REQUIREMENT ANALYSIS REPORT

by

MIR

N. Özlem ÖZCAN - 1250596
Hamza KAYA - 1250448
Mustafa Öztorun - 1250638
Ozan Şimşek - 1250760

TABLE OF CONTENTS

INTRODUCTION

PURPOSE OF THE REPORT

This report is written for establishing a starting step for the specifications of 3D Adventure Game project. The document is going to form a roadmap for future steps of the project and cover the details of requirements.

PROJECT DESCRIPTION

GENERAL DESCRIPTION

Computers were barely seen as useful tools in business, and were viewed as having even less potential for entertainment. Technology has a way of changing the way we think about things, and it wasn't long before early video games captured the minds and quarters of an entire generation. Video games are interesting in that they represent the first form of interactive digital entertainment. Perhaps more importantly, video games represent the first truly interactive art form. When you think about it, there aren't many paintings, sculptures, or musical compositions that allow you to interact with them and change them in any way. Video games allowed their creators to share something more with the game player by giving players a chance to put part of themselves into a game, and then see what comes out.

In fact, the best games are the ones that allow players to express their own unique style and technique when playing the game, while at the same time being entertained by the game designer's vision of an imaginary world portrayed by the game. What makes a game better is an interesting scenario, better 3D graphics including light & shadow effects, different camera views, realistic texturing, sprite animations etc, intelligent ‘bad guys’, sampled digital sound effects and a realistic physics engine.

The purpose of this project is to develop a 3D action-adventure game. Keeping the above points in mind, we aim to include the following features to our game:

Scenario

The game will consist of different levels. In each level hero will have a quest to complete the level. In order to move to the next level he/she has to solve puzzles, investigate the environment, fight with creatures, gather information. Game will start with an intro giving background information about the scenario. Furthermore, there will be informative parts related to the scenario throughout the game, especially between levels.

3D Graphics

Unlike the common 3D games, this game will include two different view modes. First-person and third-person views will be used interchangeably. In the third-person view, camera will follow the hero. A wide area will be viewed in this mode. In the first-person mode the player will see the world from the eyes of the hero. Objects and environment will be relatively big compared to the third-person view mode.

Game will include levels which are taking place in day and night. During the day time, sun will be the major light source, while the street lambs will be the source at nights. In the indoor scenes, lambs will be the light source both for day and night.

Also some of extra 3D effects like weather effects, smooth transitions between different viewing modes will be integrated into the game.

Good graphics and performance has always been a big trade-off. Therefore while doing the above facilities; performance should always be kept in mind.

Sound

It is hard to argue the impact of compelling sound effects and music on video games of all types. Sound effects are important artifacts for creating fascinating game atmosphere. To accomplish this, we will integrate continuous background music and realistic sound effects. Background music will be well-adjusted to the scenes.

Game Controllers

The keyboard and mouse are by far two most standard user input devices on computers and are just about guaranteed to be supported on all PCs. To ease the life of the player, we specified different controllers for different modes. In third-person view mode, mouse will be used as the main controller whereas the player interacts with game by keyboard in first-person view mode. Also he/she will be able to customize the shortcut keys.

Game Engine

A game engine represents an organization of the code for a game so that general application tasks are separated from game specific tasks. Additionally using a game engine will allow us to simplify the code for our game and let us focus our attention on the game code that matters most. Our engine will address the unique needs of our game such as initialization, detecting collisions, managing memory, file and window operations, rendering 3D scenes and input reading.

Artificial Intelligence

AI is defined as the techniques used to emulate the human thought process in a computer. In our game related AI will be a relatively small subset of whole AI knowledge. The objects (creatures) in the environment will react intelligently to the hero’s actions and this will make the player feel the effects of his/her actions more deeply.

HARDWARE and SOFTWARE REQUIREMENTS

Hardware Requirements

User Side (Minimum)

- Pentium III-1GHz/ AMD 1GHz Processor
- 128 MB of memory
- 64 MB of video card (display adaptor) with 3D support
- Some free disk space for installation
- Sound card

Developer Side

- Pentium III-500MHz/ AMD 500MHz Processor
- 64 MB of memory
- 32 MB of video card (display adaptor) with 3D support
- Some free disk space depending on the size of company needs
- Internet connection

Software Requirements

User Side

- Windows 98-2000-XP Operating System

Developer Side

- Microsoft Office and Visio
- C++ Compiler (Microsoft Visual C++)
- Open GL
- 3D Max
- Adobe Photoshop
- Corel Painter & DRAW

PROJECT MODELS

USE CASES

Configure Graphics Card Options

In order to change the graphics card settings the player should enter the “Options” menu by either using mouse or keyboard. From this menu, player will use “Graphics” option to customize these settings. In this new screen player will be able to change the resolution and the color depth of the game. By clicking the “OK” button, player will save the settings.

Configure Sound Card Options

In order to change the sound card settings the player should enter the “Options” menu by either using mouse or keyboard. From this menu, player will use “Sound” option to customize these settings. In this new screen player will be able to change the volume of the game music and game sounds. Player can save these settings by clicking “OK”.

Configure Game Controller Options

In order to change the controller settings the player should enter the “Options” menu by either using mouse or keyboard. From this menu, player will use “Customize Controls” option to customize these settings. In this new screen player will see a set of actions and shortcut keys that are assigned to these actions. To change a shortcut key, player have to first click on the action and than press the key (or mouse button) that will replace the old one. Also player can reset the shortcuts to their default values by clicking the “Defaults” button.

Start New Game

For starting a new game, the player selects “New Game” button in the “Main Menu”. The player can select one of the default or previously defined characters from the “Character Selection” screen. Or the player can also customize a new character by following the “Customize” link in the same screen. After selecting or customizing the hero, the player starts the game by selecting the “Play Game” button with mouse or keyboard.

Exit From Game

In order to exit from the game, the player should press ESC button. To quit to the main menu, s/he should select “Main Menu” button in the appearing game menu. To quit to windows, s/he should select “Quit to Windows” button in the appearing game menu.

Load Game

Player should choose “Load” from “Main Menu” to load a previously saved game. In load game screen all the saved games will be displayed. Moving the mouse over a saved game will display the last frame of the saved game. The player has to double-click on the file to load it.

Save Game

After each level the player will be asked whether he/she wants to save the game or not. After choosing “Yes” the player will be asked to provide a file name. If the player select “No” he/she will continue with the next level. During the game player will not be able to save the game in any other way.

Move Character

Game will have two different views. First one will be a third-person view. During this mode player will move his/her hero by using mouse. Clicking an available target area will make the hero move there. The other view will be first-person. In this view player will use the keyboard to route and mouse to look. Player can make the hero run by pressing a previously specified shortcut key.

Solve Puzzles

In every level player may face a set of puzzles. Player will try to solve the puzzles by following the instructions given at the begging of the puzzle.

Take & Drop Objects

In the third-person view, player should click on the item in order to put the item into his/her inventory. In the first-person view, player has to walk over the item to get it. For both view modes player can drop an item by drag & drop technique.

Interact With ‘Others’ (monsters, friendly creatures etc.)

In the third-person view, player will click on the ‘others’ in order to attack/talk etc.

Read Diary

Player will have a diary containing all the quests and major events. To reach this diary player will press a predetermined key. The player may use the icons to turn the diary pages.

