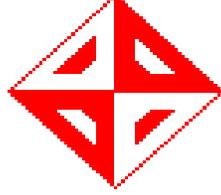


Middle East Technical University
Department of Computer Engineering



CENG491
Computer Engineering Design I
Requirement Analysis Report

ÖZGÜR YAZILIM



**Özgür Özgür
Fırat Erdoğan
Onur Demircan
Abdulkerim Mızrak**

FALL 2006

1. INTRODUCTION.....	4
1.1. PROJECT DEFINITION.....	4
1.2. PROJECT GOALS.....	4
1.3. SCOPE OF THE PROJECT:.....	5
2. PROCESS.....	5
2.1. Process Model.....	5
2.2. Team Organization.....	6
2.3. Major Constraints.....	6
2.3.1. Project Schedule.....	6
2.3.2. Language constraints.....	6
2.3.3. Execution Speed.....	6
2.3.4. User Interface.....	6
3. RESEARCH.....	6
3.1. Market Research.....	6
3.2. Technical Research.....	8
3.3. Literature Survey.....	9
4. PROJECT REQUIREMENTS.....	11
4.1. Functional Requirements.....	11
4.1.1. Menu Requirements.....	11
4.1.2. Game Flow Requirements.....	13
4.1.3. Operational and Structural Requirements.....	15
4.2. Non Functional Requirements.....	17
4.2.1. Usability and Playability.....	17
4.2.2. Reliability.....	18
4.2.2. Portability.....	18
4.3. Software Requirements.....	18
4.4. Hardware Requirements.....	18
5. SYSTEM ANALYSIS and MODELLING.....	19
5.1 Functional Model and Structured Analysis.....	19
5.1.1 Data Flow Diagram Level 0.....	19
5.1.2 Data Flow Diagram Level 1.....	20
5.1.3 Data Flow Diagram Level 2.....	21
5.2. Use Case Analysis.....	22

6. RISK MANAGEMENT24

 6.1. Scope24

 6.2 Project Risks24

 6.3 Risk Table26

 6.4.Overview of Risk Mitigation, Monitoring, Management...27

7. PROJECT SCHEDULE.....28

 7.1 Gantt Chart.....28

1. INTRODUCTION

1.1. PROJECT DEFINITION

Treasure Hunt is a massively multiplayer online game (MMOG) with 3D graphics. It is capable of supporting lots of online players to interact and play together in a virtual environment. The game has a nice scenario. Players are restricted in a multi-storey building with lots of rooms connected by several paths. Players are given the chance of choosing one of the pre-defined characters. A player then starts to wander inside the building and retrieve new tasks by finding different objects. There will be limited number of objects and several tasks will lead the players to the same objects. By this way, some of the players will be eliminated at each task, and finally only one of the players will reach the last object (i.e. the treasure)

1.2. PROJECT GOALS

Our goals in this project are mainly to create an interactive program combining 3D graphics with AI and network components. They can be grouped as:

3D Graphics: provide virtual reality by modeling real world objects with a simple and nice user interface

Network: Serve many users efficiently at the same time in a multiplayer environment.

AI: Simulate virtual players and prevent monotony.

Achieving these goals, we will end up with an enjoyable game running efficiently and maybe a commercial product attracting

gamers.

1.3.SCOPE OF THE PROJECT:

Treasure Hunt involves

- Rendering real world objects and designing 3D computer graphics with advanced graphics engines.
- Using a physics engine to simulate and predict effects under different conditions that would approximate what happens in real life or in a fantasy world.
- Using some audio engines to support virtual reality.
- Serving users via Internet by using network programming.
- Implementation of artificial intelligence and use of an engine for integration.

2. PROCESS

2.1. Process Model

We have chosen linear sequential model as our process model, because this model proposed by the course and there is a strictly progression though analysis, design, coding and testing. After the design phase we can use prototyping in order to get more in formations about the process and see possible problems.

The object-oriented approach will be applied in project development progress. We plan to increase modularity while on the contrary decrease the complexity of the system by creating components and packages.

2.2. Team Organization

We are aware of the importance of team structure for accomplishment of project. We choose a team leader for coordination and interaction between group members. Communication among group members is horizontal. All group members will work during all phases before according to their interests. Decisions are discussed and taken by all project members. These reasons lead us to choose democratic decentralized as team organization.

2.3. Major Constraints

2.3.1. Project Schedule

We have to finish the project until end of May 2007. During this time line we will go through every step. We have to design, build and test the project during this period. For project schedule refer to Gantt chart.

2.3.2. Language constraints

We will implement our project using C++ and MS .NET.

2.3.3. Execution Speed

We will provide at least 24 frame per second rendering speed for smooth playable game.

2.3.4. User Interface

We will created user friendly menus and interfaces. Menus should be easy to use and understandable.

3. RESEARCH

3.1. Market Research

Game programming is a very broad category in computer world. There are thousands of game and hundreds of game type, but as expected only some of them have a good popularity all over the world. Unfortunately, the reason why some games spread so fast and wide is unknown by anyone. To catch some clues about why some games spread so wide we examined the most popular game types.

Most popular kind of games and their definitions with some examples are listed below.

- **Role Playing Game (RPG) :**

A role playing game is typically a game that is very long and drawn in. Typically a RPG involves a group of characters that you need to build up or develop to reach a common end goal. To get to this goal, you will need to strengthen your character by leveling him up or gathering new equipment.

Some examples of RPG's are Final Fantasy, Dragon Warrior, Breath of Fire, Secret of Mana.

