Intelligent Ludo

Prepared by:

Mahnoor Hamid

Supervised by:

Ms. Ifrah Farukh Khan

QAU, Computer Science Department

(2013 – 2017)

In the name of Allah Almighty, the most merciful and beneficent

A report submitted to the Department of computer science, Quaid-i-Azam University, Islamabad, as a partial fulfillment of the requirement for the award of the degree of Bachelor of Computer Science

Acknowledgement

First of all I would like to express my gratitude to Allah Almighty, whose blessings I have seen during my time at Quaid-e-Azam University in the form of achievements, success and learning. I thank my parents and family for their prayers and being concerned about my education all my life and for supporting it.

Among the teachers, I would like to thanks all the teachers specially thank to Ms. Ifrah Farukh Khan for her supervision and guidance over this project. I am thankful for her precious time, invaluable criticism and friendly advice during this project.

I am also greatly thankful to all my friends for keeping me company and encouraged during this time. I greatly appreciate their caring attitude towards my work.

Thank you everyone!

Mahnoor Hamid.

Abstract

Game development is a particular challenge within the field of software engineering. Specific knowledge of Mathematics calculations is necessary to achieve a satisfactory result.

The main purpose of making this game is to learn and implement AI. The idea behind the development of this game is to entertain Android users. In today's time, Unity3D is popular for game development. So I learned it and use it for my project. The challenge was that unity3d uses 3d objects in a 3d space and you cannot control them from inside some code like a main class, rather you have to attach scripts or code to these 3d objects and control their behavior by using them. The real complexity of this project is AI.

Intelligent Ludo is a four-player turn based game, in which there at least one Android player (AI) and other Human Players. Each player has set of four tokens. The game is done playing by rolling a dice on ludo board. The player races all its four token to finishing point first, wins the game. AI, behind the Android player, make its move among best choice with the help of algorithms. The game will be three-dimensional which gives feel like real Ludo environment.

Overall, it was a good experience to be able to implement such a project by myself.

List of Tables

List of Figures

Table of Contents

Chapter 1:

Project Management Plan

1.1 Introduction SPMP

This document specifies the project management plan for Intelligent Ludo game. It specifies the process of Software development cycle and deliverables such as Software Requirement Specification, Software Design Description and Software Test Documentation.

1.2 Project Overview

Ludo is a famous four-player turn based game played on a ludo board. It consist of a set of four token and a dice. Each player has four tokens and they are differentiated by four different colors (red, blue, yellow and green). The game is done playing by rolling the dice. The player races all of his four tokens to the finishing point first, wins the game.
This project plan is for making the game Android based which gives user a complete real Ludo environment. In this game, user will be able to play with Android player as well. Android Player plays game automatically with the help of AI evaluations. It supports portability, so that anyone can play this game anywhere on a hand held device instead of carrying a board. During play, player can pause the game at any time.

1.3 Project Deliverables

Project deliverables for this game are:

1. Software Project Management Plan (SPMP) document.
2. Software Requirement Specifications (SRS) document.
3. Software Design Description (SDD) document
4. Software Test Documentation (STD) document.

1.4 Project Organization

1.4.1 Software Process Model

Waterfall process model will be used for the development of this project. This model is preferred because it is simple, easy and reasonable approach when requirements are well understood. It is a straight forward model which means there is no risk and no change in requirements for this project. Each phase of this model has specific deliverables and a review process.

1.4.2 Roles and Responsibilities

There is no division of roles and responsibilities. All roles and responsibilities of analysis, design, implementation and testing is done by Mahnoor Hamid.

1.4.3 Tools and Techniques

Following are the tools and techniques used for this project.

Table 1. 1 Tools and Techniques

Tools and Techniques	
MS-Word	It is used for documentation.
StarUML	It is used for making Use case diagram, System Sequence diagrams, Domain models and Class diagrams.
Unity 3D (v5.6)	A game development platform used for the game objects.
ProjectLibre	It is used to make time table for project plan distributing tasks over the working months.
Photoshop	It is used for making editing images.
MonoDevelop	Used for coding of game.

1.5 Project Management Plan

1.5.1 Tasks

Two phases of project plan are define below, Analysis and Design and Implementation. In Analysis phase, the major tasks are Identify Requirements, develop SRS, define use cases, develop analysis model, and review SRS.
In Design phase, the major tasks are develop design, develop algorithm and evaluate design.
Following figure shows tasks:

1.5.1.1 Analysis:

i. **Description:**

In this phase, Problem statement is defined, case study is reviewed, functional/non-functional requirements are identified and use cases are derived.

ii. **Deliverables and Milestones**

For Analysis phase, SPMP document and SRS document are delivered. Milestones are shown in figure.

iii. **Resources Needed**

Other than project developer and supervisor, resources for this phase are Computer, Softwares (MS-Word, Usecase tool) and Internet Connection.

iv. **Dependencies and Constraints**

All dependencies required for these tasks are listed in the Analysis figure.

v. **Risks and Contingencies**

None.

	Name	Duration	Start	Finish	Predecessors	Resource Names
1	**Analysis**	46 days	**05/10/16 08:00**	**08/12/16 08:00**		
2	**Identify Requirements**	10 days	**05/10/16 08:00**	**18/10/16 17:00**		
3	Problem Definition	2 days	05/10/16 08:00	06/10/16 17:00		Mahnoor Hamid
4	Review Case Study	3 days	07/10/16 08:00	11/10/16 17:00	3	Mahnoor Hamid
5	Define Requirements	5 days	12/10/16 08:00	18/10/16 17:00	4	
6	**Develop SRS**	21 days	**17/10/16 08:00**	**14/11/16 17:00**		
7	Define Functional Requirements	5 days	19/10/16 08:00	25/10/16 17:00	5	Mahnoor Hamid
8	Review Functional Requirements	2 days	26/10/16 08:00	27/10/16 17:00		Mahnoor Hamid;Supervisor
9	**Define Usecases**	8 days	**28/10/16 08:00**	**08/11/16 17:00**		
10	Identify Usecases	2 days	28/10/16 08:00	31/10/16 17:00	7	Mahnoor Hamid
11	Write Usecase Description	3 days	01/11/16 08:00	03/11/16 17:00	10	MS-Word;Mahnoor Hamid;PC
12	Draw Usecase Diagram	2 days	04/11/16 08:00	07/11/16 17:00	11	Mahnoor Hamid;PC;UML Tool
13	Review Usecases	1 day	08/11/16 08:00	08/11/16 17:00		Mahnoor Hamid;Supervisor
14	Define Non Functional Requirements	4 days	09/11/16 08:00	14/11/16 17:00		
15	Review Non Functional Requirements	1 day	17/10/16 08:00	17/10/16 17:00		
16	Review SRS	2 days	18/11/16 08:00	21/11/16 17:00	6	Mahnoor Hamid;Supervisor
17	Submit SPMP and SRS	1 day	22/11/16 08:00	22/11/16 17:00		
18	**Develop Analysis Model**	6 days	**23/11/16 08:00**	**30/11/16 17:00**		
19	Develop System Sequence Diagram	2 days	23/11/16 08:00	24/11/16 17:00	11	Mahnoor Hamid;PC;UML Tool
20	Review System Sequence Diagram	1 day	25/11/16 08:00	25/11/16 17:00		Mahnoor Hamid;Supervisor
21	Develop Domain Model	2 days	28/11/16 08:00	29/11/16 17:00	11	Mahnoor Hamid;PC;UML Tool
22	Review Domain Model	1 day	30/11/16 08:00	30/11/16 17:00		Mahnoor Hamid;Supervisor
23	**Finalize SRS**	3 days	**05/12/16 08:00**	**07/12/16 17:00**		
24	Review SRS	3 days	05/12/16 08:00	07/12/16 17:00		Mahnoor Hamid;Supervisor
25	Analysis Phase Done	0 days	08/12/16 08:00	08/12/16 08:00	2	

Figure 1. 1 Project Planning Time Table

1.5.1.2 Design

i. Description:

After Analysis phase, Design phase is done. In this phase, Usecase diagram is developed from use cases then system sequence diagram, domain model and class diagram is created.

ii. Deliverables and Milestones

For Analysis phase, SDD document and STD document are delivered. Milestones are shown in figure.

iii. Resources Needed

Other than project developer and supervisor, resources for this phase are Computer, Softwares (MS-Word, UML tool, designing tool, design table creator) and Internet Connection.

iv. Dependencies and Constraints

All dependencies required for these tasks are listed in the Design figure.

v. Risks and Contingencies

None.

