PROJECT CONTORIUM

CENGBALL

Software Requirements Specification

Alper Demir

Doğa Uzuncukoğlu

Emre Can Küçükoğlu

Necati Çevik

Change History

Date	Revision	Comment
21.10.2013	1.0	Created.
28.12.2013	1.1	Preface added. Figures are updated. Figures table added. Planning section updated.

Preface

This document contains the system design information about CengBall Project. This document is prepared according to the “IEEE Recommended Practice for Software Requirements Specifications – IEEE Std 830 – 1998”.

This document provides a full system description, design and viewpoints of the CengBall. The first section of this document includes scope and purpose of project and gives overall description of it. The following sections include detailed description and requirements of the project.

TABLE OF CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

1.1 Purpose

This document will describe the Player Developed Artificial Intelligence Controlled Football Simulation Platform project and its requirements in a clear and detailed form.

This document is written by the designers and developers of the project and it will satisfy the people who would like to learn about the requirements of this particular one.

1.2 Scope

This project aims to develop the Cengball, for this project the team that is responsible will design and develop;

1. The main simulator which will simulate the game using the player developed AIs and our game's rules.
2. The API for AI developers which will allow players (users) to create AIs compatible with our simulator.
3. Two AIs to play the game so there is a demo for display.
4. The visualizer which will visualize the simulation so the players and others can easily observe the process.

This product will allow multiple players to compete with each other by developing artificial intelligences that play a football-like video game. The desired outcomes of the project are;

1. The players will have a good time while developing AIs because it is a fun game to play.
2. Players will get a chance to try out their AI development skills and probably enhance it.
3. This project might set up the groundwork for an annual event where CEng students and AI developers get together and compete in a tournament using their AIs.

1.3 Definitions

The terms and words that might not be understood at first sight in this document are;

AI	Artificial Intelligence
API	Application Programming Interface
Video Game	A game which uses the help of algorithms and computer generated graphics.
OS	Operating System

C.Eng	Computer Engineering
METU	Middle East Technical University

1.4 Overview

The rest of the document will explain the project from various productive perspectives, will talk about the product functions and user characteristics.

2 OVERALL DESCRIPTION

2.1 Productive Perspective

This game will be a new self-contained programming based video game. It will not require any other product. In the project, both the platform and the bot AIs will be developed from the scratch.

2.2.1 System Interfaces

System interfaces and how a user interacts with them can be shown in the Figure 1.

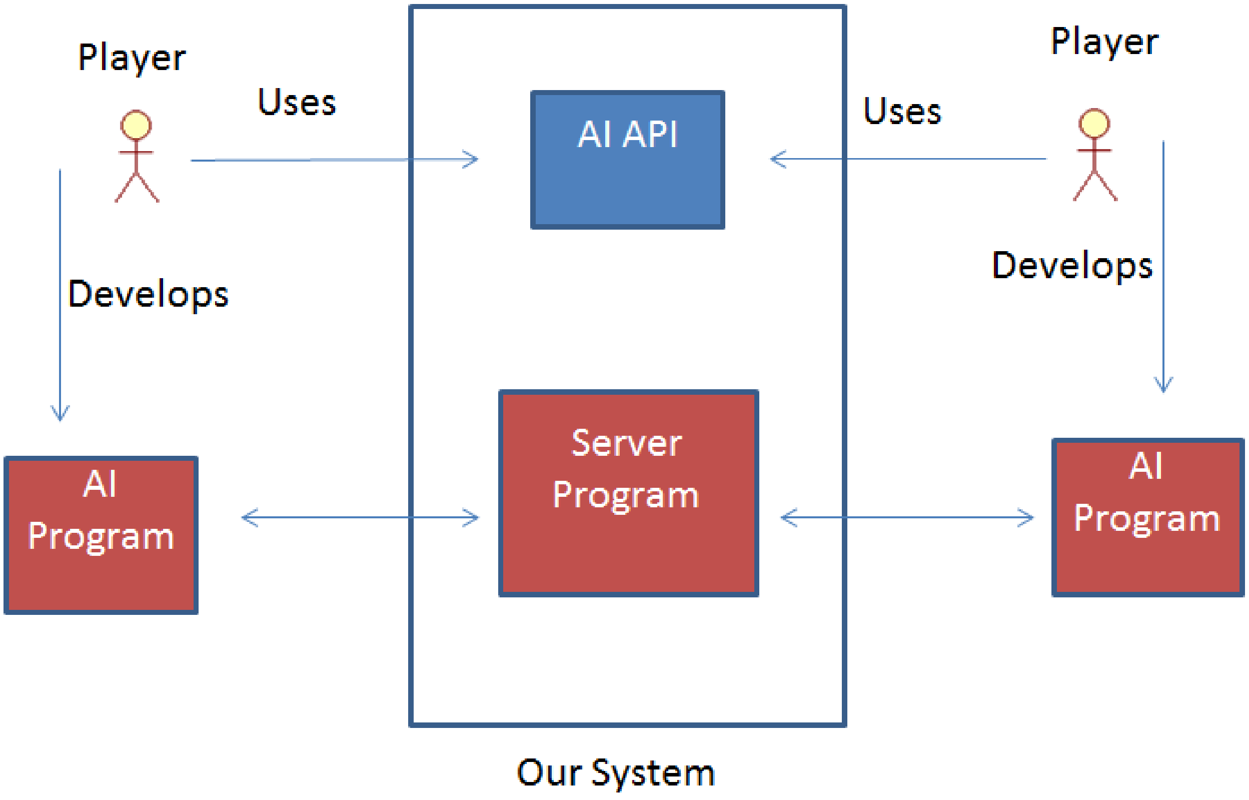


Figure 1: System Interface

2.1.2 User Interfaces

While the full system is a video game, since the way to play the game differs from a standard game, the user- system interactions are quite different from a regular video game.

