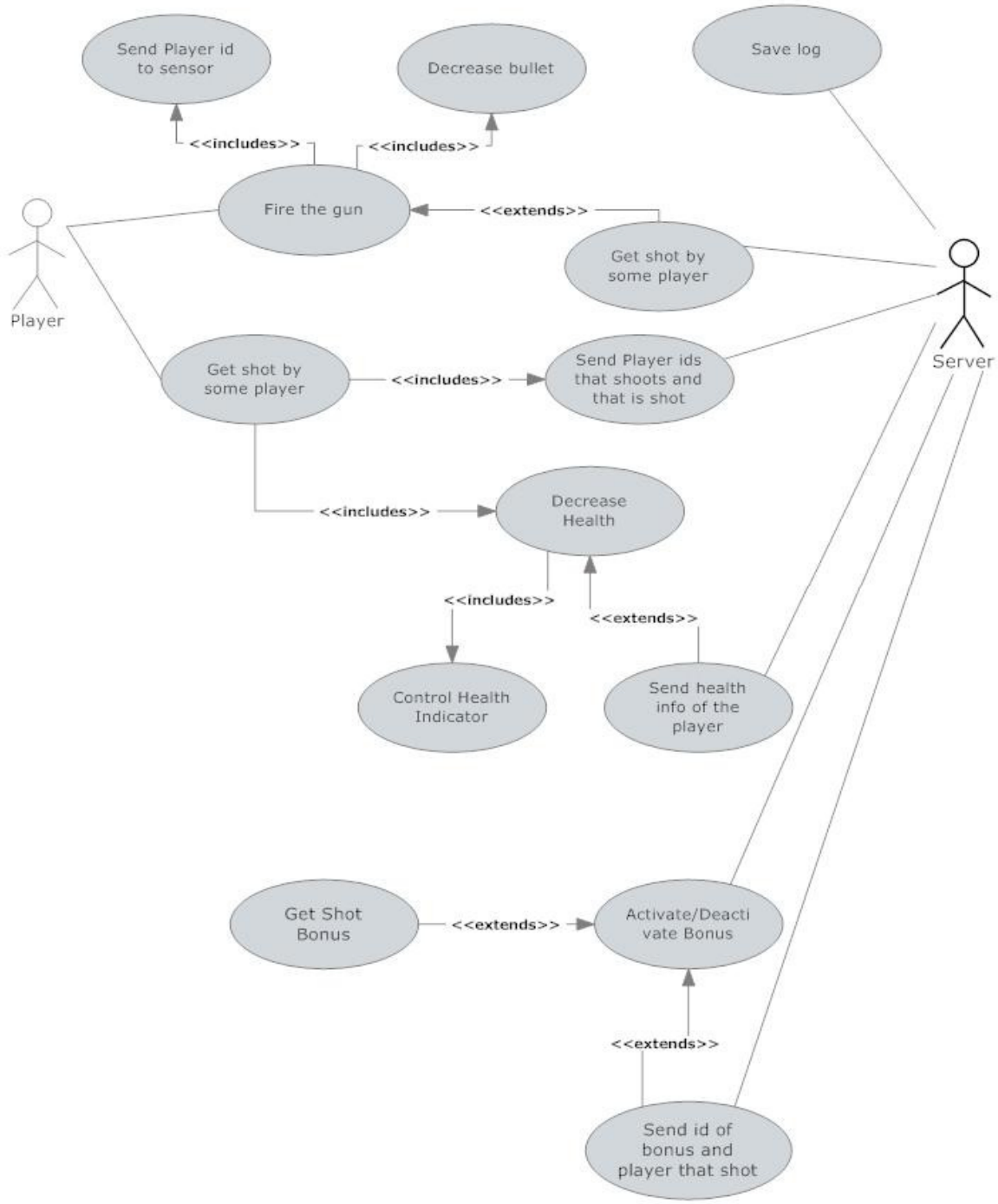

Figure 6 : Use Case Diagram

5 Project Modeling

5.1 Functional Modeling

5.1.1 Data Flow Diagrams

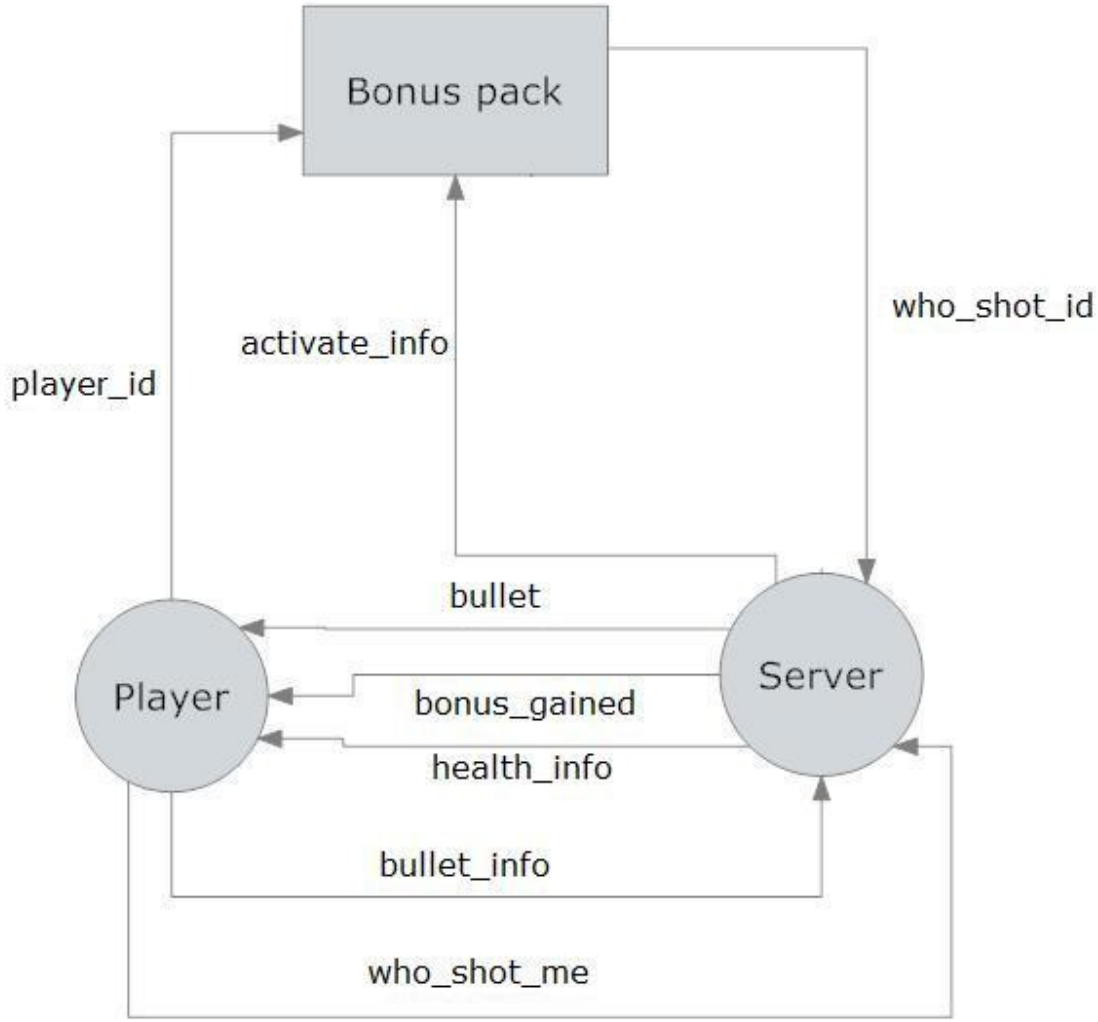

Figure 7 : Level 0 Data Flow Diagram

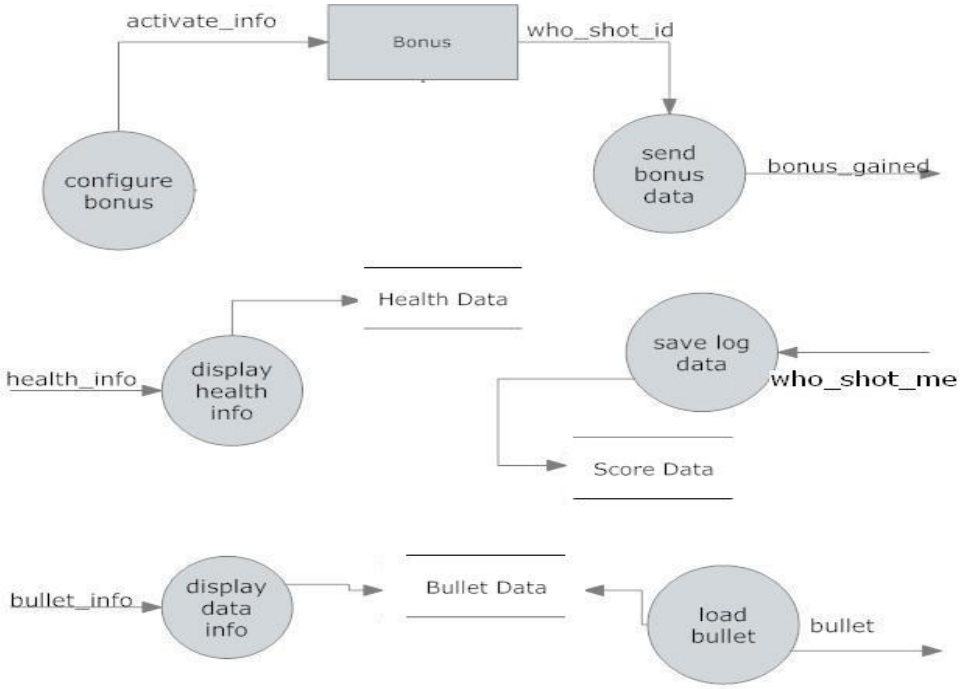

Figure 8 : Level 1 PC

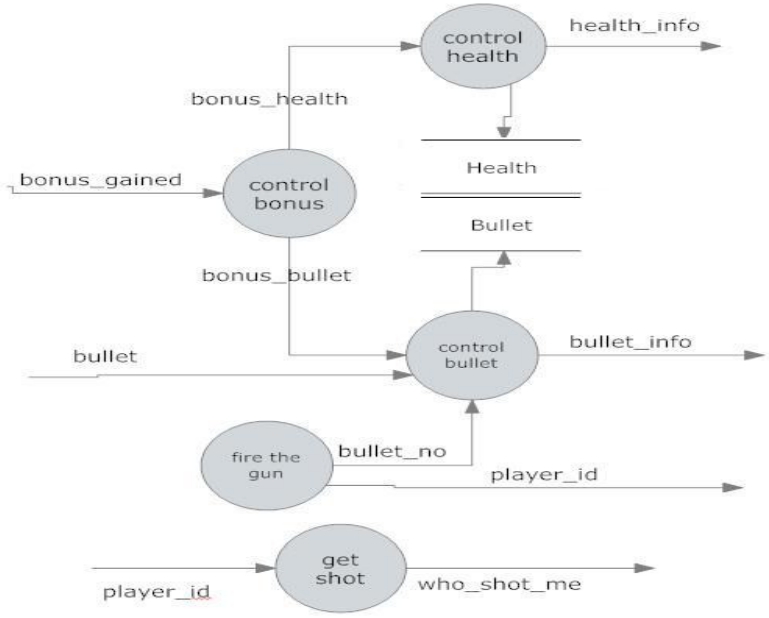

Figure 9 : Level 1 Player

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	bullet
Where used?	Level 0 , Level 1-Player,Level 1-Server Output from ‘load bullet’, Input to ‘control bullet’
Description	sends bullets to the guns of the players at some predefined intervals.

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	health_info
Where used?	Level 0 , Level 1-Player, Level 1-Server Output from ‘control health’, Input to ‘display health info’
Description	Player’s RF kit sends the health degree to Server and this information will be used in Server side user interface.

Name	bullet_info
Where used?	Level 0 , Level 1-Player, Level 1-Server Output from ‘control bullet’, Input to ‘display bullet info’
Description	Player’s RF kit sends the bullet info to Server and this information will be used in Server side user interface.

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.

6 Project Estimations

We decided to use intermediate COCOMO in project estimation. The reason why we choose intermediate model is that intermediate model includes cost drivers for subjective assessments of products, hardware, personnel and project attributes. Our KLOC estimation is 6.5. As our project is an embedded project, required lines of code for the project is lower compared to other projects.

The Intermediate Cocomo formula:

$E=a_i(KLoC)^{(b_i)}.EAF$

$D=c_b(E)^{d_b}$

$P=E/D$

Where E is the effort applied in person-months, KLoC is the estimated number of thousands of delivered lines of code for the project, and EAF is the factor calculated above. The coefficient a_i and the exponent b_i are given in the next table.