26		**Design**	29 days	**12/12/16 08:00**	**19/01/17 17:00**		
27		**Develop Design**	8 days	**12/12/16 08:00**	**21/12/16 17:00**		
28		**Architectual Design**	4 days	**12/12/16 08:00**	**15/12/16 17:00**		
29		Develop Architectural Design	2 days	12/12/16 08:00	13/12/16 17:00	25	Designing Tool;Mahnoor Hami...
30		Review Architectural Design	2 days	14/12/16 08:00	15/12/16 17:00		Mahnoor Hamid;Supervisor
31		**Interface Design**	4 days	**16/12/16 08:00**	**21/12/16 17:00**		
32		Develop Interface Design	2 days	16/12/16 08:00	19/12/16 17:00	29	Designing Tool;Mahnoor Hami...
33		Review Interface Design	2 days	20/12/16 08:00	21/12/16 17:00		Mahnoor Hamid;Supervisor
34		Create Sequence Diagram	2 days	22/12/16 08:00	23/12/16 17:00	19	Mahnoor Hamid;PC;UML Tool
35		Create Design Class Diagram	2 days	26/12/16 08:00	27/12/16 17:00	34	Mahnoor Hamid;PC;UML Tool
36		**Develop Algorithms**	9 days	**29/12/16 08:00**	**10/01/17 17:00**		
37		Draw Flow Chart	2 days	29/12/16 08:00	30/12/16 17:00	27	Mahnoor Hamid;PC;UML Tool
38		Write Pseudo Code	3 days	02/01/17 08:00	04/01/17 17:00	37	MS-Word;Mahnoor Hamid;PC
39		Review Pseudo Code	1 day	05/01/17 08:00	05/01/17 17:00		Mahnoor Hamid;Supervisor
40		Draw Decision Table	2 days	06/01/17 08:00	09/01/17 17:00	38	Decision Table Creator;Mahn...
41		Review Decision Table	1 day	10/01/17 08:00	10/01/17 17:00		Mahnoor Hamid;Supervisor
42		**Evaluate Design**	6 days	**11/01/17 08:00**	**18/01/17 17:00**		
43		Validate Design	2 days	11/01/17 08:00	12/01/17 17:00	27	Mahnoor Hamid;Supervisor
44		Verify Design	2 days	13/01/17 08:00	16/01/17 17:00	27	Mahnoor Hamid;Supervisor
45		Review and Refine Design	2 days	17/01/17 08:00	18/01/17 17:00		
46		Design Phase Completed	0 days	18/01/17 08:00	18/01/17 08:00		
47		Submit SDD and STD	1 day	19/01/17 08:00	19/01/17 17:00		

Figure 1. 2 Project Planning Time Table 2

Chapter 2:

Software Requirements Specification

2.1 Introduction SRS

The following document will provide you the complete Software Requirement Specification (SRS) for the game project "Intelligent Ludo". In this chapter, purpose, scope and product overview is given for the understanding of this project

2.1.1 Purpose

The purpose of this SRS is intended to give a complete overview of making the game "Intelligent Ludo". It describes the functional and non-functional requirements of our system. It helps us for better understanding of the system, Intelligent Ludo. It is intended for developer, designer, supervisor and other relevant audience. Its current version is Intelligent Ludo version 1.0.

2.1.2 Scope

The scope of the project is to develop an Android based game, Intelligent Ludo. It consists of a board with tokens and die on it. It includes four players in which there is at least one human player and rest Android players. Android player plays game with the help of Artificial Intelligence. Each player has four tokens. It is a turn based game in which game is done playing by rolling a die.

It supports portability, so that user can play this game anywhere on a hand held device instead of carrying a board. User can give Touch input and receive output on the Screen.

It is stand-alone game. It also provides instructions to let user know about the rules of the game.

2.1.3 Project Overview

It is an Android based game. The current Ludo game is played manually and needed at least 2-4 players at a time. It cannot be played alone. It is less portable because game requires certain area with stability so that it can be played with comfort. Game is based on some rules which are sometimes follows and sometime not. As our lives turned very fast and such games are left back to lockers at home and enjoyed by dust, it cannot be played with speed and young people do not have much time and interest in it, so we are making it Android based that will help young people to enjoy it. It also supports portability.

The goal of the game is to provide a facility to play against Android in a user friendly environment. It represents a traditional and 2D environment of Android Ludo Game. It consists of a board having four columns, representing each of the four player's starting square. The game includes four players, human (at least one) and Android. There are set of four tokens for each player of different colors (red, green, yellow and blue) and a pair of dice to play this game. In this game, every players have to move their four tokens from start to finish according to dice rolls. It is a turn based game in which one player takes his turn while other player waits. It follows all the rules of traditional game. These rules are as follows:

- Initially all tokens are in Home (Starting Square). Players take turns in a clockwise/anticlockwise order; highest throw (6) of the dice.
- The player decides which token is to move at each throw. A token simply moves around the track given by the number thrown. If no piece can legally move according to the number thrown, play passes to the next player.
- A throw of 6 gives another turn but cancels the turn if a player gets four 6 on the dice.
- If a player's token lands on the opposite player's token, it captures or beat that token. The opposing player's token moves back to its home and the beater player gets bonus turn.
- If a player's token reaches its finishing point, he gets bonus turn.
- When a token has navigated the board, it proceeds up the finishing path.
- The first person to move all 4 tokens to the finishing point, wins the game.

Now, we will implement these traditional rules for an Android based game application so that anyone can play this game with Android as well. Player can interact with system by giving input which system responds and displays output. Another purpose for making this game is to handle cheating. Algorithms are used for basic game play and decision making.

The game includes menus interface having some options. Backgrounds are used to make it more attractive. During play, player can pause the game anytime.

2.1.4 AI

Today, games and intelligence go side by side. Ludo is an intelligently played mind game. It requires some basic knowledge for different moves that need to be taken on other player's move. Artificial intelligence defines the behavior of non-player characters. It is a technique that have similar features as human being acting as artificial intelligent agents.

Intelligent Ludo provides environment for users, playing with artificial agents. Its main purpose lies in the comparison of Risks and Rewards. At one side, Ludo is a very simple game played with full observation, while on other side it contains few challenges due to its multi-agent environment. Therefore it offers a good balance between Risk and Reward (simplicity and complexity) that is able to attract a wide audience, not only professionals.

In this project, different choices are made on the basis of level (number of face values) in a turn. AI player plays on the basis of Positive and Negative Utility. Negative Utility (Risk) shows that the token is danger to move and get killed by opponent's token if it moves while Positive Utility (Reward) shows that the token is in safe place and there is no risk to move forward. Utility (Ranks) are given to each possible move, for example new position, to kill token, to take token out of home, risk of being killed and to win the token. Risk have low rank which means that the token has less possibility to move forward. It is calculated by checking number of enemies in specific range, behind the token, at its current position. Rewards have higher rank which means

that there is more possibility in moving forward. The rank of Reward increased which depends on how the token is far from home. The token near its finishing point is more valuable than the token near its home. If the new position of the token has less danger, rank would increase otherwise it decrease.

2.2 Specific Requirements

Specific Requirements includes External Interface Requirements, System features, System Quality attributes and Database Requirement.

2.2.1 External Interface Requirements

External interface requirements includes user interface, software and hardware interface.

2.2.1.1 User Interface

As a gamers and game developers, we know that immersion is everything. A game becomes more attractive due to its presentation, i-e how well designed its UI is. So, for this game:

- A visual GUI is used to display the menu and sub menus.
- Buttons are displayed on menus and sub menus for selecting options.
- Backgrounds are used make our interface more attractive.
- A Canvas is used to display the game environment, just like a common Ludo board.
- As player make moves, the screen is updated to reflect the moves made in the game.

2.2.1.2 Hardware Interface

Since the game is based on Android, so it supports any Android System, meeting the following criteria:

- At least 1GB ram and 100MB memory storage to install and load the game application.
- Touch screen input for selecting options from menus and play the game.
- A WIFI connection to download and install the game.