Improve Skills

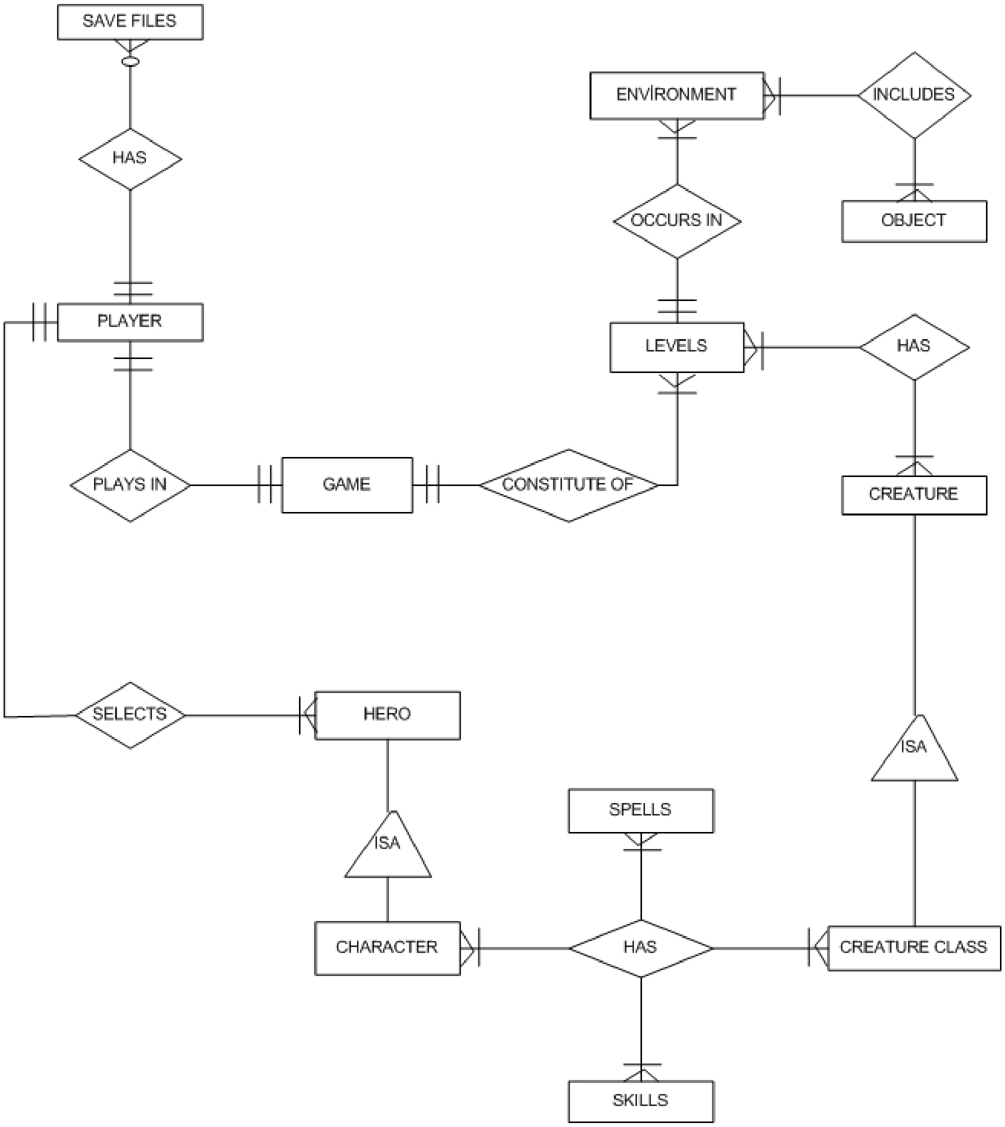
Player will be given a number of skill points at specific game states. Player can distribute these points among the skills that are displayed on the screen by clicking on the icon on the sides of skills.

Improve Spells

Player will press a predetermined key to reach the spell tree. By clicking on the spells at this tree, player will be able to either increase the level of a previously learned spell or learn a new spell (if he/she has a spell point).