First, users will program an AI according to designated rules and they will use the AI API. AI API will allow them to have full control over the game as if the users themselves were playing.

Later, the simulator interface of the system will simulate the game according to user's AI. This part of the system does not contain any user-system interactions since the simulator only interacts with the AIs and the visualizer.

Finally, the visualizer interface of the system will visualize the simulation using the input from the simulator. This part of the system, again does not require any specific user-system interaction.

2.1.3 Hardware Interfaces

Since the system does not have any high level hardware requirements, it will be able to run on any C.Eng department inek machine which is a monitor and a standard computer system.

2.1.4 Software Interfaces

Only software requirement the system has is an operating system and it can run a standard OS such as Ubuntu or Microsoft Windows 7.

2.1.5 Communications Interfaces

Communicating parts of the system can be seen in Figure 1.

2.1.6 Memory Constraints

System does not deal with any amount of large data so there are not any constraints on memory.

2.1.7 Operations

After the user develops the AI, his interactions and operations with the system will be minimal (Watching the simulation).

2.2 Product Functions

The platform will;

- Simulate the game with the rules and submitted codes
- Display the game in visual way
- Have at least three AI bots as examples
- Let the participants to submit their codes to it
- Let the participants to have interfaces and functions to develop their codes

2.3 User Characteristics

Target users to this platform are programmers. They are expected to have experience on programming and to be interested in computer science and artificial intelligence area. They should be able to write a computer program.

2.4 Constraints

While the system does not have any restricting hardware or software constraints, the system have some constraints on the user which are described in the 2.3 User Characteristics part of this document.

2.5 Assumptions and dependencies

This project does not have any assumptions or dependencies.

3 SPECIFIC REQUIREMENTS

3.1 External Interfaces

- Main Screen : This screen includes mode selection and exit operation. After mode is selected, relative screen will be shown.
- Single Player Screen : This screen makes it possible to submit the code of the user. Also, it is possible to select the opponent AI difficulty and start the game.
- Two Player Screen : This screen makes it possible to submit the codes of the both users and start the game.
- Match Screen : This screen is driven by visualizer. It is used to display the match.