2.2.1.3 Software Interface

The game runs only on Android Operating System and it is implemented on Unity3D v5.6, which supports C# scripting language. This development tool is preferred because it is a best game developing engine. It is portable and cross platform which means that the same code, developed via Unity engine, can be ported on many platforms with minimal modifications.

2.2.1.4 Communications Protocol

No Communication Protocol is required for this game.

2.2.2 Software Product Features

The following tables summarizes the use cases of the system in which User can interact with system by giving some input and system displays output as a response:

- To displays the main menu at the start of the game having background and some options like ‘Start Game’, ‘Instructions’ and ‘Exit’.
- To show the animation of dice.
- To control the movement of tokens.
- To change and manages the state of the game.
- To process the player’s command to the game environment.
- To play the game turn by turn by the player. (At least one human and Android)

2.2.2.1 Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

Given below is the use case diagram of the system

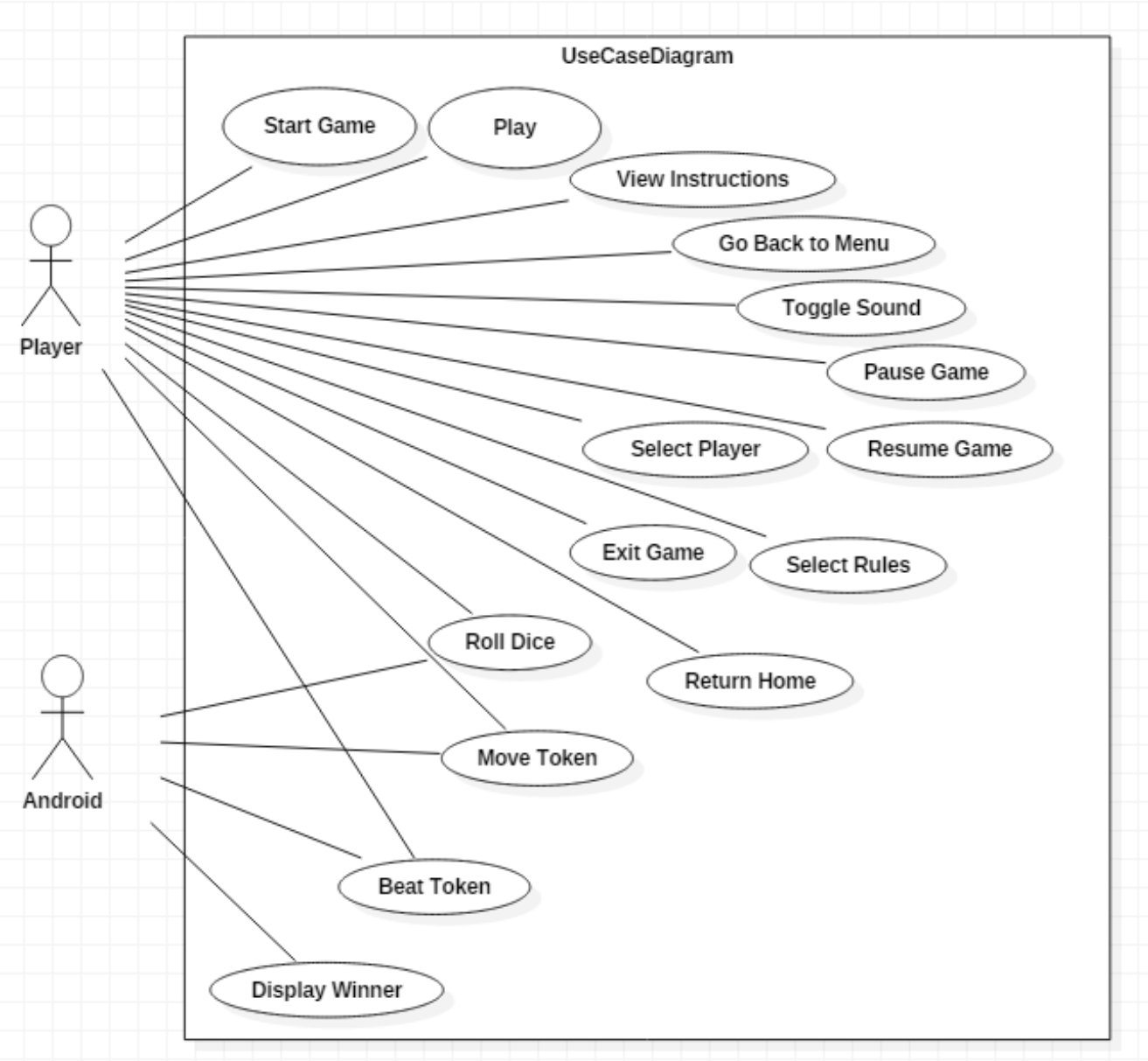

Figure 2. 1 Use Case Diagram'

2.2.2.2 Use Case Description

Table 2. 1 Use Case 1

UC-1 name: Start Game	
Primary actor	Player
Stakeholder & Interests	Player will be able access the Main Menu.
Pre-condition	1. System should support the Game configurations. 2. Player opens Game app
Post-condition	Start Menu has been opened successfully.
Main Success Scenario	1. System loads Main Menu. 2. Main Menu displays. 3. Player selects Start option 4. System loads Start Menu 5. Start Menu displays
Alternate Flows	2a. System cannot load Main Menu. 5a. System cannot load Start Menu.
Special Requirements	None.
Technology	Touch input. LCD display for output.
Frequency	Multiple times.

Table 2. 2 Use Case 2

UC-2 name: Exit Game	
Primary actor	Player.
Stakeholder & Interests	Player will be able to Exit the game.
Pre-condition	Main Menu should be opened.
Post-condition	Game has been exit to mobile desktop successfully.
Main Success Scenario	1. Player selects the Exit option. 2. System closes the application. 3. Game exits.
Alternate Flows	2a. System does not closed the application, system failure.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Multiple times.

Table 2. 3 Use Case 3

UC-3 name: Select Player	
Primary actor	Player
Stakeholder & Interests	Player will be able to select single player or multiplayer game with Android.
Pre-condition	Start Game Menu should be opened.
Post-condition	Number of Player(s) has been selected successfully.
Main Success Scenario	1. Player taps at least one of the four options below "Select Player". 2. Player selects either "Human" or "None". 3. User changes option from default player to his/her choice. 4. System updates Player's choice.
Alternate Flows	3a. Default player is fixed. 4a. System displays error if Player selects no Human Player.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Multiple times.

Table 2. 4 Use Case 4

UC-4 name: View Instructions	
Primary actor	Player
Stakeholder & Interests	Player will be able to view instructions to play according to game rules.
Pre-condition	1. Main Menu should be opened. 2. Instructions should be defined.
Post-condition	Instruction has been viewed successfully.
Main Success Scenario	1. Player selects Instructions option. 2. System loads instructions to the screen. 3. Instructions displays.
Alternate Flows	2a. Instruction does not displays
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Few times.

Table 2. 5 Use Case 5

UC-5 name: Select Rules	
Primary actor	Player
Stakeholder & Interests	Player will be able to select Game Rules of his/her choice.
Pre-condition	1. Start Menu should be opened. 2. Game supports Game rules
Post-condition	Rules have been selected successfully.
Main Success Scenario	1. Player selects the Rules given below "Select Rules". 2. Player updates default option to his choice. 3. System updates Player's Choice.
Alternate Flows	3a. Player's choice does not update.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Multiple times.

Table 2. 6 Use Case 6

UC-6 name: Play	
Primary actor	Player
Stakeholder & Interests	Player will be able to play the game
Pre-condition	1. Start Menu should be opened.
Post-condition	Game has been started successfully.
Main Success Scenario	1. Player selects Player's Option. 2. Player selects Rules. 3. Player selects "Play" option. 4. System loads required game data into the memory. 5. Game starts.
Alternate Flows	4a. System cannot loads game data, system failure.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Multiple times.

Table 2. 7 Use Case 7

UC-7 name: Move Tokens	
Primary actor	Player
Stakeholder & Interests	Player will be able to move the tokens according to dice rolls.
Pre-condition	1. Die must be rolled at Player's turn or 2. Die must show highest throw (6) to take out token from the starting square.
Post-condition	Token has been moved successfully.
Main Success Scenario	1. Player's all tokens shows highlight animation. 2. Player selects his/her desired Token. 3. System advances the Token to number of rolled units. 4. Token moves into required position on Game track.
Alternate Flows	2a. Player selects invalid token.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Many times.