Software project	a_i	b_i	c_b	d_b
Embedded	2.8	1.20	2.5	0.32

Intermediate COCOMO computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes. This extension considers a set of four "cost drivers", each with a number of subsidiary attributes (*):[5]

Cost Drivers	**Ratings**					
	Very Low	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	

[1] Barry Boehm,*Software cost estimation with COCOMO II*

Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Use of software tools	1.24	1.10	1.00	0.91	0.82	
Application of software engineering methods	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

Calculated EAF=1.26

$E = 2.8 \times (6.5)^{1.20} \times 1.26$
$E = 33.4$ person-months

$D = 2.5 \times 33.4^{0.32}$
$D = 7.7$ months

$P = 33.4 / 7.7$

$P = 4$ people

7 Risk Management

To give clear risk information, we divided this part into 3 parts.

7.1 Hardware Risks

Risk definition	Probability of the risk	Reflection	Possible solution
Need of unexpected hardware parts	%25	critical	Urgent research about the part. Buying it if necessary.
Inconsistency of designed hardware	%20	critical	Make it work with software if possible , or change the nonworking part
Fatal errors in produced hardware	%10	critical	Urgent redesign and reproduction (partial or whole)
High cost of desired hardware	%30	normal	Looking for alternative ways or making it anyway

7.2 Software Risks

Risk definition	Probability of the risk	Reflection	Possible solution
Software design or implementation incompatible with the hardware	%15	normal	Redesign and reimplementation of software(partial or whole)
Inexperience in subject	%35	negligible	Urgent research about the subject

7.3 Team Management Risks

Risk definition	Probability of the risk	Reflection	Possible solution
Lack of time	%80	normal	Redevelopment of schedule
Lack of interest	%10	critical	Motivating each other anyway
Lack of meetings and arrangements	%10	normal	Changing team structure or meeting policy
Possible disagreements	%5	negligible	Respecting each other and solving it anyway
Lack of one person or more (with official excuses)	%10	critical	Increasing the workload of present members
Lack of one person or more (without official excuses)	%5	critical	Increasing the workload of present members and punishing the absent one(s)

8 Project Schedule

Number	Task	Start	End	Duration
1	Choosing the project	1/10/2008	8/10/2008	7
2	Understanding the project	10/10/2008	15/10/2008	5
3	Project Proposal	15/10/2008	16/10/2008	1
4	PROJECT PROPOSAL SUBMISSION	16/10/2008	16/10/2008	0
5	Technical meeting	21/10/2008	22/10/2008	1
6	Hardware research	22/10/2008	3/11/2008	12
7	Software research	22/10/2008	1/11/2008	10
8	Complementary material research	24/10/2008	8/11/2008	15
9	Hardware and Complementary material design	8/11/2008	10/11/2008	2
10	Requirement analysis report	8/11/2008	14/11/2008	6
11	REQUIREMENT REPORT SUBMISSION	14/11/2008	14/11/2008	0
12	Initial architectural design	17/11/2008	27/11/2008	10
13	Initial Hardware implementation	17/11/2008	27/11/2008	10
14	Initial Complementary material implementation	18/11/2008	28/11/2008	10
15	Initial Graphical User Interface Implementation	18/11/2008	28/11/2008	10
16	Testing the hardware components	28/11/2008	30/11/2008	2
17	First Prototype	25/11/2008	5/12/2008	10
18	Initial Design Report	17/11/2008	30/11/2008	13
19	INITIAL DESIGN REPORT SUBMISSION	30/11/2008	30/11/2008	0
20	Testing the software design	2/12/2008	7/12/2008	5
21	Development of first Prototype	5/12/2008	15/12/2008	10
22	Second Prototype	16/12/2008	21/12/2008	5
23	Final Design	10/12/2008	9/1/2009	30
24	FINAL DESIGN SUBMISSION	11/1/2009	11/1/2009	0
25	Prototype Demo Preparation	12/1/2009	18/1/2009	6
26	PROTOTYPE DEMO	18/1/2009	18/1/2009	0

9 Conclusion

Our project “HitMe” is a intersection point of software, hardware and physics. Without making a good research about the subject and requirements, we would surely mess up. We aimed to avoid this with a complete and satisfying requirement analysis report.

Hardware research has taken too much time, but finally we have found what we need. Also, we have finished our initial design related with the hardware.

The optic parts that we will use in our project are nearly ready. We have implemented an optical setup and experimented our ideas and it worked as we wanted. Together with a literature research, we have completed optic research.

We have arranged our schedule and risks such a way that nothing unexpected will happen.

To sum up, we have chosen this project because we wanted to combine our technical knowledge about hardware, software and other engineering concepts like optics. Our goal is to conclude our opinions and researches with this report.