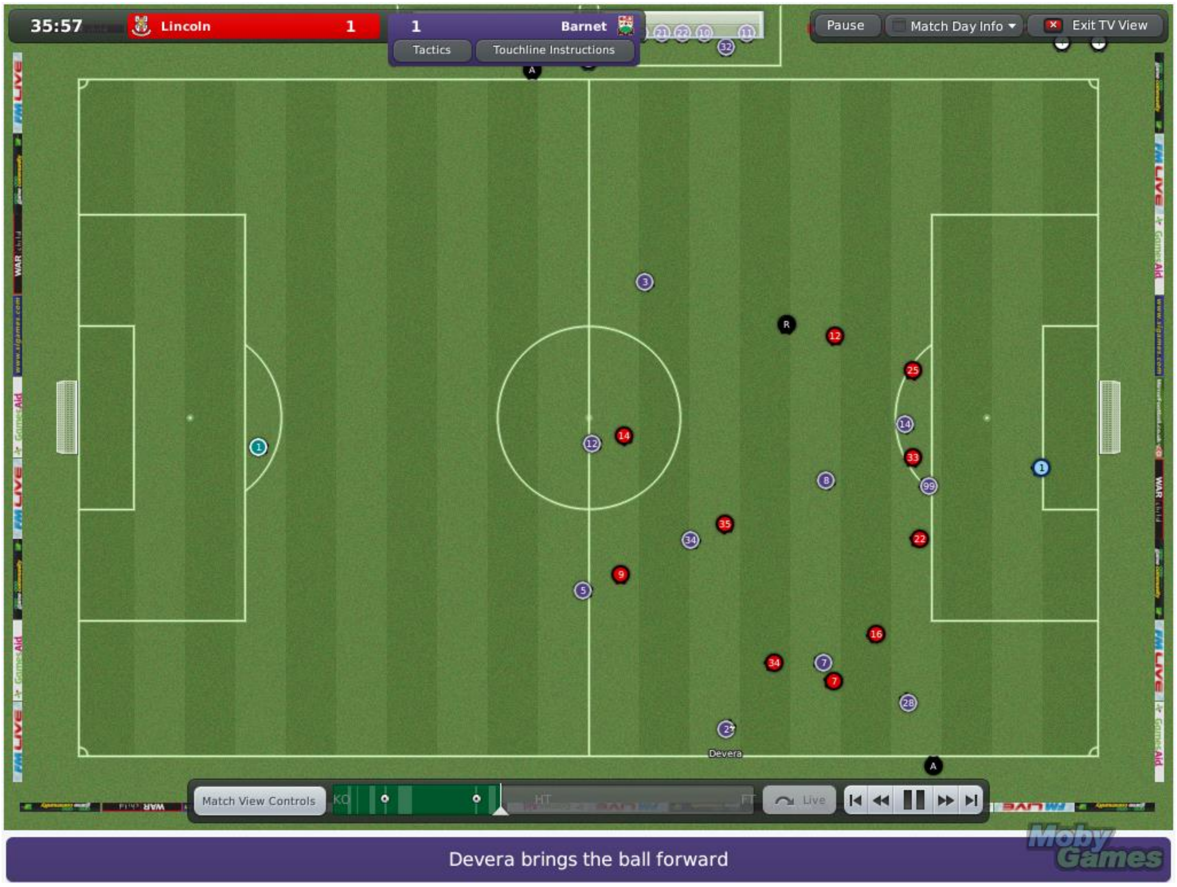
3.2 Functions

3.2.1 API Functions

AI API will allow the players to have full control over the field of play but of course their power will be rationed according to the rules of the game. Functionalities of the API are;

- Players (Developers) will be able to get location data of all virtual football players and the ball.
- Players will be able to do real life football functions such as shooting the ball and passing the ball. Those functions will be determined at design phase.
- Players will not be able to control opponent virtual football players.

3.2.2. Visualizer Functions



Devera brings the ball forward

Figure 2: Visualizer Example

The visualizer is responsible for getting the move information and displaying them on the 2D pitch. The display will look like the Figure 2. Also, the visualizer will have a commentary panel. The panel will be updated according to the events of the match. The visualizer;

- Initializes the pitch, the ball and the players
- Reads the replay file in periodic time
- Updates the display and shows the movements so that they will look smooth to human eye

3.2.3. Simulator Functions

Simulator will simulate the game according to well defined rules. It will communicate with both the submitted codes and the visualizer. The main tasks of the simulator are;

- Running the codes of the players by calling the must functions in periodic time
- Checking the moves of the players for whether their moves are valid
- Updating the movements of the players and writing them to a replay file
- Reporting the errors in the simulation
- Finishing the simulation