- **First Person Shooters (FPS):**

A first person shooter puts you into the role of a single character toting many weapons and the common goal of reaching the end, escaping doom and all that fun stuff. When playing in a multiplayer game, you can play in a FFA (Free For All), Team game or any other varieties.

Some fine examples of a FPS game would be Half-Life, Counter-Strike, Doom, Castle Wolfenstein.

- **Real Time Strategy (RTS):**

The role of a real time strategy game is to put the strategic part of your mind into the views of a fast paced environment. Typing your goal in a RTS game would be to defeat the opposing army with an army of yours that you have to build up by getting resources.

Fine examples of RTS games would be Starcraft, Warcraft, Total Annihilation, Command and Conquer, Age of Empires, Total War in Rome.

- **Sports Games:**

The sports genre typically denotes those sports at which you can play in real life. Sports like, skateboarding, snowboarding,

tennis, basketball, football and likewise. They feature real life pro teams and they put you into the game.

Some good examples of games like this would be Madden NFL, Cool Boarders, NBA Slam, Tiger Woods Golf, Brunswick Bowling, FIFA series.

- **Racing Games:**

Typically they revolve around one of the following: stunts, speed, tricks, collisions, or any of the above together. It can be trucks, snowboards, skateboarding, dune buggies, sports cars. Off-road Racing and Need for Speed are fine examples.

Treasure Hunt is not in any of these categories but it will look like First Person Shooters games with respect to multiplayer support and different camera views, and it will look like Role Playing Game with respect to Inventory future (capability of user to collect objects and use them). We hope that the success of FPS, and RPG will be repeated by our game with no violence practices.

3.2. Technical Research

Technical research is very important before starting the project. We have to specify which tools and libraries will be used in the project. Development environment, libraries, engines and tools must be carefully selected so that no integration problem will be faced up later. We conducted some research through internet, and got information from some experienced people to gain enough knowledge about the tools, libraries and development environments that will be used while constructing our game.

To create images and textures there are some valuable tools in the market:

- Adobe Photoshop
- CorelDRAW
- Reptile
- Macromedia Fireworks

Since all the project members are experienced in Adobe Photoshop we choose Photoshop. Fortunately, Photoshop is the most powerful and easy-to-use software to handle images and textures.

To handle 3D modeling there are some popular in the market:

- 3D Studio Max
- Poser
- Maya
- Cinema4D
- Life Forms Studio

Since one member of the project is experienced in the 3D Studio Max we choose it. As we research, 3D Max is one of the most powerful software in this area already.

To develop our game we examined two most popular libraries:

- OpenGL
- DirectX

Since, all project members take graph course, everybody has experience on OpenGL. As we research, there are not much difference between these libraries in terms of efficiency and capabilities. In addition to that OpenGL is an open source code, so that we will construct our game without any worries on copyright. Moreover, all the members are experienced in C++ language, so that using OpenGL will be a good choice again.

Moreover we will also use OGLE(openglextractor) for 3D graphics. OGLE is a software package that allows for the capture and re-use of 3D geometric object data from 3D graphics applications running on Microsoft Windows.

We will develop our game in Microsoft XP Professional operating system with Microsoft .NET platform. We select Windows due to the possible problems that may occur with Linux operating system, and because of window's widely usage in the world.

We have to decide which physics engine to use. We have searched the below list but we have not decided yet which one to use.

- Open Dynamics Engine
- OPAL
- Bullets
- Tokamak

3.3. Literature Survey

Since our game has to have network components, we have searched about network, internet, and their protocols. We benefit from a book about network, "Yourself TCP-IP In 14 Days", to learn a basic functionality of network. This research was very valuable for us, because no one of the project member was experienced in network.

Some basic information about network are briefly as below:

- TCP/IP

TCP/IP uses a 32-bit address to identify a machine on a network and the network to which it is attached. IP addresses identify a machine's connection to the network, not the machine itself—an important distinction. Whenever a machine's location on the network changes, the IP address must be changed, too.

- IP:

Internet Protocol (IP) is responsible for moving the packets of data assembled by either TCP or UDP across networks. It uses a set of unique addresses for every device on the network to determine routing and destinations.

- ICMP:

Internet Control Message Protocol (ICMP) is responsible for checking and generating messages on the status of devices on a network. It can be used to inform other devices of a failure in one particular machine. ICMP and IP usually work together.

- Ports and Sockets

All upper-layer applications that use TCP (or UDP) have a port number that identifies the application. In theory, port numbers can be assigned on individual machines, or however the administrator desires, but some conventions have been adopted to enable better communications between TCP implementations. **This enables the port number to identify the type of service that one TCP system is requesting from another.** Port numbers can be changed, although this can cause difficulties. Most systems maintain a file of port numbers and their corresponding service.

Typically, port numbers above 255 are reserved for private use of the local machine, but numbers below 255 are used for frequently used processes. A list of frequently used port numbers is published by the Internet Assigned Numbers Authority and is available through an RFC or from many sites that offer Internet summary files for downloading.

Each communication circuit into and out of the TCP layer is uniquely identified by a combination of two numbers, which together are called a socket. The socket is composed of the IP address of the machine and the port number used by the TCP software. Both the sending and receiving machines have sockets. Because the IP address is unique across the internetwork, and the port numbers are unique to the individual machine, the socket numbers are also unique across the entire internetwork. This enables a process to talk to another process across the network, based entirely on the socket number.

4. PROJECT REQUIREMENTS

4.1. Functional Requirements

4.1.1. Menu Requirements

The game will include two menus that provide the user to interact with the game. The content of these menus is determined, of course, according to the general properties of game.

a) User Authentication Menu:

Since Treasure Hunt is a Massively Multiplayer Online Game with 3D component, it requires not only installation on client machine, but also a connection to the server machine with a valid identification.