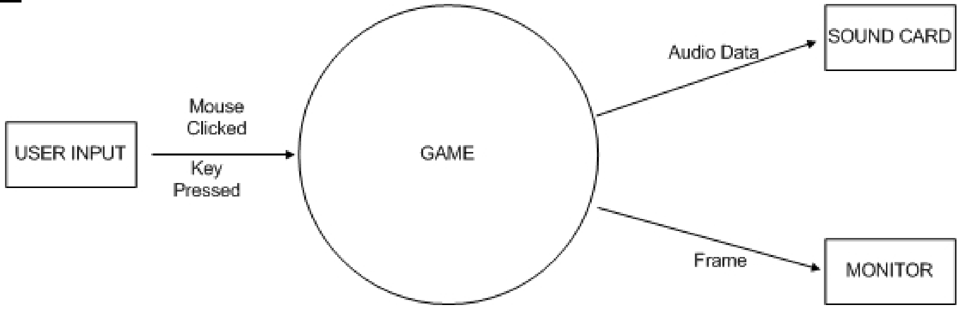
DATA MODEL

ER DIAGRAM

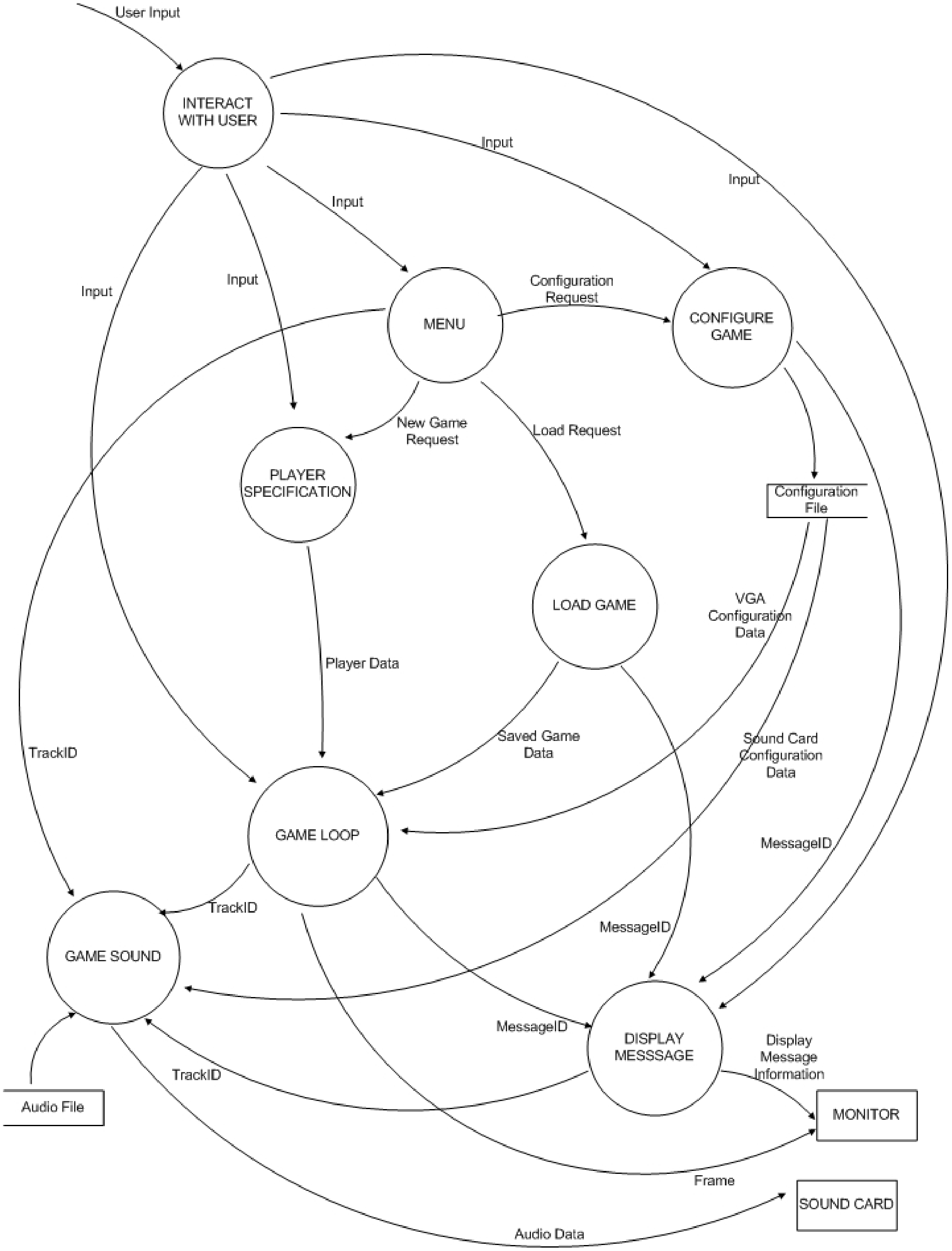


FUNCTIONAL MODEL AND INFORMATION FLOW

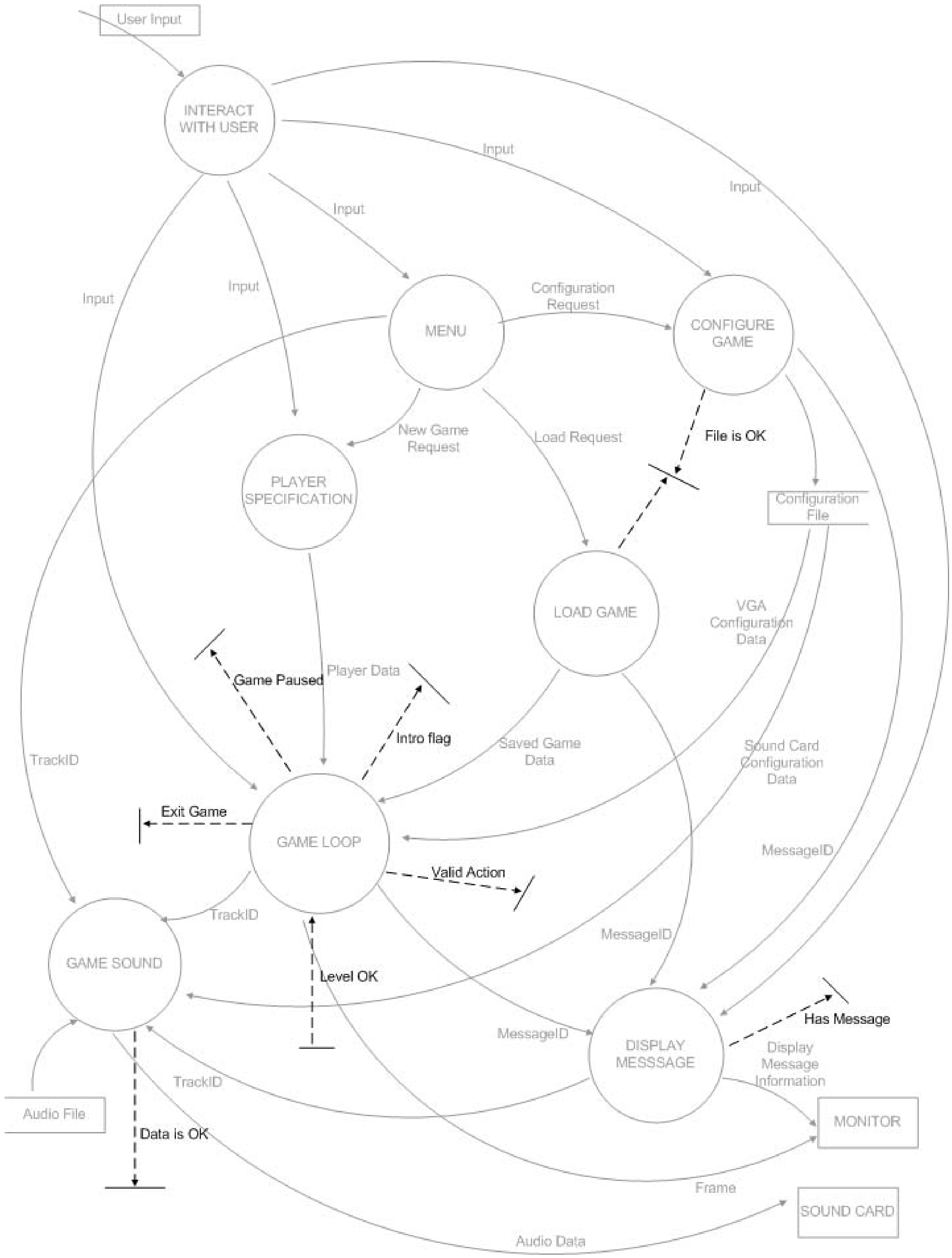
DFD LEVEL 0



DFD LEVEL 1

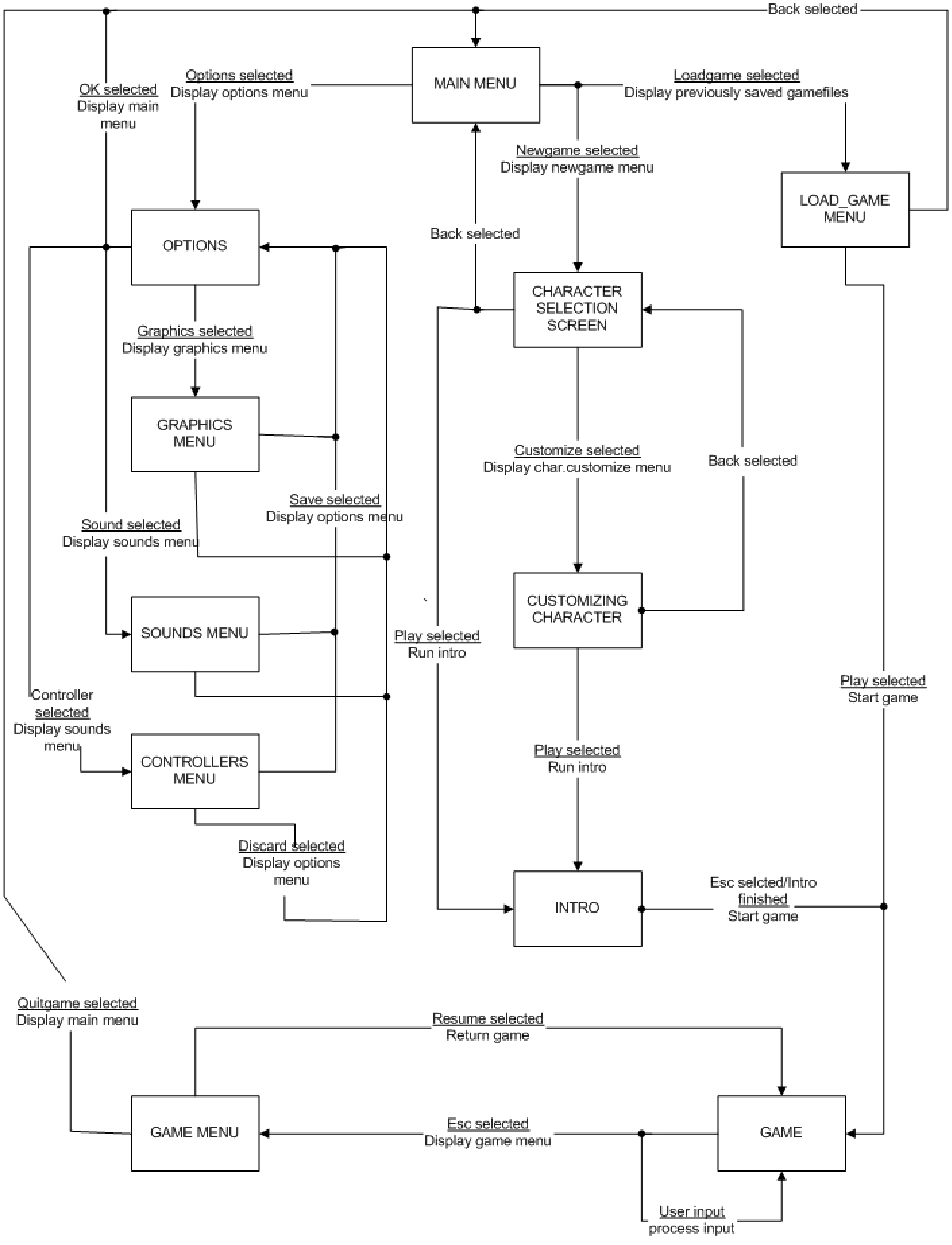

User Input
INTERACT WITH USER
Input
Input
Input
Input
Input
MENU
Configuration Request
CONFIGURE GAME
New Game Request
Load Request
PLAYER SPECIFICATION
Configuration File
LOAD GAME
VGA Configuration Data
Player Data
Saved Game Data
Sound Card Configuration Data
TrackID
GAME LOOP
MessageID
TrackID
GAME SOUND
MessageID
MessageID
DISPLAY MESSSAGE
Display Message Information
TrackID
Audio File
MONITOR
Frame
SOUND CARD
Audio Data

CFD LEVEL 1


User Input
INTERACT WITH USER
Input
Input
Input
Input
Input
MENU
Configuration Request
CONFIGURE GAME
New Game Request
Load Request
File is OK
PLAYER SPECIFICATION
Configuration File
LOAD GAME
VGA Configuration Data
Game Paused
Player Data
Intro flag
Saved Game Data
TrackID
Sound Card Configuration Data
Exit Game
GAME LOOP
MessageID
Valid Action
TrackID
GAME SOUND
MessageID
Level OK
Has Message
MessageID
DISPLAY MESSSAGE
Display Message Information
TrackID
Audio File
MONITOR
Data is OK
Frame
SOUND CARD
Audio Data

BEHAVIORAL MODEL

STD



TEAM ORGANIZATION

TEAM STRUCTURE

The project requires a good planning and a background on the project topics. Therefore, the project requires great and innovative ideas from different minds. Our project group consists of four people as a small sized team. All of us will participate both in designing and implementing phases of the project. Decisions on problems and approaches will be made by group consensus and communication among team members is horizontal which means that there is no any interaction between team members in formal way. Nobody has special dominance over others, so everyone is equal and the team is self-organizing. All of the staff working on the project will have an influence on the project. Because of this kind of reasons, we believe that Democratic Decentralized type of management would be the best.

The management should be decentralized because the project team is very small and the members are not specialized in any kind of duties. Every member is responsible to finish the assigned part on time and discuss it
with others. There are no specific roles like manager, designer, programmer, tester, documenter etc. Also by using the Democratic type, we believe that solving the problems we encounter will be easier.

Another point is that project management will be up to all group members rather than one person. In this way we are planning the make sure that responsibility of the project is shared equally between the members of the project team. However, a leader is useful for holding the group together and organizing the events such as meetings. That's why we chose to have a leader, but this leader does not decide for us, he organizes us. We believe that is fair and sensible since benefits and income of a successful project will be equal for all members.

As a result of "Democratic Decentralized" management approach, decisions and approaches will be taken by group consensus. This will have a fair affect on sharing the responsibilities in an equal manner while there is a leader that organizes us.

PROJECT MANAGEMENT

METHODOLOGY

We have a strict restriction on designing and implementation phases. First we have to fully design the project within three months, and then we have to implement the project within the next four months. As a result of that, we should follow the Waterfall Method principles in order to achieve these tasks. But, it can not be the only suitable approach for the project because the requirements of the project are not well defined, clear and are not strictly stated.
.