3.2.4. System Functions

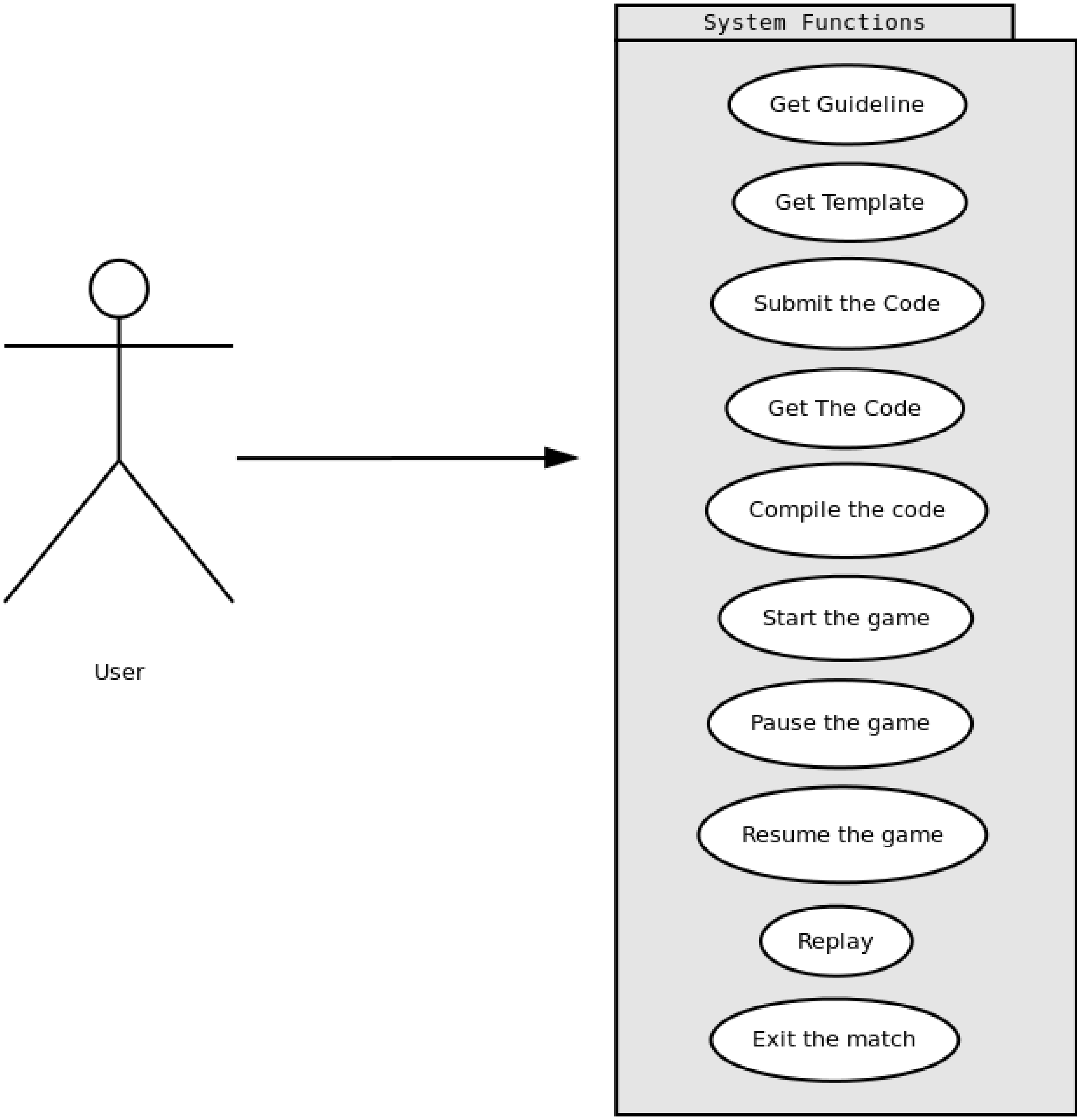


Figure 3: System Functions

3.2.4.1. Get Guideline Use Case

Use Case ID	use_case_1
Use Case Name	Get Guideline
Trigger	The user selects to get the guideline
Precondition	-
Basic Path	1. The user selects to download the guideline to a directory in the computer 2. The user selects the directory to download in browse screen. 3. The user confirms the directory. 4. The game downloads the guideline to the directory.
Alternative Paths	-
Postcondition	The guideline is downloaded to the selected directory.
Exception Paths	The user may cancel downloading.
Other	The guideline includes the API information. It has definition of functions, variables and classes. Also, it has the information about how the game works and how a submitted code should be.

3.2.4.2. Get Template Use Case

Use Case ID	use_case_2
Use Case Name	Get Template
Trigger	The user selects to get the template

Precondition	-
Basic Path	1. The user selects to download the template to a directory in the computer 2. The user selects the directory to download in browse screen. 3. The user confirms the directory. 4. The game downloads the template to the directory.
Alternative Paths	-
Postcondition	The template is downloaded to the selected directory.
Exception Paths	The user may cancel downloading.
Other	The template is an example code for a player AI. It includes the must functions for the game to be played. Also, it has information about the functions in comments.

3.2.4.3. Submit the Code Use Case

Use Case ID	use_case_3
Use Case Name	Submit the code
Trigger	The user selects to submit the code written
Precondition	-
Basic Path	1. The user selects to submit the code to the game. 2. The user selects the directory of the code. 3. The user confirms the directory. 4. The game gets the code.

Alternative Paths	-
Postcondition	The code is submitted to the game.
Exception Paths	The user may cancel submitting.
Other	-

3.2.4.4 Get the Code Use Case

Use Case ID	use_case_4
Use Case Name	Get The Code
Trigger	The user selects to get the code of his/her
Precondition	The code must be submitted to the game.
Basic Path	1. The user selects to download the code to a directory in the computer 2. The user selects the directory to download in browse screen. 3. The user confirms the directory. 4. The game downloads the code to the directory.
Alternative Paths	-
Postcondition	The code is downloaded to the selected directory.
Exception Paths	The user may cancel downloading.
Other	-

3.2.4.5 Compile the Code Use Case

Use Case ID	use_case_5
Use Case Name	Compile the code
Trigger	The user selects to compile the code submitted
Precondition	The code must be submitted to the game.
Basic Path	1. The user selects to compile the code. 2. The game compiles the code, checks whether it obeys the rules of the game and returns back the result of compile operation.
Alternative Paths	-
Postcondition	The code is compiled, checked and the result of the compile operation is returned.
Exception Paths	-
Other	-

3.2.4.6 Start the Game Use Case

Use Case ID	use_case_6
Use Case Name	Start the game
Trigger	The user selects to start the match.
Precondition	In single player mode, the code of the user must be submitted to the game. In two player mode, the codes of the both users must be submitted to the game.

Basic Path	1. The user selects to start the match. 2. The game opens a new screen and starts to simulate and visualize the match.
Alternative Paths	-
Post condition	The match is started and getting visualized.
Exception Paths	-
Other	

3.2.4.7 Pause the Game Use Case

Use Case ID	use_case_7
Use Case Name	Pause the game
Trigger	The user selects to pause the game.
Precondition	The match must be started and being played.
Basic Path	1. The user selects to pause the match. 2. The game freezes and pauses the match. Also, a pause screen will be shown.
Alternative Paths	-
Postcondition	The match is paused.
Exception Paths	-
Other	The pause panel will have resume, replay and exit option.

3.2.4.8 Resume the Game Use Case

Use Case ID	use_case_8
Use Case Name	Resume the game
Trigger	The user selects to resume the game.
Precondition	The match must be paused.
Basic Path	1. The user selects to resume the match. 2. The game continues to simulate and the visualize the match.
Alternative Paths	-
Postcondition	The match is resumed.
Exception Paths	-
Other	-

3.2.4.9 Replay Use Case

Use Case ID	use_case_9
Use Case Name	Replay
Trigger	The user selects to replay.
Precondition	The match must be paused.
Basic Path	1. The user selects to replay a scene. 2. The replay screen opens. 3. The user selects to play the scene.

	4. The game plays the scene.
Alternative Paths	-
Post condition	The scene is played.
Exception Paths	The user may exit the replay screen.
Other	-

3.2.4.10 Exit the Game Use Case

Use Case ID	use_case_10
Use Case Name	Exit the match
Trigger	The user selects to exit the match.
Precondition	The match must be paused.
Basic Path	1. The user selects to exit the match. 2. The game shows a popup screen to confirm exit operation. 3. The user confirms the operation. 4. The game finishes the simulation, closes the match screen and returns to the main screen.
Alternative Paths	In step 3, the user cancels the exit operation and the game returns back to the pause screen.
Post condition	The match is exited.
Exception Paths	-
Other	-

3.3 Performance Requirements

The system shall support two participants (players) at the same time. It will use their codes in parallel to simulate the game.