This menu is used to manage login issues and has following items:

- New User

This item will lead user to the sign up screen where he/she registers his/her profile information (e.g. username, password, name, e-mail, etc.)

- Username & Password Fields for existing users

These items will be user to identify the existing users. After this validation process, they will be able to enter the game main menu.

b) Game Main Menu

This menu involves items to start a game. The items in this menu are:

- Join Game

This item is used to register user for the game of next session. Server starts each session of game in a predefined time period. The player that wants to join the game has to wait current session to terminate.

- Choose Character

This item is to make the user select one of pre-defined characters in the game.

- Options

This item provides change of graphics, audio and keyboard controls.

- Manage Profile

This item leads user to a page to change his/her profile information.

- Exit

This item is used to exit from the game. Selecting this item

c) In Game Menu

- Objectives

The last task assigned to that user is seen by this item.

- Map

A 2D map of the current storey is observed from top view by this menu item.

- Chat

Player can chat with each other.

d) Game Pause Menu

By this menu only the player pauses.

- Leave Game

The user will quit from the current game session.

- Options

This item provides change of graphics, audio and keyboard controls.

- Return To Game

This item provides to resume game.

4.1.2. Game Flow Requirements

These are the requirements needed in before-game and during-game periods. These requirements involve abilities of the

characters, character-character interactions, character-game interactions, and logical harmony of the game.

a) Game Logic Requirements

- The game starts a new session, when a pre-defined time passed after the last game session.
- The session will start only when a pre-defined number of players are available. If there is less than the minimum number of players for a game at new session time, the game starts with the help of some virtual players supplied by server.
- The game consists of a pre-defined number of steps. The user should not start the next step, before completing previous ones. The difficulty of steps increase through the chain of tasks. The final clue directs the player to the treasure.
- The player cannot leave the building. But he/she can wander inside the allowed rooms in the building.
- Players chat during and before games with each other.

b) Environmental Requirements

- Building:

The multiple storey building is the main environment that the game takes place. There's no way out. But, the building consists of many rooms, some of which are locked.

- Objects

i) Movable Objects:

The only movable objects are the ones that are used as clues for the next step. So they can be carried by the characters in the game.

ii) Static Objects:

There will be furniture (tables, chairs, etc.) and electrical devices (like televisions, ovens) that are not movable.

- Characters

- i) They walk
- ii) They run
- iii) They jump
- iv) They turn
- v) They pick an object
- vi) They carry clues
- vii) They open doors
- viii) They move upstairs and downstairs between storey

c) Artificial Intelligence Requirements

- Virtual Characters

They are characters with behaviors which are directed by AI components. Same as real characters they do the things stated above. Moreover

- i) They can interpret clues
- ii) They act according to the clues
- iii) Virtual players have different intelligence levels.

4.1.3. Operational and Structural Requirements

a) Game Engine

Game Engine is core component of Treasure Hunt game. It provides interaction between graphics, audio, server-client interaction, physics engines and AI components. It is the code required specifically to make that game work and decides how controls work according to the user response.

b) Graphics Engine

This engine will render real world objects and design 3D computer graphics. The models will be texture mapped by this engine.

c) Audio

Game will have different soundtracks during game and waiting for the next session. The soundtrack will change between the steps.

Sound effects will be used during games.

Sound options will be controlled and volume level is adjusted by user.

d) Network Component

The network component is used for data transfer between server and client during game.

Network component is also used for authentication of players to the game engine.

The response given by client will be transmitted to the game engine and the decision given by game engine will be distributed.

e) Physics Engine

It will simulate and predict effects under different conditions that would approximate what happens in real life.

It will be used to make Treasure Hunt game more realistic.

f) Artificial Intelligence Component

Virtual characters are controlled by artificial intelligence component.

This component will provide virtual players if adequate number of players are not reached.

g) Game Data

Buildings

Objects

Characters

Textures

Images

Sound Effects

Soundtrack Music

4.2. Non Functional Requirements

4.2.1. Usability and Playability

Obviously if a game does not have any of these two futures, it will not be played by any player. To supply these two futures, game engine has to provide interaction between network component, audio, graphics engine, physics engine and artificial intelligence component. Since Treasure Hunt is a multiplayer game, server-client interaction must be fast enough to coordinate players in the game. The server has to serve all players in real time. Players have to play the game synchronously. Moreover, we have to make creative use of a keyboard-specific feature that

greatly enhances the playability of the game. Keyboard dependencies will be lowered as possible as we can. We have to provide mouse or joystick option to play the game. Sounds and soundtracks will be selected carefully.

We have to design the menu items and controls as possible as simple and accurate.

4.2.2. Reliability

Treasure Hunt has to be as possible as error and bug free. We have to prevent errors and we have to use fault detection mechanism.

4.2.3. Portability

Treasure Hunt can be installed on any personal computers which have Windows Operating System and a connection to the internet.

4.3. Software Requirements

User Requirements:

Windows Operating System with network connection.

Developer Requirements:

Microsoft Visual .NET 2006

Adobe Photoshop 7.0

OpenGL

3D MAX

Tokamak (most probably)

OGLE

4.4. Hardware Requirements

A new generation graphics card with 3D support.

5. SYSTEM ANALYSIS and MODELLING

5.1 Functional Model and Structured Analysis

5.1.1 Data Flow Diagram Level 0

Level 0 DFD in Figure 1 of 5.1 is the overall system. It shows the interaction between user (game client) and game server. Game Data is transferred from game clients to game server and from game server to clients.