Table 2. 8 Use Case 8

UC-8 name: Roll the Dice.	
Primary actor	Player.
Stakeholder & Interests	Player will be able to Roll the dice at his/her turn.
Pre-condition	There must be Player's turn.
Post-condition	Dice have been rolled successfully.
Main Success Scenario	1. Player selects Dice, placed on the board. 2. Dice animates. 3. System displays face value as an output.
Alternate Flows	2a. Dice does not show animation, system failure. 3a. System does not displays outcome
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Many times.

Table 2. 9 Use Case 9

UC-9 name: Pause Game	
Primary actor	Player.
Stakeholder & Interests	Player will be able to pause the game.
Pre-condition	Game environment should be opened.
Post-condition	Game has been paused successfully.
Main Success Scenario	1. Player selects the Pause option on the screen. 2. System stops the game. 3. Game pauses.
Alternate Flows	1a. Game does not pause.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Many times.

Table 2. 10 Use Case 10

UC-10 name: Resume Game	
Primary actor	Player.
Stakeholder & Interests	Player will be able to resume the game.
Pre-condition	Game should be paused.
Post-condition	Game has been resumed successfully.
Main Success Scenario	1. Player selects the Resume option 2. System loads the game from last checkpoint. 3. Game resumes.
Alternate Flows	2a. System is not be able to load the game. 3a. Game resumes.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Many times.

Table 2. 11 Use Case 11

UC-11 name: Toggle Sound	
Primary actor	Player.
Stakeholder & Interests	Player will be able to switch the game sounds ON/OFF.
Pre-condition	Game environment or Main Menu should be opened.
Post-condition	Game sounds has been switched successfully.
Main Success Scenario	1. Player selects the Speaker on the screen 2. System changes the sound. 3. Sound switches to ON/OFF.
Alternate Flows	2a. System is in mute mode. 3a. Sound does not switched.
Special Requirements	None.
Technology	Touch input. LCD display and Speakers for output
Frequency	Many times.

Table 2. 12 Use Case 12

UC-12 name: Go Back to Menu	
Primary actor	Player.
Stakeholder & Interests	Player will be able to go back to the previous Menu from the sub menus.
Pre-condition	1. Start Game Menu should be opened or 2. Game Mode should be opened.
Post-condition	Menu has been opened successfully.
Main Success Scenario	1. Player selects the Back option. 2. System loads Previous Menu screen. 3. Menu opens.
Alternate Flows	2a. Menu does not loads.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Few times.

Table 2. 13 Use Case 13

UC-13 name: Beat Token	
Primary actor	Player.
Stakeholder & Interests	Player will be able to beat other Player's Token.
Pre-condition	1. There should be Player's turn. 2. Token of different colors should be placed on same unsecured track.
Post-condition	Token has been stroked and moved to starting square successfully.
Main Success Scenario	1. Player selects his/her desired Token. 2. System moves the Token to number of rolled units. 3. System detects different colors of Token. 4. Token strikes other Token. 5. Stroked Token moves to starting square.
Alternate Flows	1a. Player select invalid Token, Token does not moves. 2a. System does not moves Token, System failure. 3a. System does not detect color.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Many times.

Table 2. 14 Use Case 14

UC-14 name: Return Home	
Primary actor	Player.
Stakeholder & Interests	Player will be able to return to Main Menu
Pre-condition	Game environment should be opened
Post-condition	Player has be returned to Main Menu successfully.
Main Success Scenario	1. Player selects Home option. 2. System asks for game quit 3. Players selects either Yes option. 4. System loads Main Menu 5. Main Menu displays.
Alternate Flows	4a. System does not loads Main Menu
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Many times.

Table 2. 15 Use Case 15

UC-15 name: Display Winner	
Primary actor	System.
Stakeholder & Interests	Player will be able to view winner's name
Pre-condition	Game environment should be opened.
Post-condition	Winning Text has been displayed successfully.
Main Success Scenario	1. Player races all his four Tokens to finishing point. 2. System detects the winner 3. System displays Player Winning text.
Alternate Flows	3a. Winning Text does not displays.
Special Requirements	None.
Technology	Touch input. LCD display for output
Frequency	Many times.

2.2.3 Software System Attributes

The Non-functional requirements of the system includes following quality attributes that affect run-time behavior, system design, and Player experience:

2.2.3.1 Reliability

The system can perform its intended functions and operations in a system's environment without experiencing any failure, except Operating System error. All errors are handled in a graceful manner. The system is 90% reliable.

2.2.3.2 Availability

Once the game is installed, it is always available unless and until the game would uninstall.

2.2.3.3 Security

There is no such security requirement defined for this system.

2.2.3.4 Maintainability

The system can be updated and modified in case of any defects. It has ability to change the system components to meet new functionality. It can easily adapt new features and customization. All upgrades can be simply and safely performed.

2.2.3.5 Portability

The system currently runs on Android Operating Systems. It is portable, as it has ability to reuse features and source code to any other Operating System as well.

2.2.3.6 Performance

All components of the system are performing well and correctly. The perceived response is immediate. The system does not waste valuable resources. It does not takes initial load time more than 10 to 12 seconds.

2.2.4 Database Requirements

No database requirement is needed for this game.

2.3 Domain Model

Domain model is an object model of problem domain. It is based on real world classes/concepts and their relationships, that is used to identify the relationships among all the entities within the scope of problem domain.

Domain model for our system is given below:

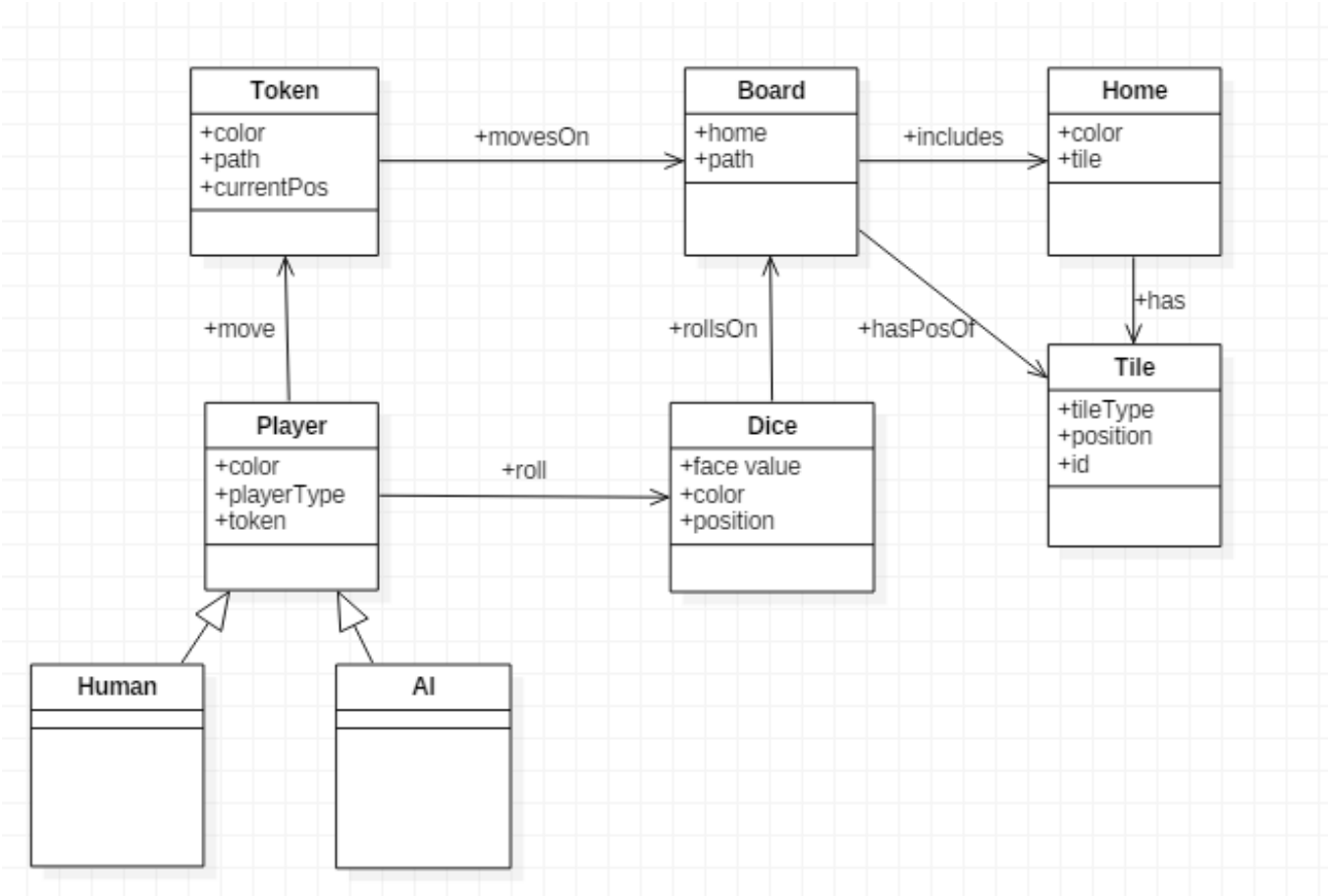

Figure 2. 2 Domain Model

Chapter 3:

Software Design

3.1 Introduction SDD

A Software Design Document (SDD) is a written description of a software product that a software designer writes in order to give a software development team overall guidance to the architecture of the software project. SDD specifies detailed features specifications of smaller pieces of the design. Usually, it contains Architecture design, Interface design, Design Class diagram, Sequence diagram and Component diagram.

This document is about software design description for the Intelligent Ludo game. This section provides you design overview, some models and interface of our project.

3.1.1 Design Overview

This document is all about description of designs of the game Intelligent Ludo. This includes Architectural design, System interface description, Component model and User interface design with includes screen images of our project. Sequence diagram and Class diagram shows the flow of the game and interaction between system and the player. This goal of this document is to provide users better understanding of the project at design level.

Algorithms are used to play the game. Alpha Beta Pruning is preferred to used, which helps player to select most preferable token to move, beat or overlap.

3.1.2 Requirement Traceability Matrix

Requirement Traceability Matrix (RTM) is a document that maps and traces user requirement with test cases. In other words Requirement Traceability Matrix or RTM captures all requirements proposed by the client or development team and their traceability in a single document delivered at the conclusion of the life-cycle. The main purpose of RTM is to see that all test cases are covered so that no functionality should miss while testing.

Table 3. 1 Requirement Traceability Matrix

Requirements	Domain Model	Sequence Diagram	Class Diagram	Test Case
Use Case				
UC-1	Figure 2		Figure 9	TC-1
UC-2	Figure 2		Figure 9	TC-2
UC-3	Figure 2	Figure 7	Figure 9	TC-3
UC-4	Figure 2		Figure 9	TC-4
UC-5	Figure 2		Figure 9	TC-5
UC-6	Figure 2		Figure 9	TC-6
UC-7	Figure 2	Figure 5	Figure 9	TC-7
UC-8	Figure 2	Figure 3	Figure 9	TC-8
UC-9	Figure 2	Figure 8	Figure 9	TC-9
UC-10	Figure 2		Figure 9	TC-10
UC-11	Figure 2		Figure 9	TC-11
UC-12	Figure 2		Figure 9	TC-12
UC-13	Figure 2	Figure 6	Figure 9	TC-13
UC-14	Figure 2		Figure 9	TC-14
UC-15			Figure 9	

3.2 System Architectural Design

A typical object-oriented information system is designed in terms of several Architectural layers or subsystems. Architectural Diagram is used to represent the components or entities of a system and the interaction between them.

It consists of three layers; Presentation, Domain and Technical services.

3.2.1 Chosen System Architecture

Our system is 2-tier Architectural design because there is no database. The components and their interaction is shown below in diagram. Double arrow line represents two way interaction between components.

Chosen architecture of the system is shown below:

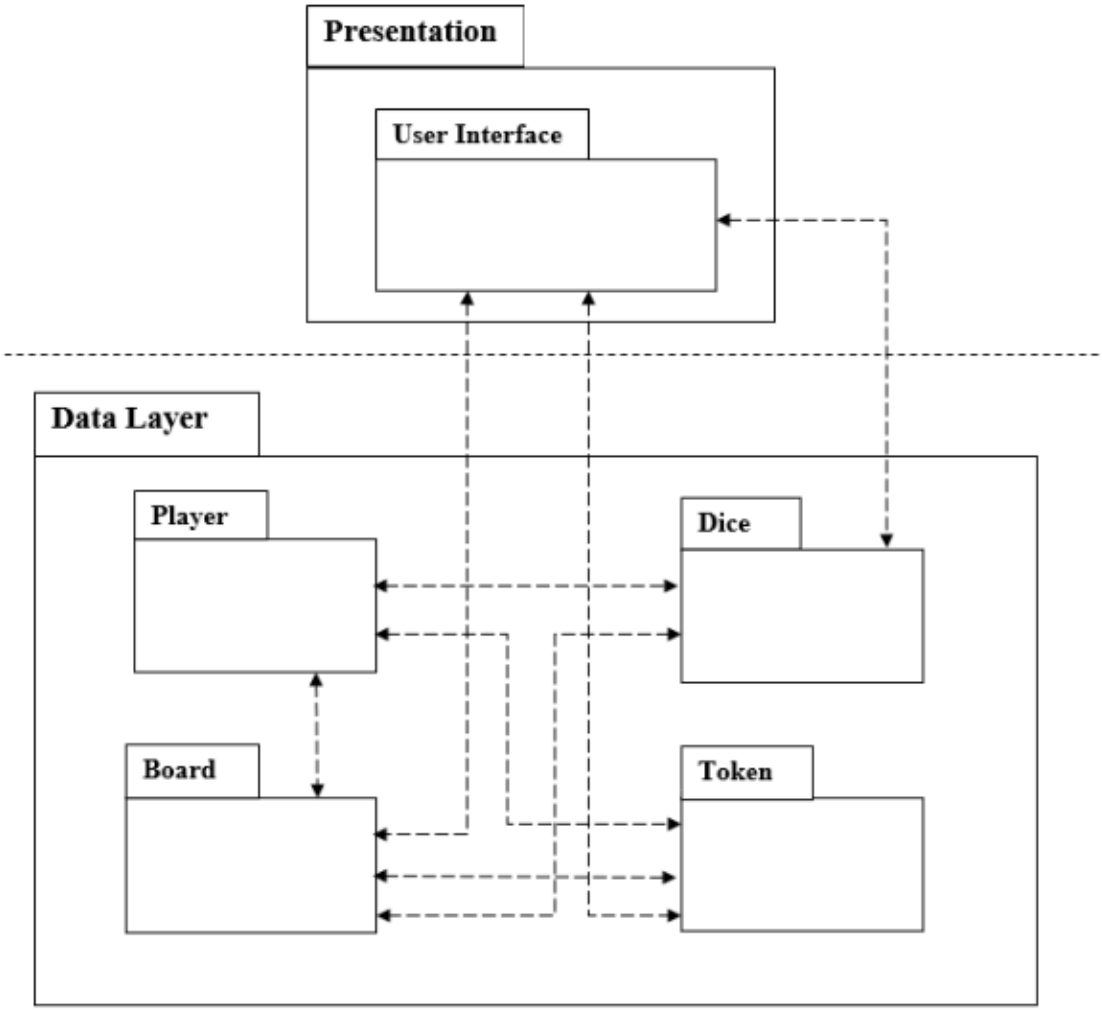

Figure 3. 1 System Architecture Design

3.2.2 Discussion of Alternative Designs

None.

3.2.3 System Interface Description

The logical characteristics of each interface between the software product and the system. The system interfaces means the internal interfaces between the software and everything else within the bigger system, which includes user interfaces, hardware interfaces and software interfaces.