It shall visualize the game in a frequency to be seen normal to human eye.

It shall run the game by calling the both codes and updating the game in parallel. By means of parallel, it shall interact to both players at the same time.

3.4 Logical Database Requirements

In the project, there won't be any database to place any information.

3.5 Design Constraints

Since the system can run on a regular computer and OS, there isn't any design constraints.

3.5.1 Standard Compliance

There is no existing standard for this project to have derived requirements.

3.6 Software System Attributes

Related issues are addressed in their respective subclauses.

3.6.1 Reliability

Since the system is basically a large executable, for the system to be reliable it only requires a regular computer and an standard operating system such as Ubuntu or Windows 7.

3.6.2 Availability

System will analyze the user written AIs for any compile errors. If the AIs compile, the system will run without any errors until the simulation ends.

3.6.3 Security

Since the platform is not open for public access there is no special security requirements.

3.6.4 Maintainability

Since the system has a clear set of interactions, the system will have a modular set of functionalities. This modular parts will ease the maintenance. In case that some AI irregularities happen, the development team will be able to react to the situation at real time.

3.6.5 Portability

System will be able to run on any inek machines at Computer Engineering department at METU. The system is not dependent on any operating system dependent nor any high technology hardware (Department inek machines are considered intermediate computer systems).

3.7 Organizing the Specific Requirements

3.7.1 System Mode

The game will have two modes such as single player and two player. These modes will be able to be selected in the main screen.

In single player mode, the user will play with a built-in AI player. There will be 3 different selection for this player. Those selections will be ordered by their difficulties as easy, medium and hard. In this mode, the user will be able to test his/her player against the players that we developed.

In two player mode, there will be two users. These users will submit their code to the game and play against each other.

3.6.2 User Class

The users of this system can be divided into two such as players and viewers. Players are the ones that develop their AIs and attend the game. On the other hand, a user can be only a viewer. The one can just watch the game playing.

3.6.3 Objects

The system does not interact with any real-world entities rather than a computer.

3.6.4 Feature

The system does not have any other feature but simulating and displaying the game.

3.6.5 Response

Simulator will interact with both AIs and the visualizers and these interactions will result in responses. Simulator will ask for input from the AIs and AIs will respond. Simulator will simulate the game then send the results to visualizer. Visualizer will respond by drawing the new object locations on the screen.

4 DATA MODEL AND DESCRIPTION

System has three main parts which communicates with each other so the data flow can be described using those parts. Those parts are the simulator, visualizer and the AI clients.

4.1 Data Description

AI clients will have a function which will be called periodically by the simulator so the communication between the simulator and clients will be exactly like a function call.

Simulator and the visualizer will communicate via a replay file. Simulator will save the simulated actions on this replay file and then the visualizer will draw the described events.