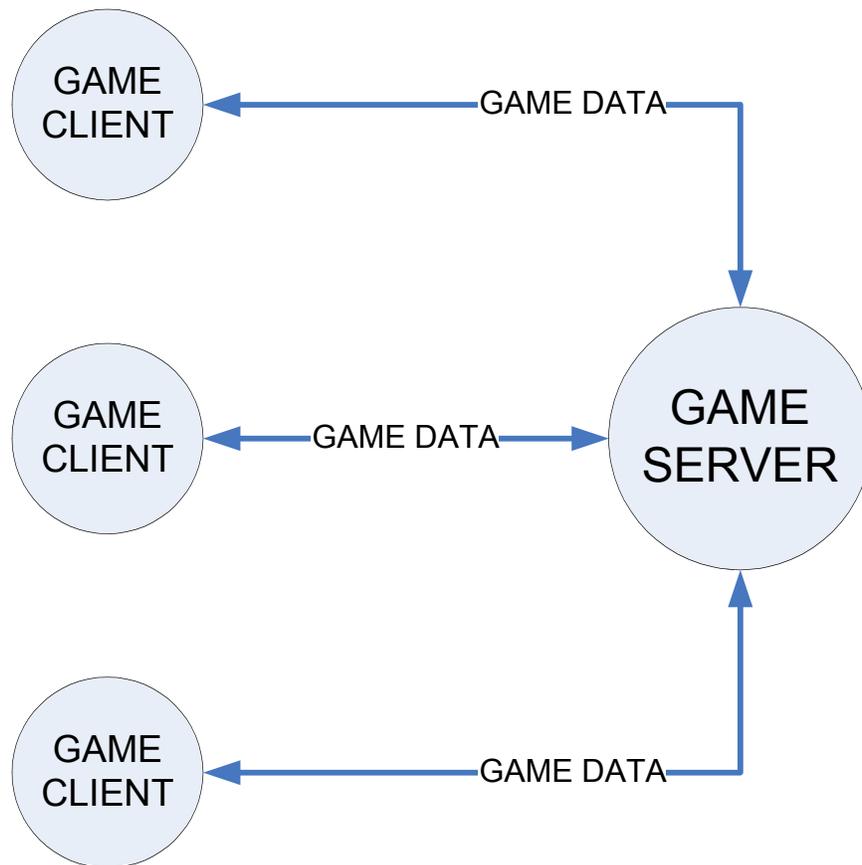


Figure 1 of 5.1

5.1.2 Data Flow Diagram Level 1

Network component provides data transfer between game client and game server in Figure 2 of 5.1. Game engine sends game data to AI engine and takes AI players (virtual characters) data, such as its behaviour, its coordinates.