Additionally, we want to see the results of our design just after we create it. By doing this, we will be able to control our design specifications that if they are realistic or not. Unfortunately, Waterfall Method does not satisfy our needs completely and therefore we decided to include

Prototyping Method within our plans. We will implement a prototype while designing the project and create a more precise way to our goal.

After the designing phase, we will start coding the program. We will follow object oriented approach in our project. Object oriented approach allows the reusability of the some modules in other parts of the project. Also object-oriented programming provides several concepts and features to make the creation and use of objects easier and more flexible. This will speed up our project. At coding phase, we may think that our prototype has been a very successful one and its codes should be used in the original version. If that happens, we will not want to throw out the code of the prototype and we will build the original program on the prototype. This leads us to the Spiral Method, where the coding part starts from the early stages of the project. Any little failure in the software may result in hazards, identifying the possible errors in the early stages is very important. Since there is a testing phase at each rotation, it will be easy to handle the errors. This model also allows us to show our improvements at each rotation. Therefore, we must include the ideology of the Spiral Method to our methodology either. As a result for our development method, we decided to use a mix of Spiral, Waterfall and Prototyping methods.

MAJOR CONSTRAINTS

Time Constraints

There are so many complications in a complete computer game. We have limited time to complete the project. We have seven months to finish all implementation, documentation, testing, bug fixings and enhancements. Furthermore as senior computer engineering students we have courses except the project. We have so many interesting ideas about our game but we may not implement all of them due to time constraints. So we should manage the time carefully and organize our works.

Performance Constraints

We are trying to develop a computer game, not such a program that people have to use. They will take and play our game only for fun. Therefore our game can be played on a moderate PC easily. Hardware resources of the players’ machines should be used carefully. The game has not to have bugs, flickering screens etc…

Design Constraints

C++ will be used as programming language by letting us to use object oriented concepts. Microsoft’s Visual C++ is used as coding environment. Open GL is the main graphics library. For graphical design part we are be using Photoshop, 3Dmax and Corel.

Lack of Experiece

We have very limited experience in game development that is such a wide area of programming and with the some of the development tools that we will use. So we may have some constraints to implement some functionality.

SOFTWARE SYSTEM ATTRIBUTES

There are a number of attributes that software should provide. For our game as a software project we will try to ensure and qualify these features as well as possible. In order to achieve reliability of our software we will perform a detailed and careful design process. Software reliability errors are usually traced in design or implementation. The best design will surely increase reliability of our software.

Ease of maintenance of software is another factor that makes software qualified. By using a widely known language C++ we guarantee the maintenance Software should be easy to understand. To achieve this we will be commenting during implementation. Therefore, it is possible to say that codes will be readily understandable. We are a team of four people and understandability of each individual's code should be surely provided. Also being an object-oriented product our game should be highly extendible and reusable.