4.1.1 Data Objects

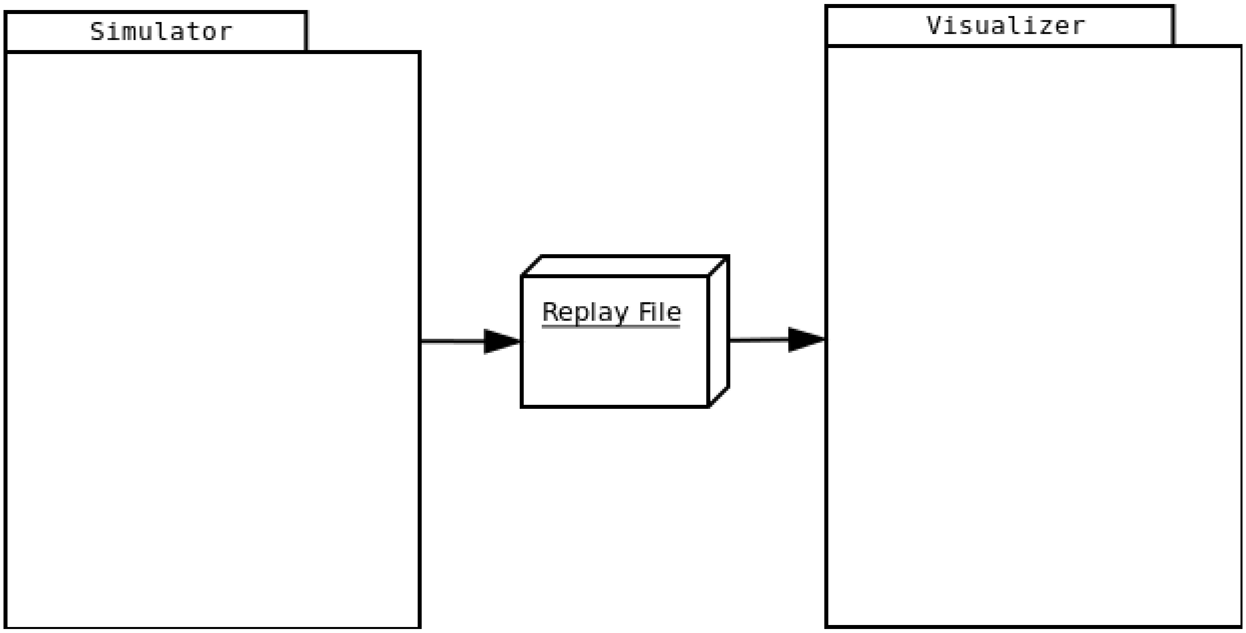


Figure 4: Data Objects

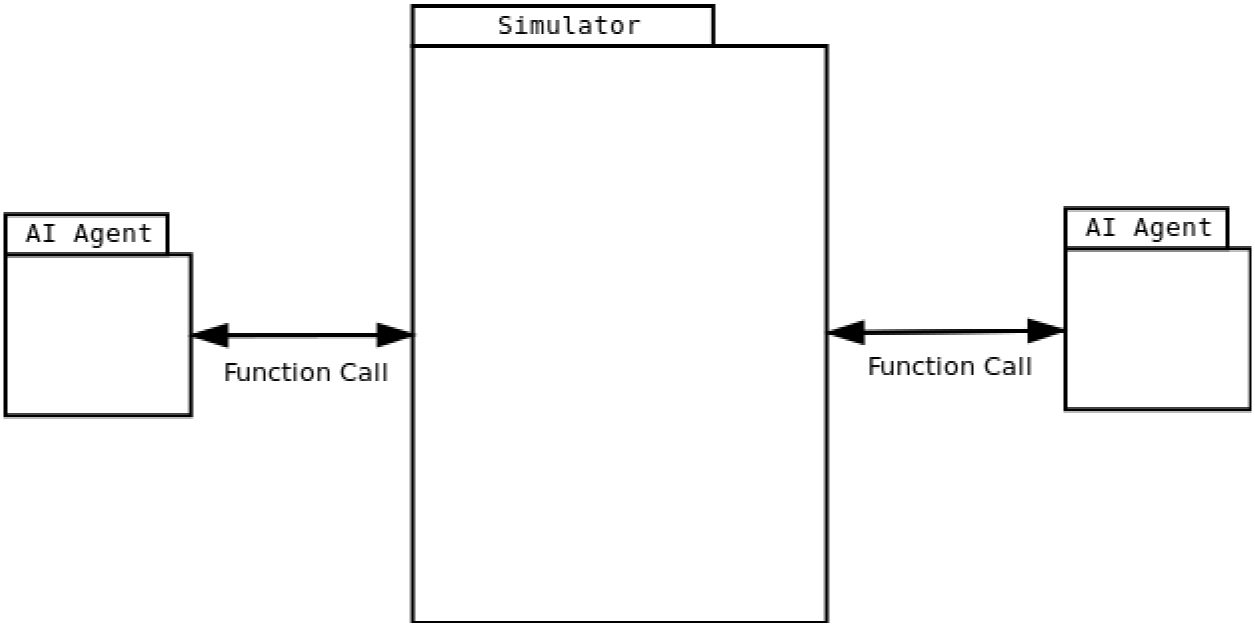


Figure 5: Data Objects

4.1.2 Data Dictionary

Replay File: It is a file for logging the movements in the game. The simulator will write the movements to this file and the visualizer will read from this file. Then, the visualizer will display the movements according to the file.

5 BEHAVIORAL MODEL AND DESCRIPTION

5.1 Description for Software Behavior

Pre submission	The system will wait for a submission.
Precompile	The system will check the compliance of submitted AIs'.
Ready	The system will be in ready position to start or exit.
Playing	The system will simulate the game.
Paused	The system will pause the game.
Replaying	The system will visualize the game according to replay files.