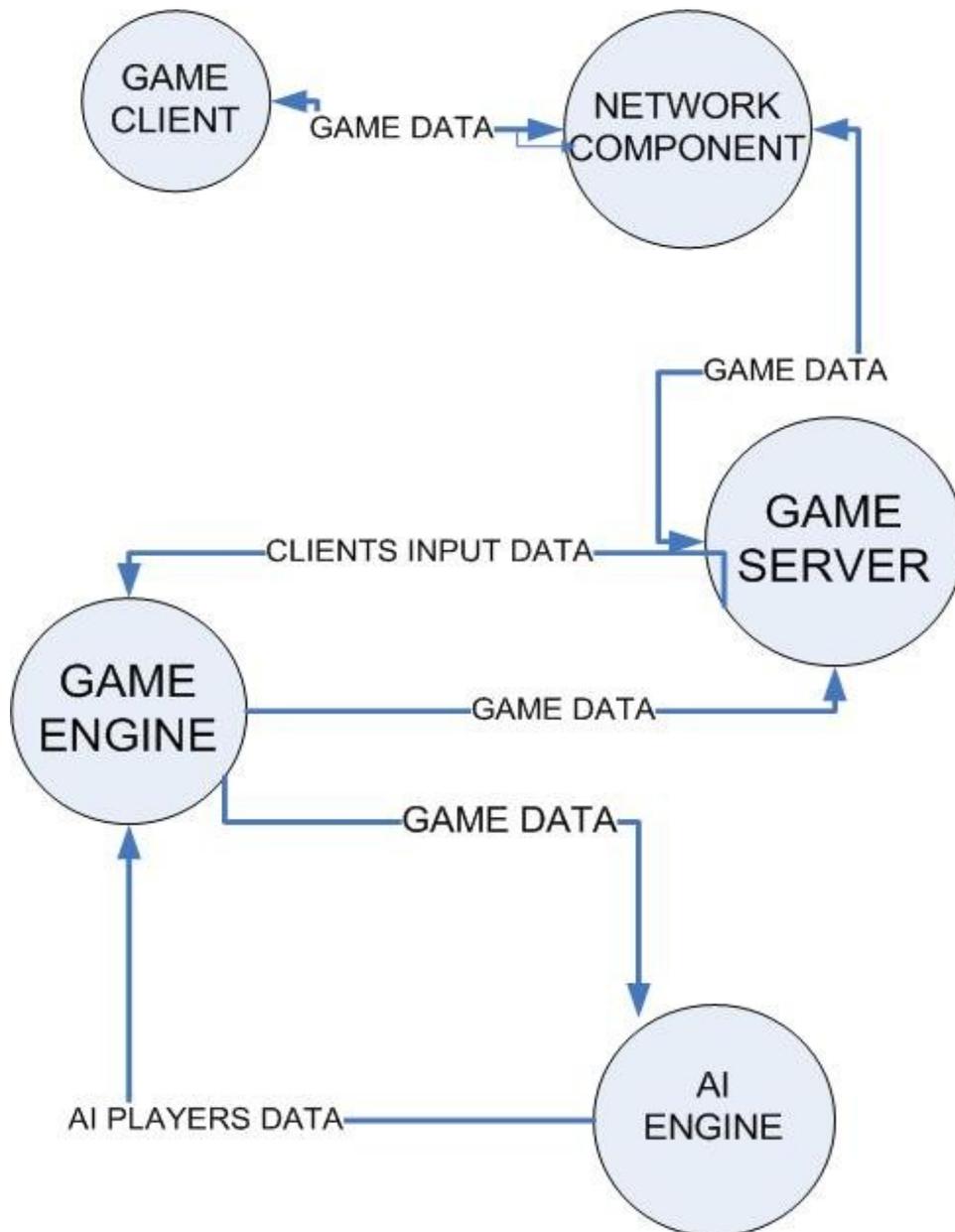


Figure 2 of 5.1

5.1.3. Data Flow Diagram Level 2

Figure 2 of 5.1 is more detailed look on overall system. Inner structures and sub-modules are described in a detailed way. Player will provide inputs using keyboard and mouse and take graphics and audio outputs. Game client sends scene data to graphics engine and graphics engine sends graphics outputs to the game client. Game client sends game state to audio and player take audio input.

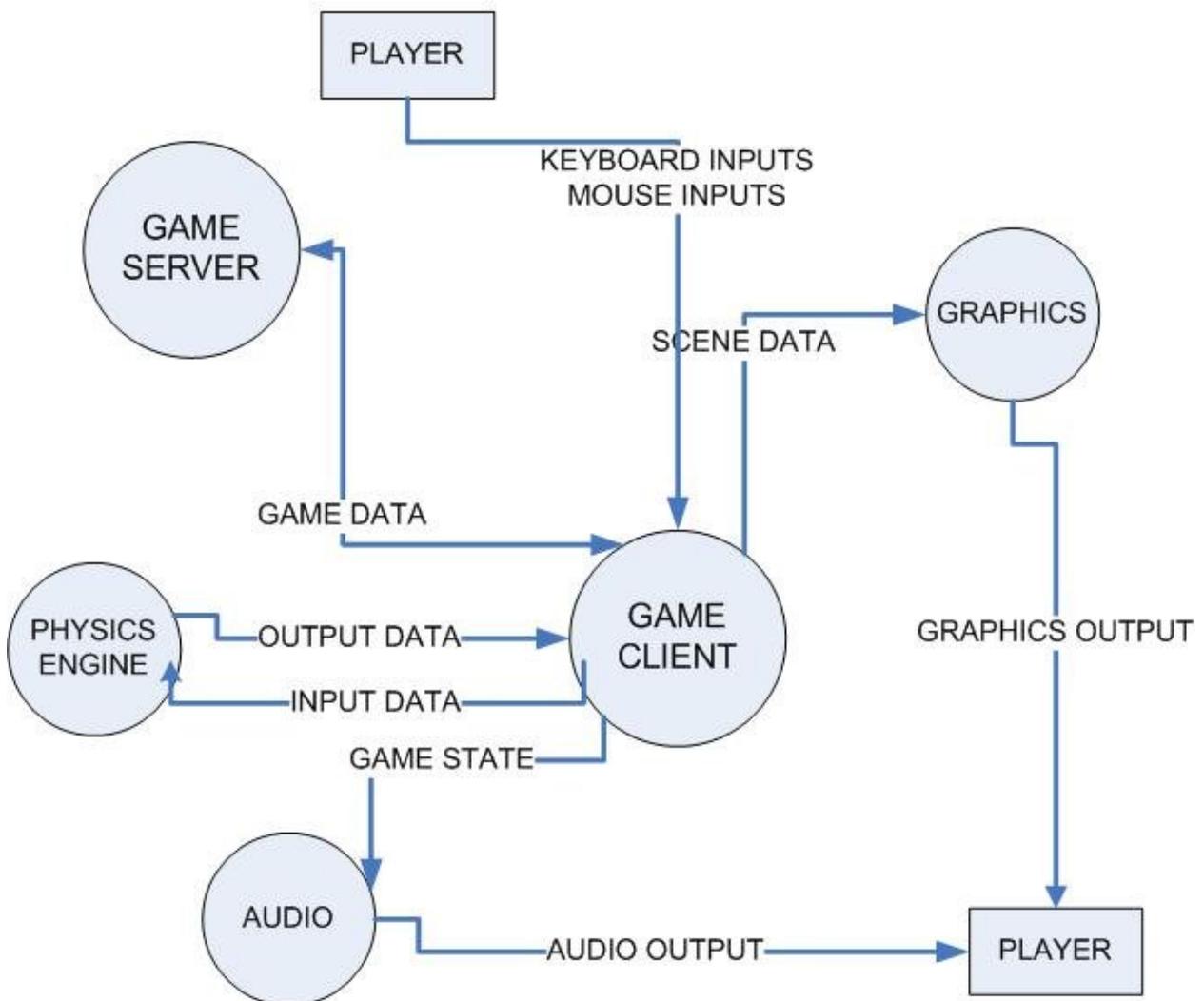
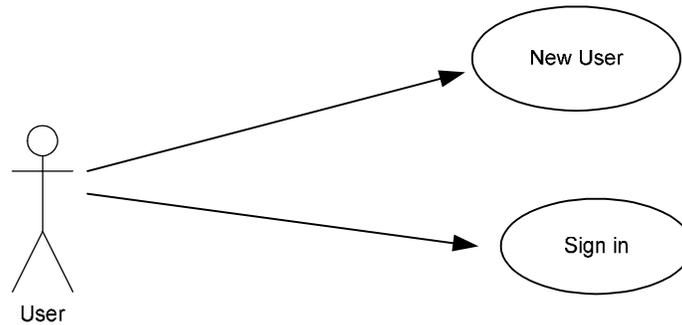


Figure 3 of 5.1

5.2. Use Case Analysis

These are uses cases in Treasure Hunt.