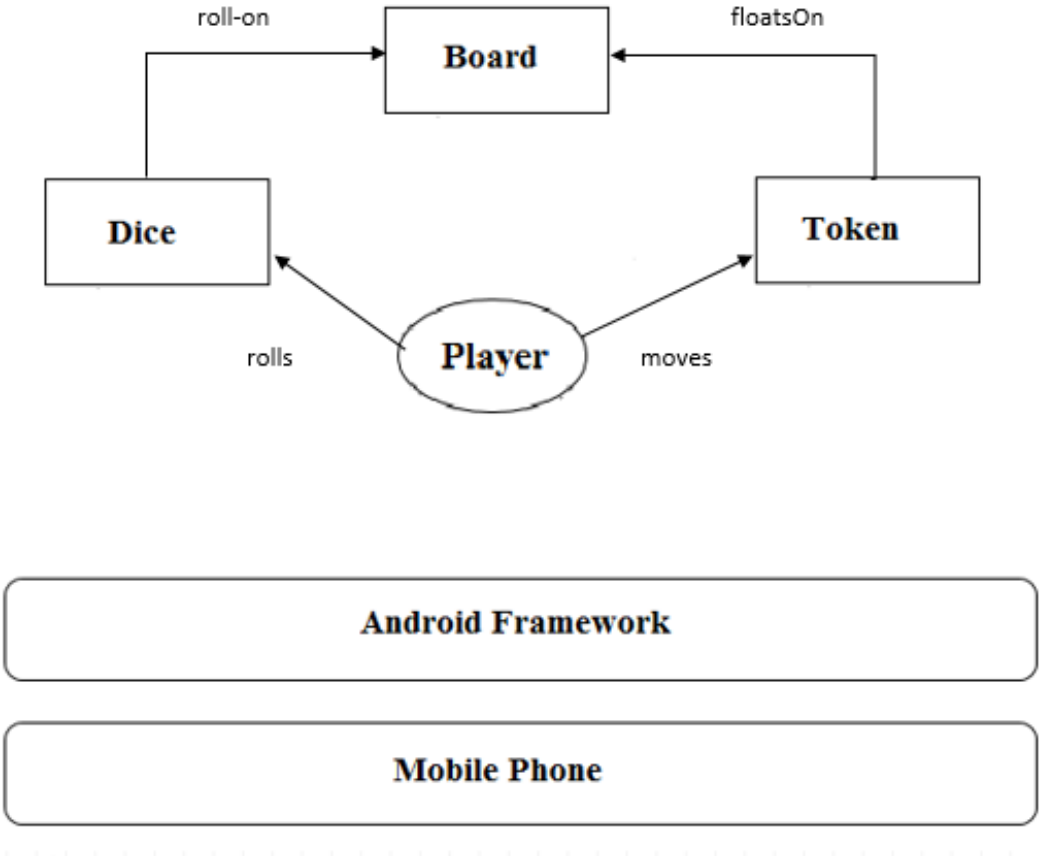

Figure 3. 2 System Interface Description

3.3 Detailed Description of Components

The detailed description of the components of Intelligent Ludo is described below:

3.3.1 Player

In this game, there are two types of players, Human and Android. The game consists of four players having four different colors. Each player has a set of four tokens. A player can roll a pair of dice, move token, beat or overlap his token. Android players plays with best possible choice using artificial intelligence which means that there are algorithms applied behind their moves.

3.3.2 Token

A token is a coin or piece which a player moves according to dice rolls. Each player has a set of four tokens. These sets are of different colors. Tokens, having same color, can be overlapped and tokens which are of different colors, can be captured or beat by opponent player.

3.3.3 Dice

A die is a six sided cube having a specific face value at each side. This game consists of two dice. A player rolls the dice at his turn. Only six sides face let a player to take out his token from starting point.

3.3.4 Board

A board is a square box consists of four columns of different colors. These columns shows the Home (starting point) of each player. It consists of number of cells on which tokens moves. There is Finishing Path for each player through which a Player races his all tokens to the Finishing Point and wins the game. Dice is also rolled on the board.

3.3.5 Game Manager

Game manager controls the other component and their interaction.

3.4 Sequence Diagram

A Sequence Diagram (SD) is an interaction diagram that shows how object operate with one another and in what order. It is a construct of a message sequence chart. In this, we identify how the function are carried out in the system. So this can help in making the different functionalities in the game. We carried out sequence diagram in each use case

3.4.1 Roll Dice

Player clicks on dice to roll and game manager gives call to Dice to roll a dice

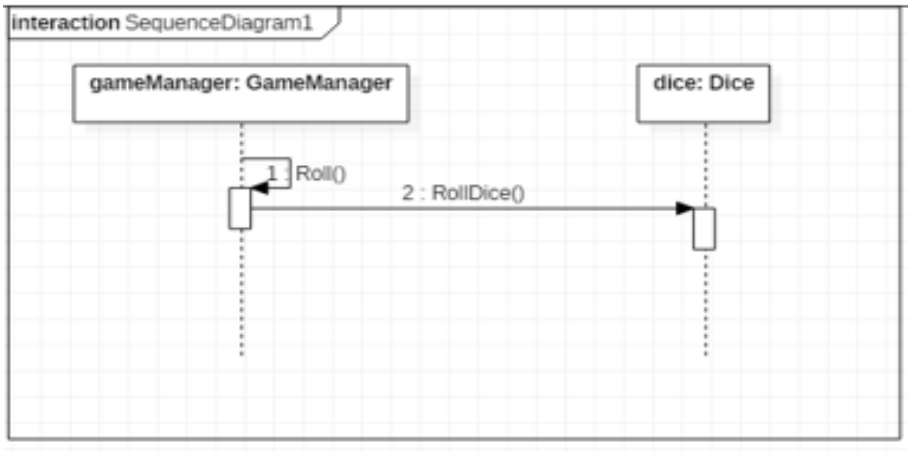

Figure 3: 3 Sequence Diagram of Roll Dice

3.4.2 Move Token (AI)

If there is AI player, then it evaluates the best choice among given choices and then move token.

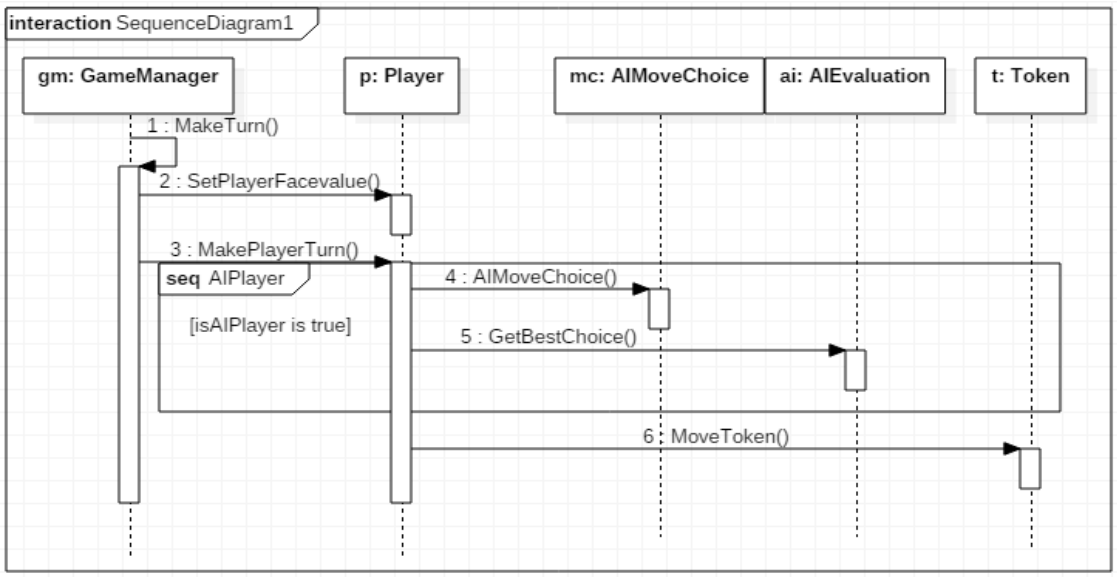

Figure 3: 4 Sequence Diagram of Move Token (AI)

3.4.3 Move Token (Human)

If there is Human player, he clicks on the highlighted token which creates a moveable at specific position and when he clicks on it, token moves to that location.