5.2 State Transition Diagrams

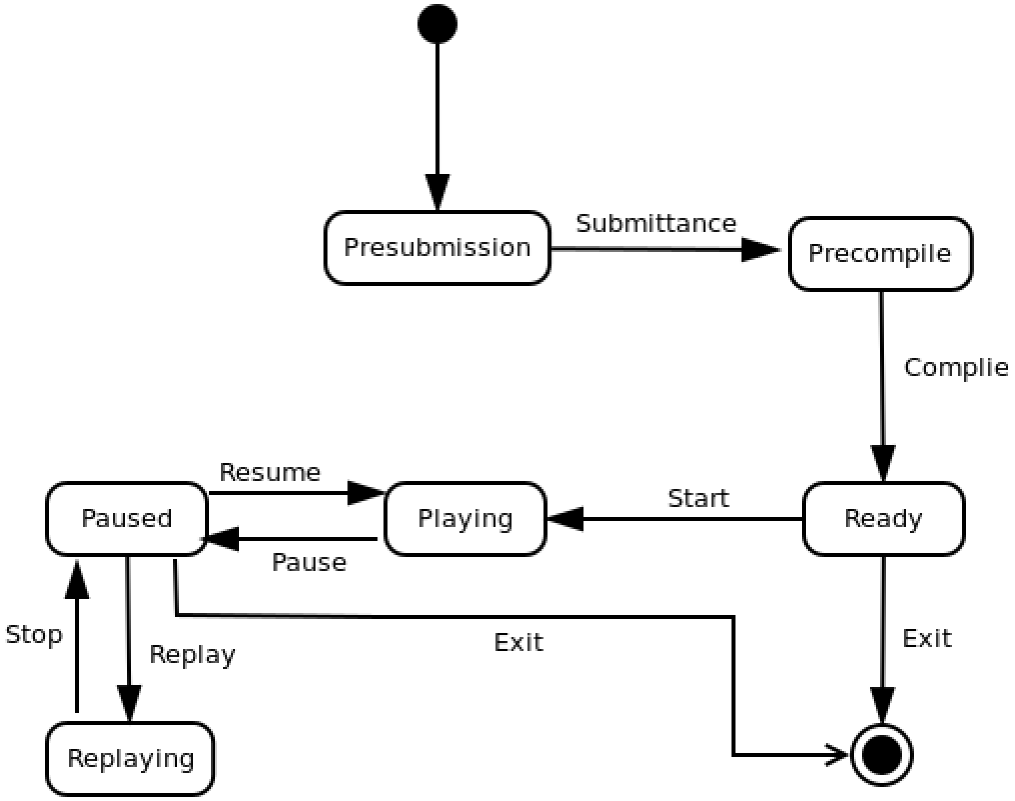


Figure 6: State Transition Diagram

6 PLANNING

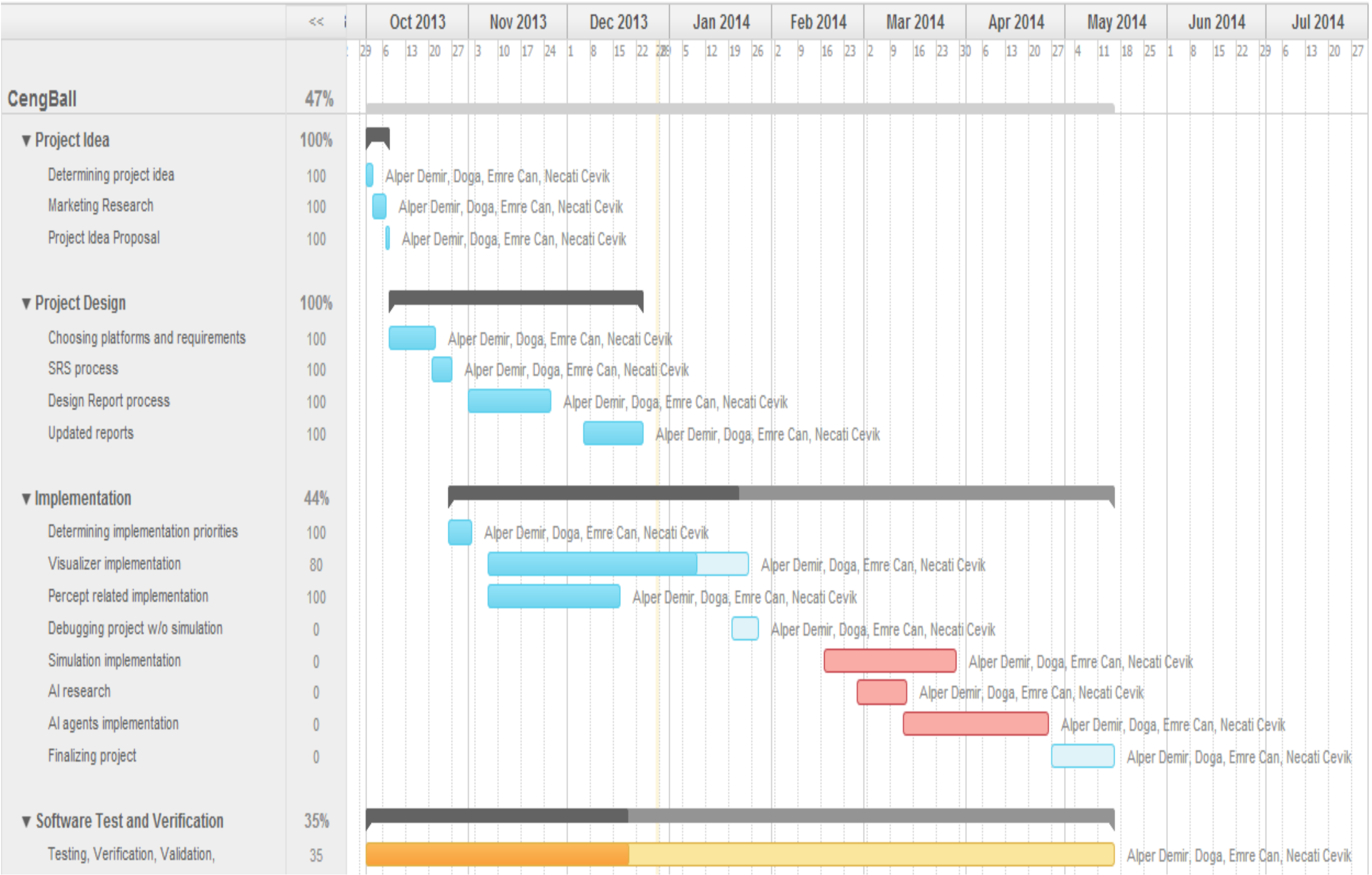
6.1 Team Structure

The development team consists of four senior computer engineering students, all members are working as both researcher and developer.

- Alper Demir
- Doga Uzuncukoglu
- Emre Can Kucukoglu
- Necati Cevik

Asst. Prof. Selim Temizer is advisor of CengBall project.