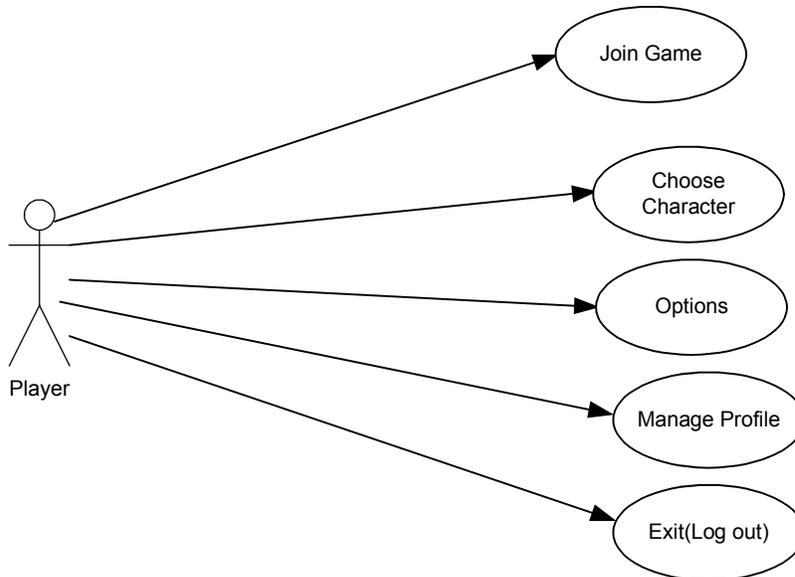
a) User Authentication Menu Use Case:



User has to become a member after installing game on his/her computer to play the game. After becoming a member he/she has to sign in with correct username and password to join the game on the server.

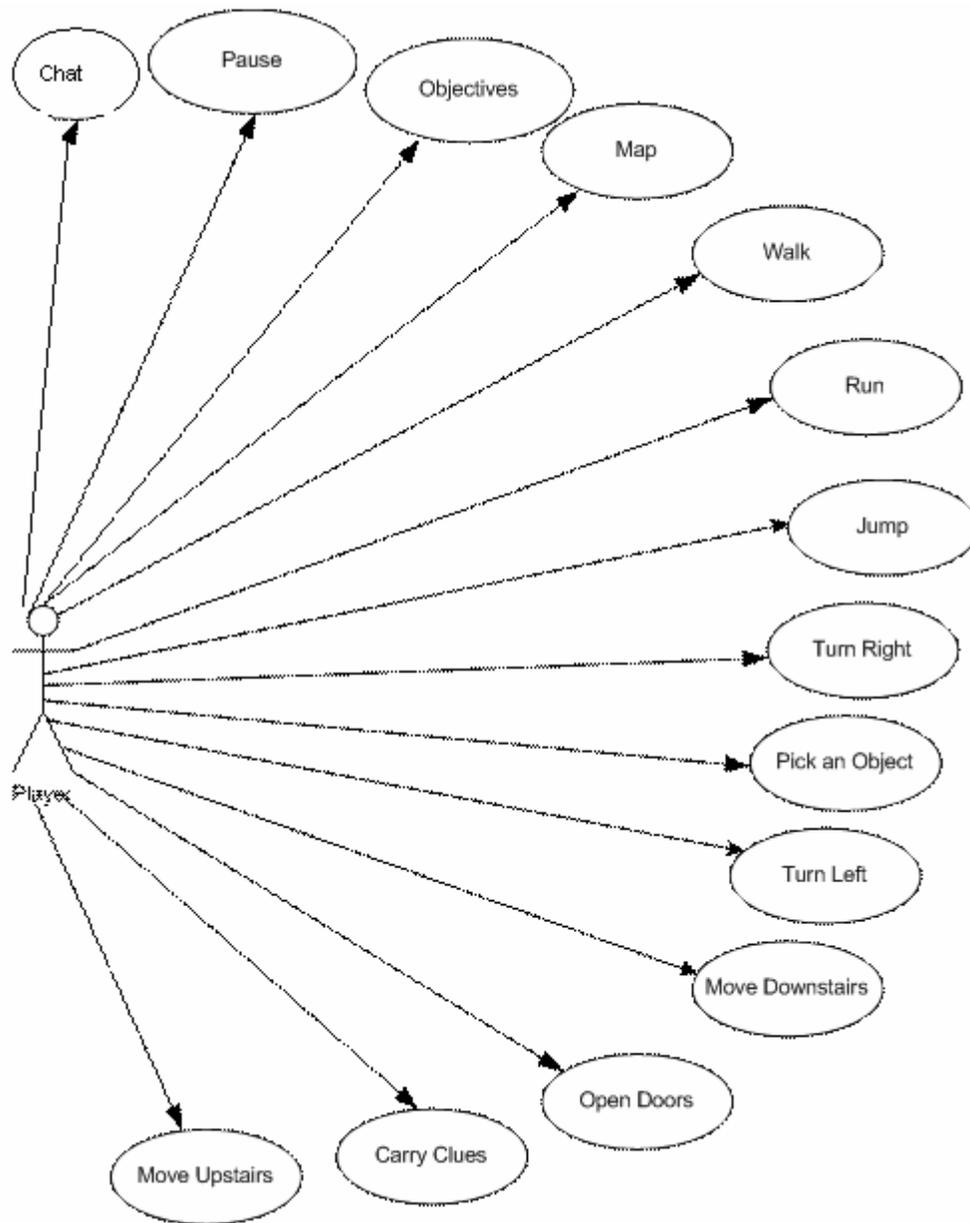
b) Game Main Menu Use Case:

These are the use cases that user will face while joining the game.