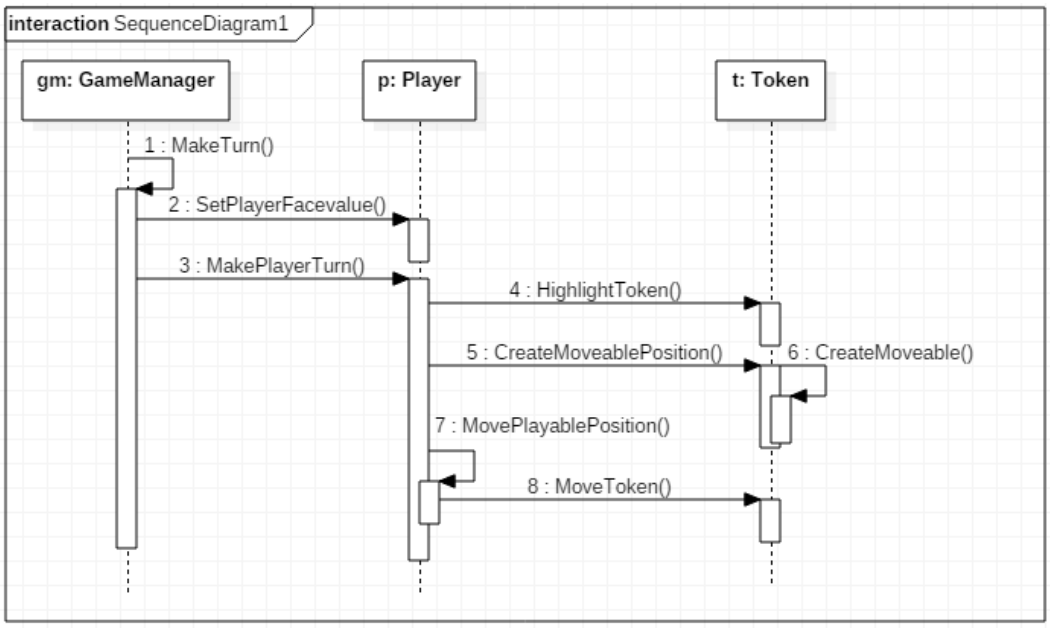

Figure 3: 5 Sequence Diagram of Move Token (human)

3.4.4 Beat Token

If there is opponent's token in the next position of the player, then it captures that and send that token to its home.

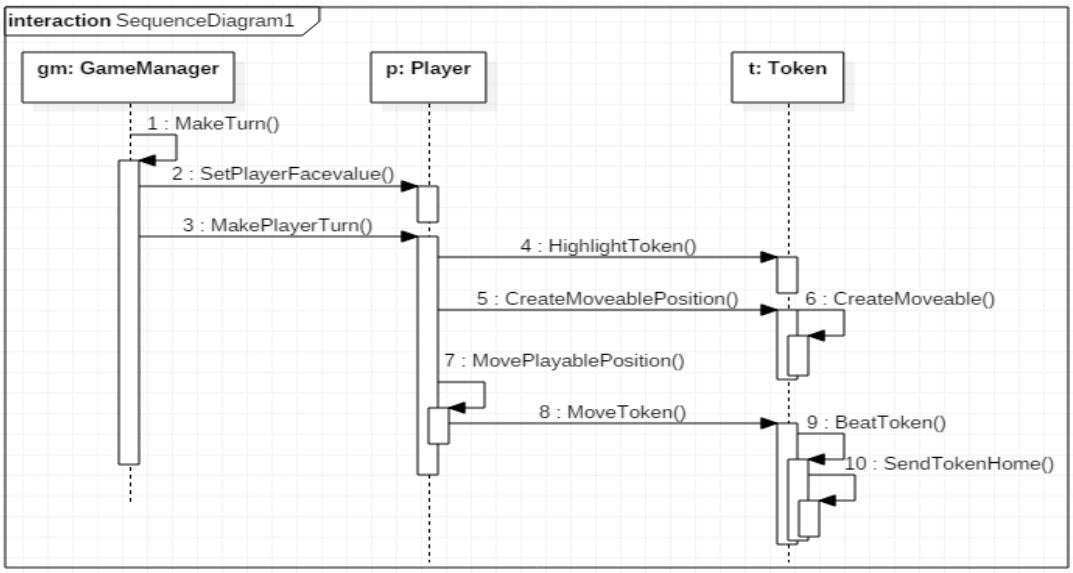

Figure 3: 6 Sequence Diagram of Beat Token

3.4.5 Select Player

User selects the Player, whether Android or Human and desired color.

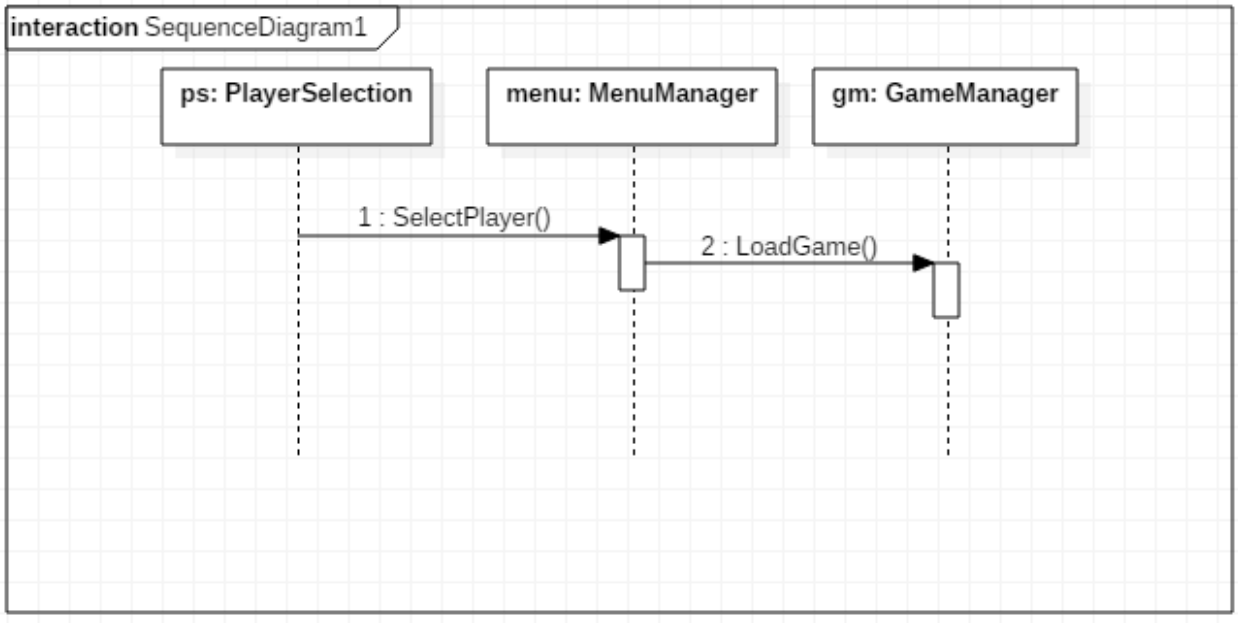

Figure 3: 7 Sequence Diagram of Beat Token

3.4.6 Pause Game

Player clicks on Pause icon to pause the game

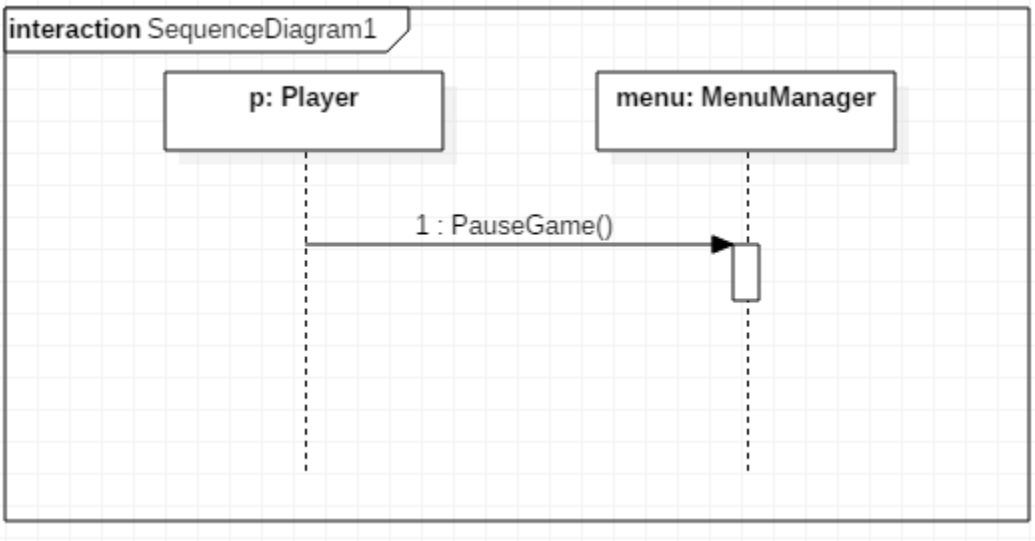

Figure 3: 8 Sequence Diagram of Pause Game

3.5 Class Diagram

A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modelling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