SOFTWARE CONFIGURATION MANAGEMENT PLAN

The main purpose of software configuration management is to identify changes, report and track them. Software configuration management will help us control change without seriously impeding change. The team members will report the changes and notify others when changing something.

Software configuration management activities will be identifying changes, controlling changes, ensuring that change is properly implemented and documenting the change.

As software configuration management organizational role, everyone in the team will do the SCM activities. For instance, if a member of the team makes a change then the others will check to see if it is properly done. So this will ensure that parallel changes made by two different people do not overwrite one another. Also the changes will be documented as the change, why it was done, when, by who, on which level. We will be able to refer to them in the future phases of the project.

First of the task must be done for a change is identification: The person who suggests a change will fill a change request form. Then approving: The team will decide if it is necessary to make the change. The next step is ensuring that change is properly implemented: When a change is made by a member of the team, the others will check the change, make corrections if necessary and finalize it. Last step is documenting the change: As stated before changes will be documented for further use. It will include a change report form (which includes the date, the member's name, the level of change etc.

(Requesting change
-> evaluating change
-> reporting change
-> making final decision on the change)

QUALITY ASSURANCE

Software Quality Assurance mechanism is an important aspect of any project. As a small team, we have no extra person for Quality Assurance of the project but instead all the people will do the testing and control of his/her part of the software as frequent as possible. We will be writing all necessary documentation, including project documentation, technical documents for the team and documents for the users, which includes help files. We will be also performing meetings in the scope of technical reviews frequently to discuss the current position of the project and assigning weekly work to the members of the team and controlling whether each member completed her part.

RISK MANAGEMENT

It is nearly impossible to develop a system which has no defects and errors, but it will be good if we minimize the number of defects or errors in system software. A risk management plan should be formed to develop a strategy to deal with any risk that can be encountered at any stage of the project. To be able to reduce the defect and error rates of the software, there must be a risk management plan to counter any difficulties that may impact the procedure of the project.

The risk management plan consists of the determination of possible risks, managing and monitoring them. The possible risks are: Process Risk (Low quality of product), Employee Risks (Lack of training, ability, willingness and experience), Development Environment Risks (Insufficient resources, availability and quality of tools), Product Size (Misjudgment and miscomputation of product size estimate in LOC or FP), and Technology Risk (Rapidly hanging, obsolete technology).

Since we are inexperienced we may face with risks of employee, process and product size.

PROJECT ESTIMATIONS

FUNCTION POINT ESTIMATIONS

Each of the information domain characteristics like input, output, inquiries, files, external interface and fourteen complexity adjustment values are estimated for our project.

	optimistic	likely	pessimistic	estimated count	weight	FP count
number of inputs	5	7	11	7	4	28
number of outputs	5	7	10	7	7	49
number of inquiries	3	5	8	5	4	20
number of files	3	6	9	6	15	90
number of external interfaces	4	10	15	10	10	100
					Count Total	287

Function point count is adjusted for the complexity of the overall system, by assessing each of the following points on a scale of 0 (no influence) to 5 (essential).

1) Does the system require reliable backup and recovery?	2
2) Are data communications required?	5
3) Are there distributed processing functions?	1
4) Is performance critical?	5
5) Will the system run in an existing, heavily utilized operational environment?	1
6) Does the system require on-line data entry?	0
7) Does the on-line data entry require the input to be built over multiple screens or operations?	0
8) Are the master files updated on-line?	0
9) Are the inputs, outputs, files or inquiries complex?	4
10) Is the internal processing complex?	5
11) Is the code designed to be reusable?	3
12) Are conversion and installation included in the design?	4
13) Is the system designed for multiple installations for different PCs?	1
14) Is the application designed to facilitate change and ease of use by the user?	5
Total	36

FP = count total × (0.65 + 0.01 × total)
= 287 × (0.65 + 0.01 × 36)
≈ **290**

Effort Estimation (Kemerer Model)

$E = 60.62 \times 7.728 \times 10^{-8} \times (FP)^3$
$= 60.62 \times 7.728 \times 10^{-8} \times (290)^3$
≈ 114 person-month

Duration Estimation

D = E / number of project members
= 114 / 4
≈ 28 months

LINES OF CODE ESTIMATIONS

The relationship between LOC and FP depends on the programming language that is used to implement the software. We will use C++ programming language in our project. There exists an estimate of the average number of lines of code required to build one function point in C++, which is 64.Thus;

LOC = FP × 64 for C++
= 290 × 64
= 18560 LOC
≈ 18.5 KLOC

Effort Estimation (Bailey-Basili Model)

$E = 5.5 + 0.73 \times (KLOC)^{1.16}$
$= 5.5 + 0.73 \times (18.5)^{1.16}$
≈ 27 person-month

Duration Estimation

D = E / number of project members
= 27 / 4
≈ 7 months

COCOMO ESTIMATIONS

Project	a^b	b^b	c^b	d^b	a^i	b^i
Organic	2.4	1.05	2.5	0.38	3.2	1.05
semi-detached	3.0	1.12	2.5	0.35	3.0	1.12
embedded	3.6	1.20	2.5	0.32	2.8	1.20

Semi-detached COCOMO model is used in this part of project estimations.