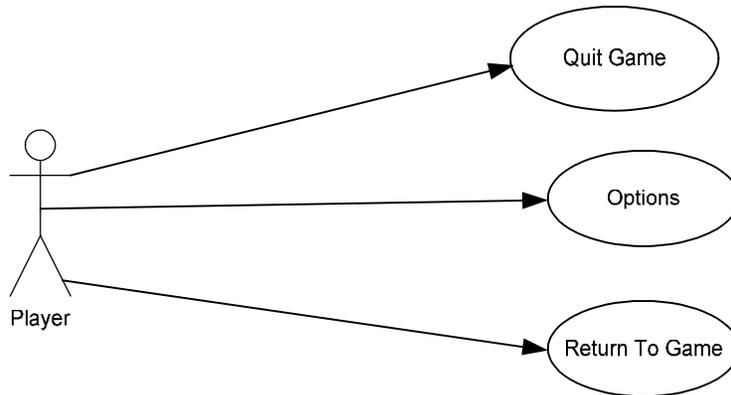
c) In Game Menu Use Cases

These are the use cases that user will face while playing the game.



d) Game Pause Menu

These are the cases that user can do while pausing the game.



6. RISK MANAGEMENT

6.1. Scope

In software projects there are always risks that may occur. So risk management is one of the facilities that should be performed in every project in order to protect the project failure or late submission.

6.2 Project Risks

Lack of Roles and Responsibilities: Since our group member has lack of professional experience, there is always risk about not fulfilling responsibilities.

Lack of Knowledge on Subject: The project may be affected because lack of knowledge on project topic. Because our group members don't have any experience on computer network, and have not much experience on computer graphics. Also tools, libraries and packages are new to group members. In order to avoid this risk we will study on computer graphics and network areas, and develop applications for get experiences before coding phase.

Member's Failure and Lack of Time: Group members can have lack of time because of their homework, exams and other projects. This can be a problem in project schedule. So obeying the schedule is important.

Misunderstanding Requirements: If we don't determine our project's specifications clearly and plan these in design phase, modifications will be required.

Tools, Libraries, Platforms: Tools, libraries, platforms that will be used in our project must be chosen very carefully, and meet all our needs. Because it will be very difficult changing these elements after design phase, and cause losing time.

Lack of Standards: The lack of standards, source code and end products may not be high quality. We will decide quality standards and use these standards in all phases to avoid this problem.

Large Project Size: Due to lack of experience the project size may not be estimated, so this may cause not finishing some parts of the projects.

6.3 Risk Table

Risks	Probability	Impact
Lack of Roles and Responsibilities	%30	2
Lack of Knowledge on Subject	%40	1
Member's Failure and Lack of Time	%20	1
Misunderstanding Requirements	%25	2
Tools, Libraries, Platforms	%10	2
Lack of Standards	%60	1
Large Project Size	%40	3

Impact Values:

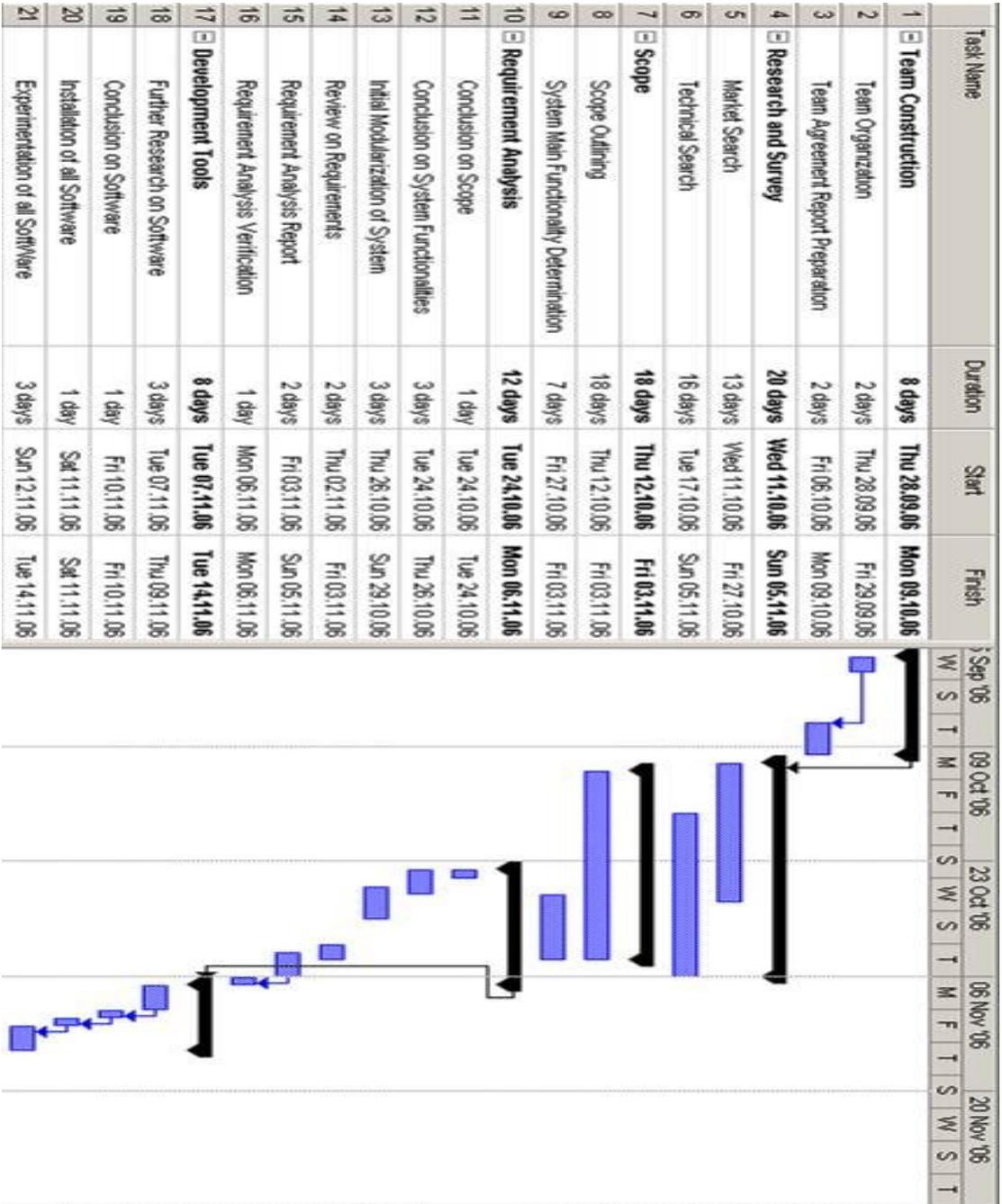
1. Negligible
2. Marginal
3. Critical

6.4. Overview of Risk Mitigation, Monitoring, Management

Risks	Mitigation	Monitoring	Management
Lack of Roles and Responsibilities	Assign a reviewer to every member	Detect the deficiency in tasks	Assign considerable task
Lack of Knowledge on Subject	Get help from experienced people	Detect subjects with less experience	Assign task to learn the subject
Member's Failure and Lack of Time	Decrease the functionality of the project	Make everybody remember the GanChart weekly	Working very hard.
Misunderstanding Requirements	Negotiate with each other	Detect the possible such requirements before	Working very hard to finish on time
Tools, Libraries, Platforms	Get help from friends	Detect unexperienced tools	Choosing known tools by project members
Lack of Standards	Forming our own standards	Detect the tools out of standards	Get help from experienced people
Large Project Size	Try to finish maximum of it	Working very hard	Lessen the scope of project

7. PROJECT SCHEDULE

7.1 Gantt Chart



Task Name	Duration	Start	Finish
19 Conclusion on Software	1 day	Fri 10.11.06	Fri 10.11.06
20 Installation of all Software	1 day	Sat 11.11.06	Sat 11.11.06
21 Experimentation of all Software	3 days	Sun 12.11.06	Tue 14.11.06
22 Initial Design	15 days	Tue 14.11.06	Tue 28.11.06
23 Design Graphical User Interface	3 days	Tue 14.11.06	Thu 16.11.06
24 Detailed Modularization of System	3 days	Fri 17.11.06	Sun 19.11.06
25 Sub-Module's Design	4 days	Wed 22.11.06	Sat 25.11.06
26 Design Data Structures	2 days	Wed 22.11.06	Thu 23.11.06
27 Design Class Hierarchy	1 day	Fri 24.11.06	Fri 24.11.06
28 Design Sub-Classes	1 day	Sat 25.11.06	Sat 25.11.06
29 Initial Design Report	2 days	Sun 26.11.06	Mon 27.11.06
30 Initial Desig Verification	1 day	Tue 28.11.06	Tue 28.11.06
31 Project Presentation 1	15 days	Tue 05.12.06	Fri 22.12.06
32 Preparation of Presentation 1	4 days	Tue 05.12.06	Fri 08.12.06
33 Preparation of Presentation 2	4 days	Tue 12.12.06	Fri 15.12.06
34 Preparation of Presentation 3	4 days	Tue 19.12.06	Fri 22.12.06
35 Final Design	33 days?	Tue 05.12.06	Mon 15.01.07
36 Final Design of Graphical User Interface	6 days	Tue 05.12.06	Mon 11.12.06
37 Conclusion of System Modules	6 days	Tue 12.12.06	Mon 19.12.06
38 Final Design of Sub-Models	12 days?	Fri 15.12.06	Mon 01.01.07
39 Final Design of Data Structures	4 days?	Fri 15.12.06	Wed 20.12.06
40 Final Design of Class Hierarchy	4 days	Thu 21.12.06	Tue 26.12.06
41 Final Design of Sub-Classes	4 days	Wed 27.12.06	Mon 01.01.07
42 Multimedia Design	8 days	Tue 02.01.07	Thu 11.01.07
43 Design of Models	8 days	Tue 02.01.07	Thu 11.01.07
44 Design of Audio	8 days	Tue 02.01.07	Thu 11.01.07
45 Final Design Report	3 days	Fri 12.01.07	Sun 14.01.07
46 Final Design Verification	1 day	Mon 15.01.07	Mon 15.01.07
47 Prototype	15 days	Fri 05.01.07	Tue 23.01.07
48 Design of Prototype	9 days	Fri 05.01.07	Mon 15.01.07
49 Preparation of Prototype	5 days	Tue 16.01.07	Mon 22.01.07
50 Demo	1 day	Tue 23.01.07	Tue 23.01.07