6.2 Estimation



CengBall	47%	
▾ Project Idea	100%	
Determining project idea	100	Alper Demir, Doga, Emre Can, Necati Cevik
Marketing Research	100	Alper Demir, Doga, Emre Can, Necati Cevik
Project Idea Proposal	100	Alper Demir, Doga, Emre Can, Necati Cevik
▾ Project Design	100%	
Choosing platforms and requirements	100	Alper Demir, Doga, Emre Can, Necati Cevik
SRS process	100	Alper Demir, Doga, Emre Can, Necati Cevik
Design Report process	100	Alper Demir, Doga, Emre Can, Necati Cevik
Updated reports	100	Alper Demir, Doga, Emre Can, Necati Cevik
▾ Implementation	44%	
Determining implementation priorities	100	Alper Demir, Doga, Emre Can, Necati Cevik
Visualizer implementation	80	Alper Demir, Doga, Emre Can, Necati Cevik
Percept related implementation	100	Alper Demir, Doga, Emre Can, Necati Cevik
Debugging project w/o simulation	0	Alper Demir, Doga, Emre Can, Necati Cevik
Simulation implementation	0	Alper Demir, Doga, Emre Can, Necati Cevik
AI research	0	Alper Demir, Doga, Emre Can, Necati Cevik
AI agents implementation	0	Alper Demir, Doga, Emre Can, Necati Cevik
Finalizing project	0	Alper Demir, Doga, Emre Can, Necati Cevik
▾ Software Test and Verification	35%	
Testing, Verification, Validation,	35	Alper Demir, Doga, Emre Can, Necati Cevik

Figure 7: Gantt Chart

6.3 Process Model

The process model will be spiral model. In each spiral, we will define our objectives and try to accomplish them. After each spiral, we will develop a prototype.

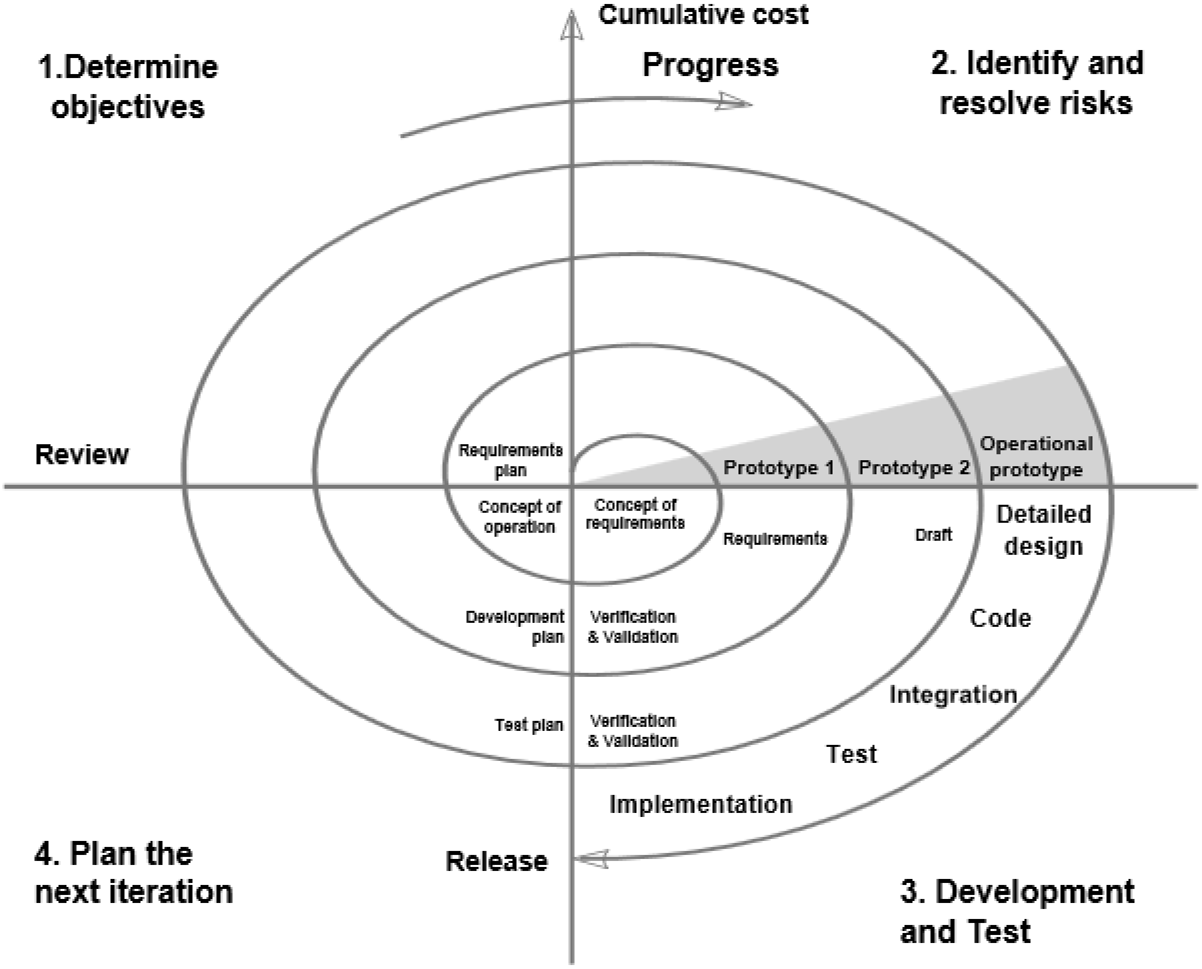


Figure 8: Process Model

7 CONCLUSION

This project aims to allow AI developers to compete on a fun and easy to learn platform. The development team is quite ambitious to create such a platform. While the end product will not be a huge market success, it will be unique.