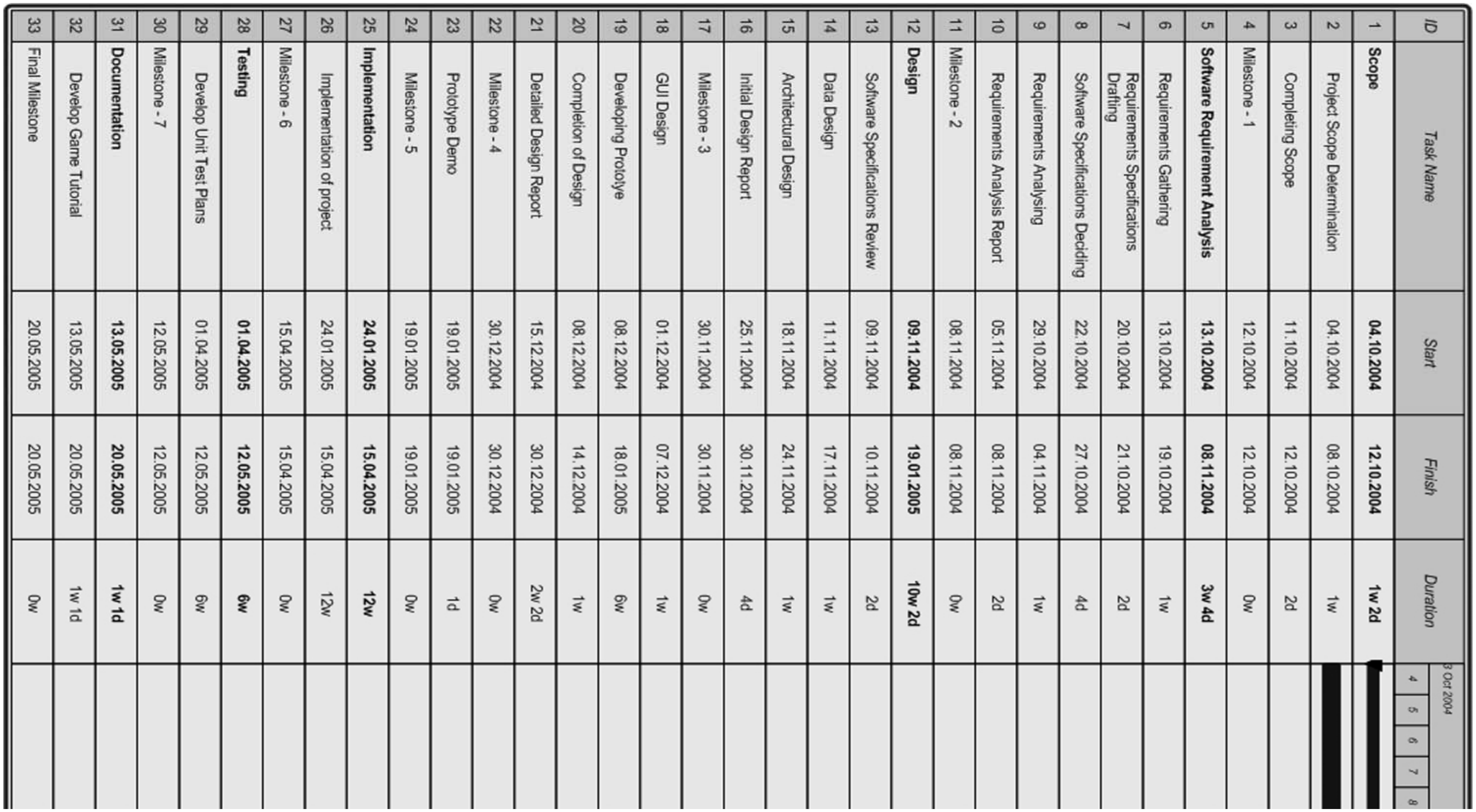
Effort Estimation

$E = a^b \times (KLOC)^{b\hat{}b}$
$= 3.0 \times (18.5)^{1.12}$
≈ 79 person-months

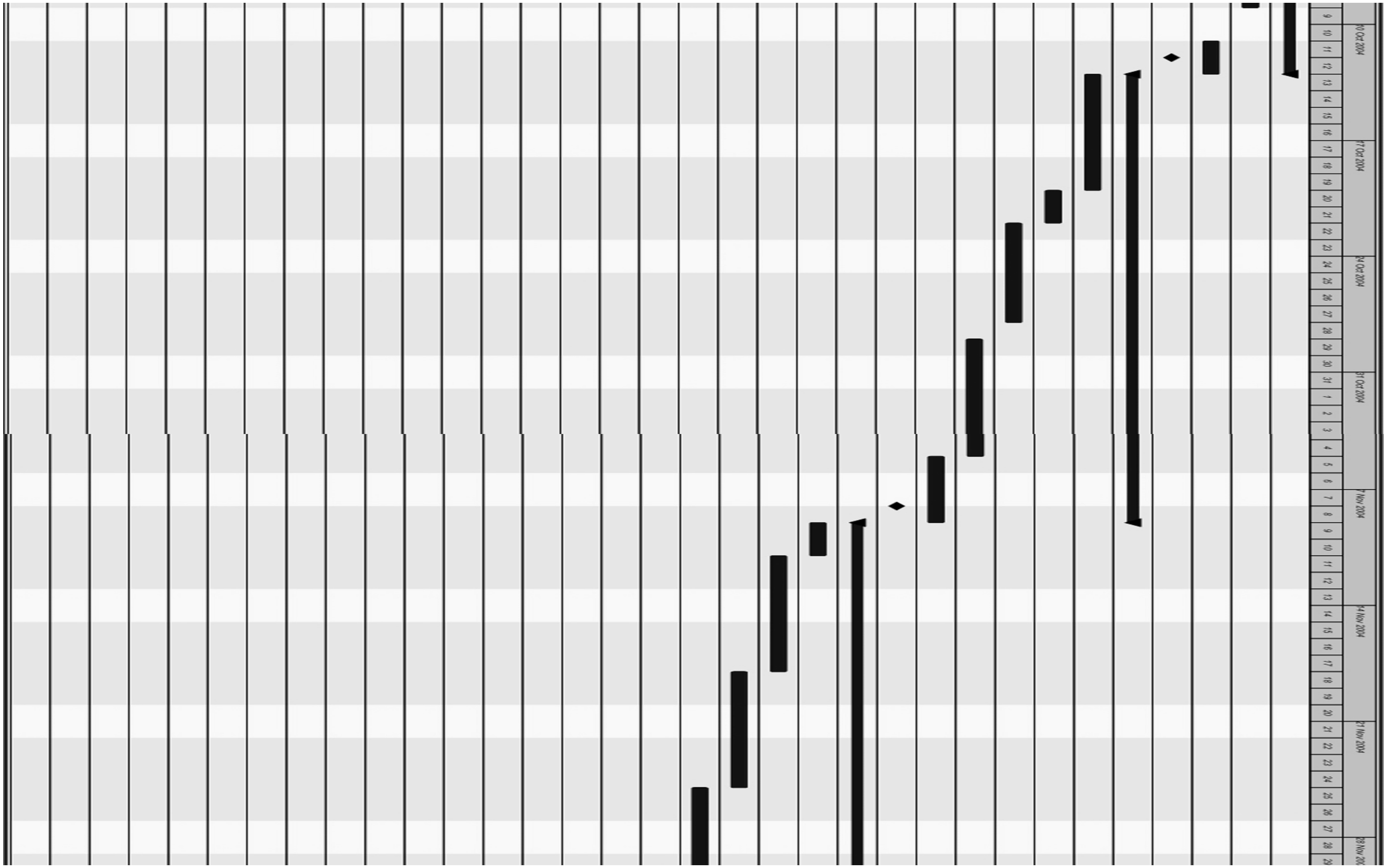
Duration Estimation

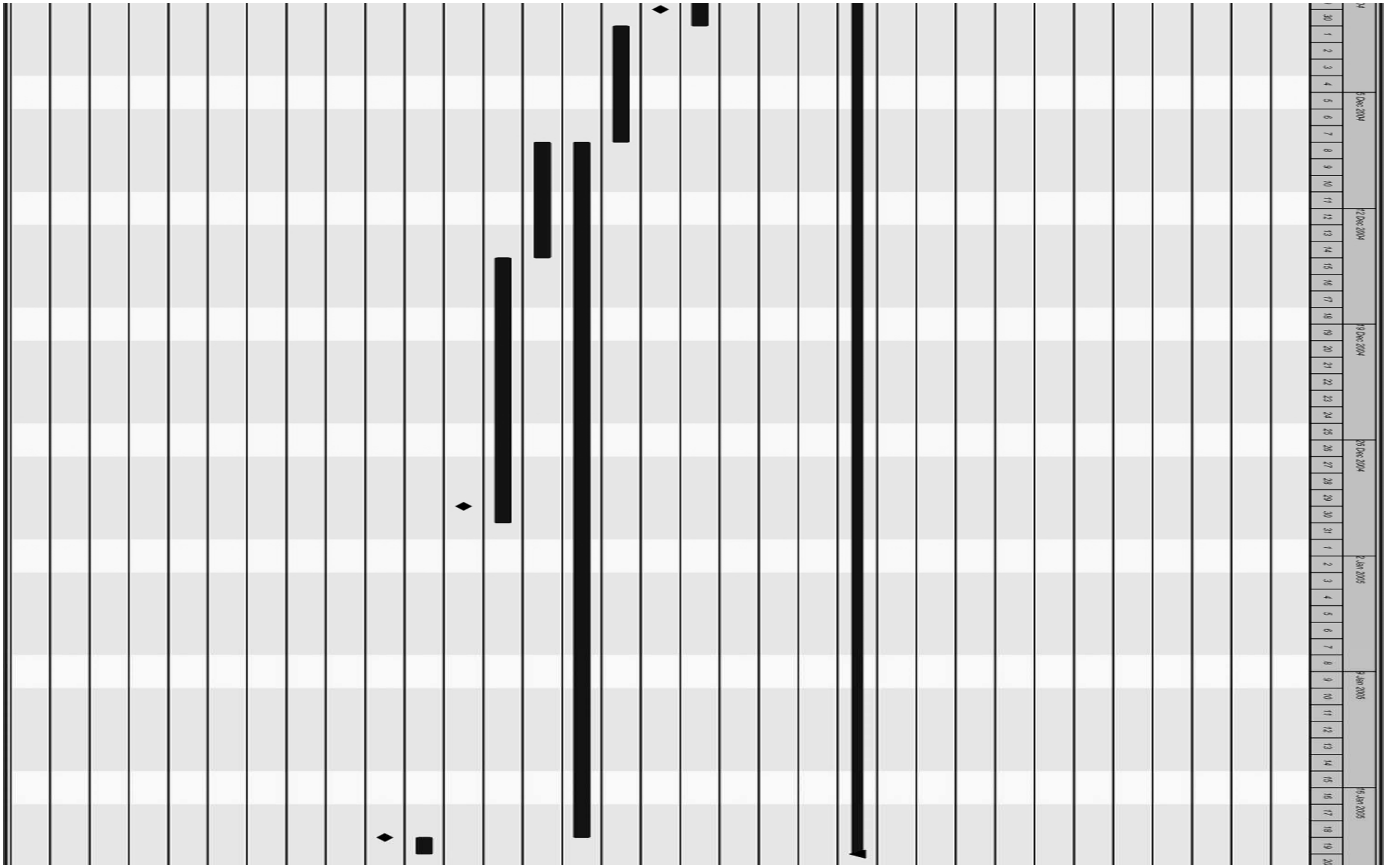
$D = c^b \times (E)^{d\hat{}b}$
$= 2.5 \times (79)^{0.35}$
≈ 11.5 months

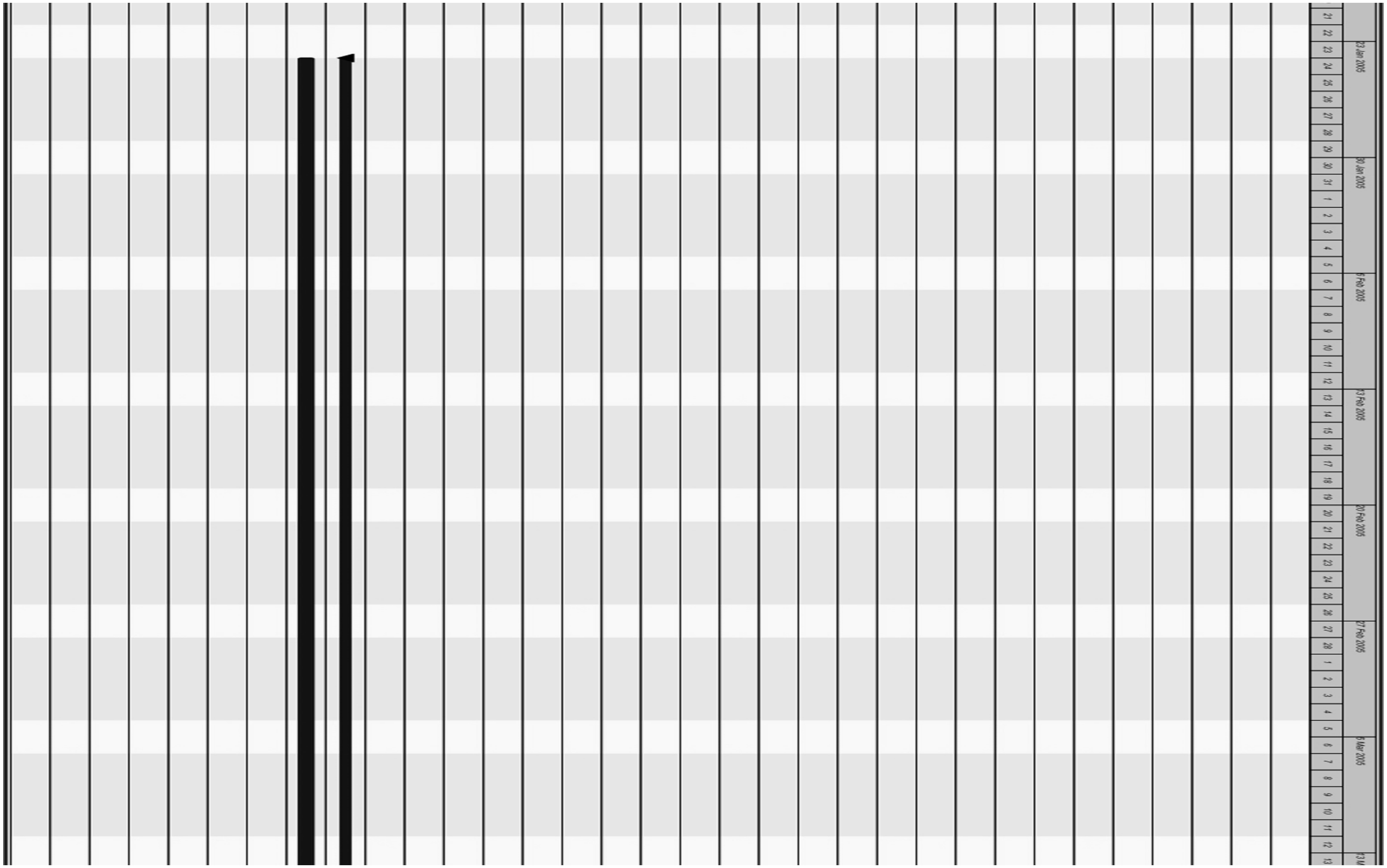
PROJECT SCHEDULE

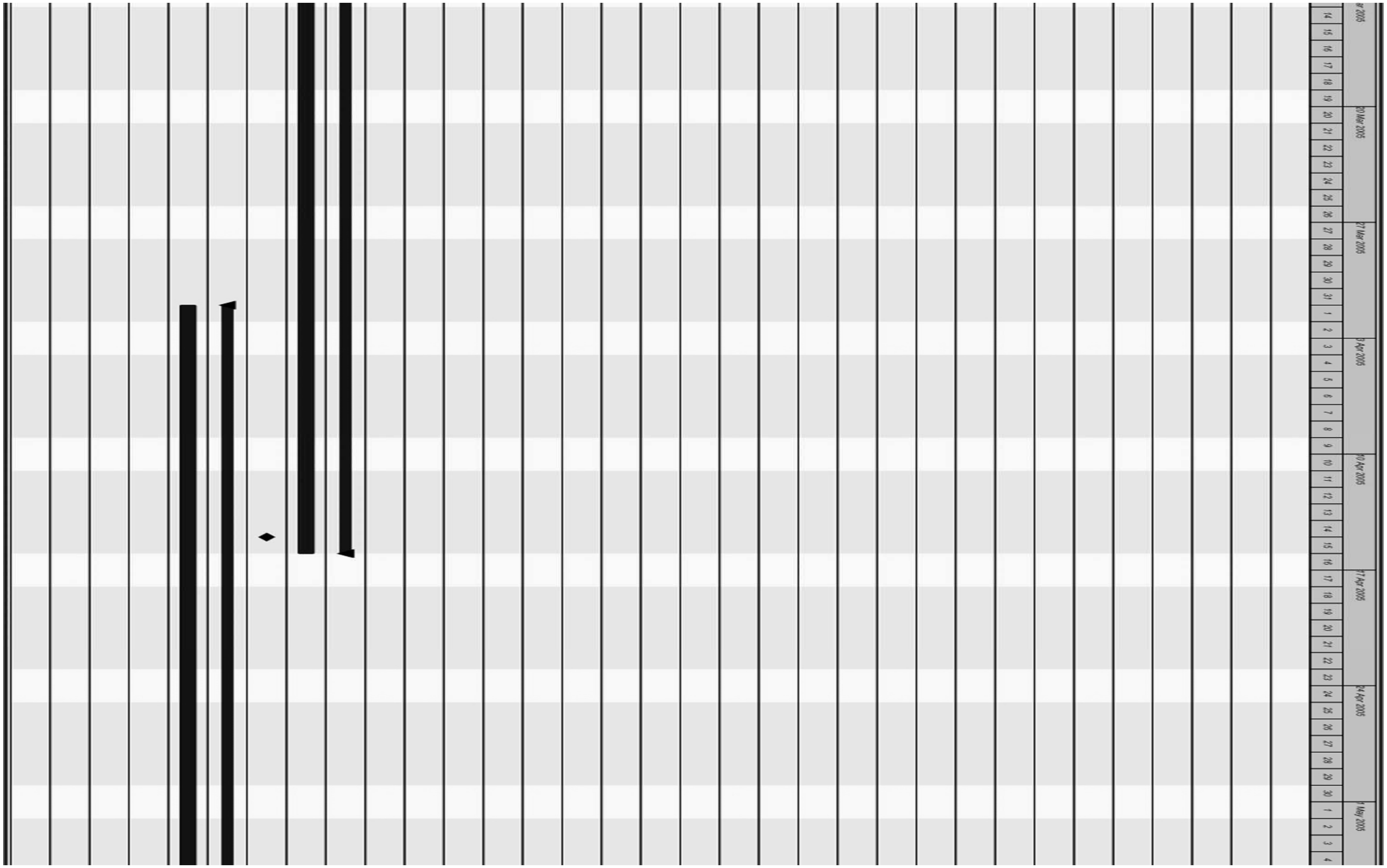


ID	Task Name	Start	Finish	Duration
1	**Scope**	**04.10.2004**	**12.10.2004**	**1w 2d**
2	Project Scope Determination	04.10.2004	08.10.2004	1w
3	Completing Scope	11.10.2004	12.10.2004	2d
4	Milestone - 1	12.10.2004	12.10.2004	0w
5	**Software Requirement Analysis**	**13.10.2004**	**08.11.2004**	**3w 4d**
6	Requirements Gathering	13.10.2004	19.10.2004	1w
7	Requirements Specifications Drafting	20.10.2004	21.10.2004	2d
8	Software Specifications Deciding	22.10.2004	27.10.2004	4d
9	Requirements Analysing	29.10.2004	04.11.2004	1w
10	Requirements Analysis Report	05.11.2004	08.11.2004	2d
11	Milestone - 2	08.11.2004	08.11.2004	0w
12	**Design**	**09.11.2004**	**19.01.2005**	**10w 2d**
13	Software Specifications Review	09.11.2004	10.11.2004	2d
14	Data Design	11.11.2004	17.11.2004	1w
15	Architectural Design	18.11.2004	24.11.2004	1w
16	Initial Design Report	25.11.2004	30.11.2004	4d
17	Milestone - 3	30.11.2004	30.11.2004	0w
18	GUI Design	01.12.2004	07.12.2004	1w
19	Developing Prototye	08.12.2004	18.01.2005	6w
20	Completion of Design	08.12.2004	14.12.2004	1w
21	Detailed Design Report	15.12.2004	30.12.2004	2w 2d
22	Milestone - 4	30.12.2004	30.12.2004	0w
23	Prototype Demo	19.01.2005	19.01.2005	1d
24	Milestone - 5	19.01.2005	19.01.2005	0w
25	**Implementation**	**24.01.2005**	**15.04.2005**	**12w**
26	Implementation of project	24.01.2005	15.04.2005	12w
27	Milestone - 6	15.04.2005	15.04.2005	0w
28	**Testing**	**01.04.2005**	**12.05.2005**	**6w**
29	Develop Unit Test Plans	01.04.2005	12.05.2005	6w
30	Milestone - 7	12.05.2005	12.05.2005	0w
31	**Documentation**	**13.05.2005**	**20.05.2005**	**1w 1d**
32	Develop Game Tutorial	13.05.2005	20.05.2005	1w 1d
33	Final Milestone	20.05.2005	20.05.2005	0w

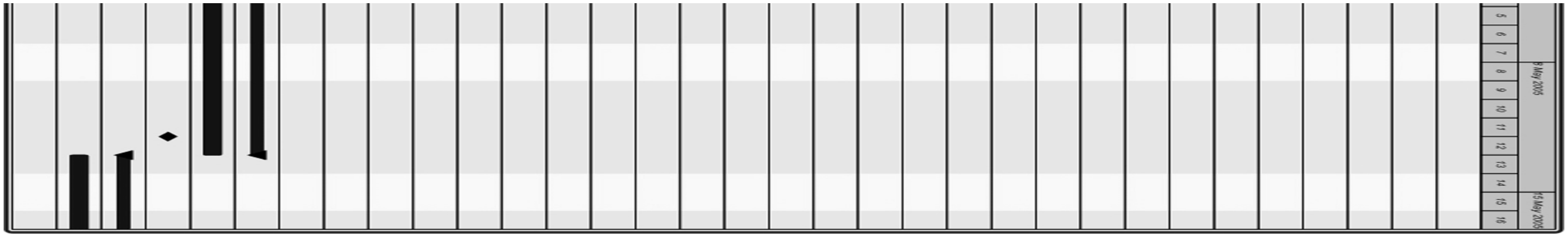

10 Oct 2004
17 Oct 2004
24 Oct 2004
31 Oct 2004
7 Nov 2004
14 Nov 2004
21 Nov 2004
28 Nov 200


5 Dec 2004
12 Dec 2004
19 Dec 2004
26 Dec 2004
2 Jan 2005
9 Jan 2005
16 Jan 2005
30
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20


23 Jan 2005
30 Jan 2005
6 Feb 2005
13 Feb 2005
20 Feb 2005
27 Feb 2005
6 Mar 2005
13 M
21
22
23
24
25
26
27
28
29
30
31
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
1
2
3
4
5
6
7
8
9
10
11
12
13



ar 2005
14
15
16
17
18
19
20 Mar 2005
20
21
22
23
24
25
26
27 Mar 2005
27
28
29
30
31
1
2
3 Apr 2005
3
4
5
6
7
8
9
10 Apr 2005
10
11
12
13
14
15
16
17 Apr 2005
17
18
19
20
21
22
23
24 Apr 2005
24
25
26
27
28
29
30
May 2005
1
2
3
4


5
6
7
8
9
10
11
12
13
14
15
16
8 May 2005
15 May 2005