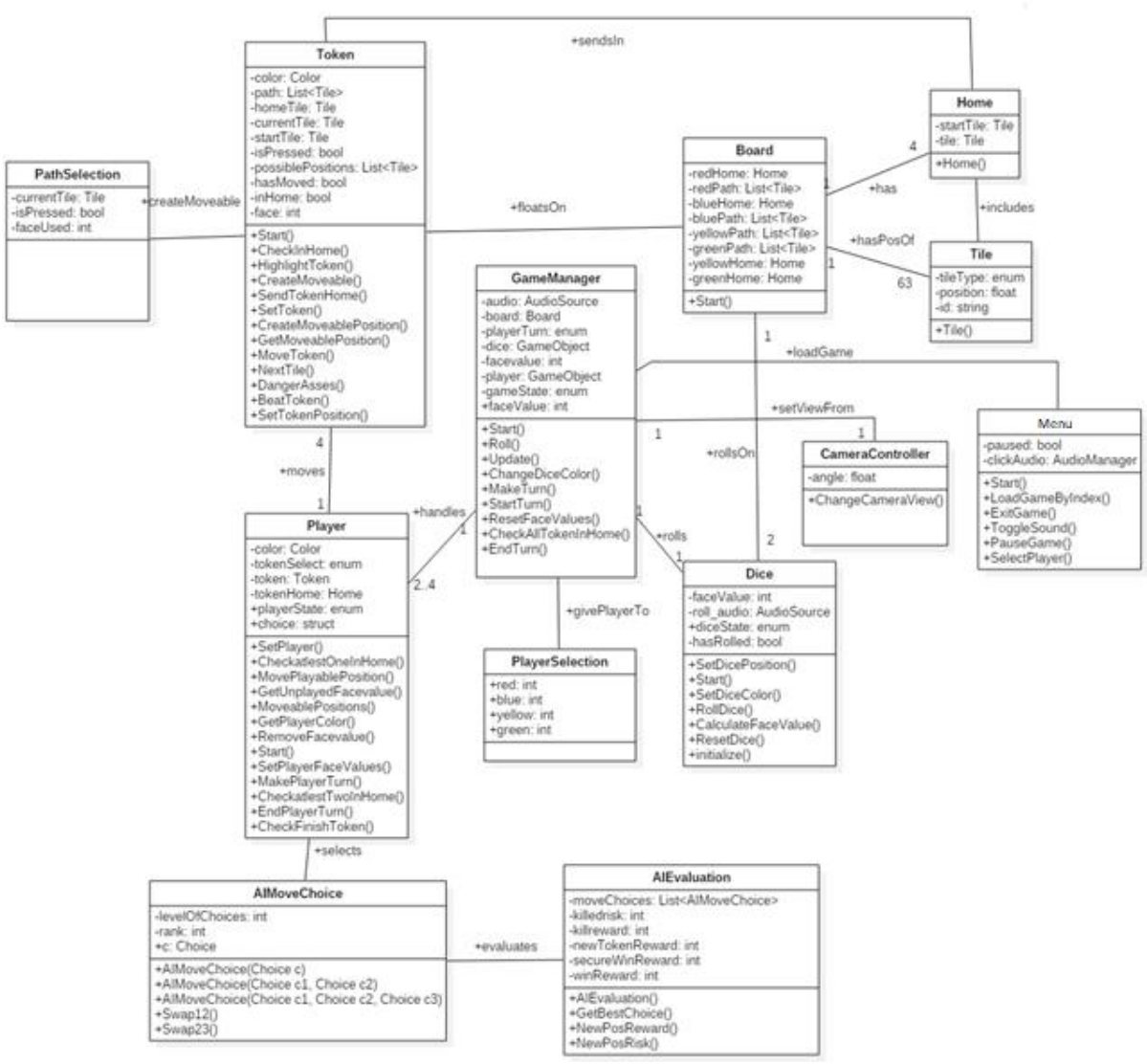

Figure 3: 9 Class Diagram

3.6 User Interface Design

The user interface (UI) is everything designed into an information device with which a person may interact. This can include display, keyboards, a mouse and the appearance of a desktop. It is also the way through which a user interacts with an application or a website. This section provides you the description of UI of Intelligent Ludo.

3.6.1 User Interface Description

In Intelligent Ludo game, a user can interact with game by using touch screen interface of Android device. User gives a touch input and see an output on the screen. When user clicks on game, a Main Menu appears. Main Menu has three buttons Start Game, Instructions and Exit. When a Player clicks on Start Game, a new Menu appears for the selection of Players and selection of the Rules. Also two buttons, Play and Back at the bottom. Player selects his/her own choice of Player (at least one Android) and selects Rules which he/she wants to play. Pressing Play button let Play moves to the Game environment and Back button let Player go to the previous Menu. In Game mode there is animation of Dice roll at each Player's turn and movement of Tokens. A sound icon and Play/Pause icon is also there.

3.6.1.1 Screen Images

Here are some screens images of our game, Intelligent Ludo.

Figure 3. 10 Screen image 1 Main Menu Menu

Figure 3. 11 Screen image 2 Selection

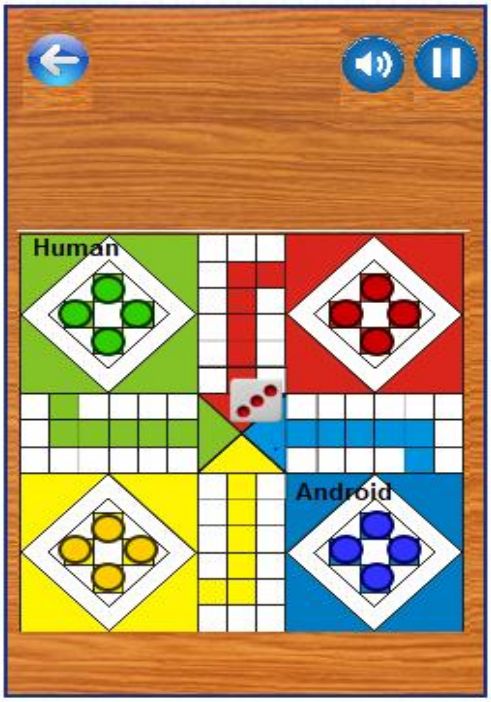

Figure 3. 12 Screen image 3 Game starts

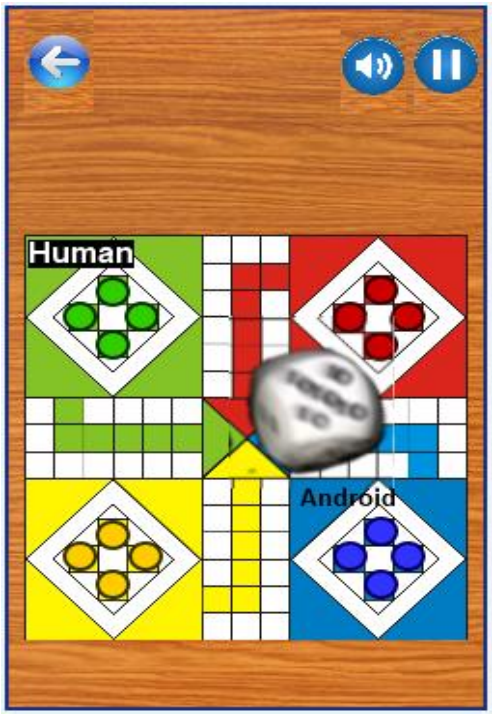

Figure 3. 13 Screen image 4 Roll Dice

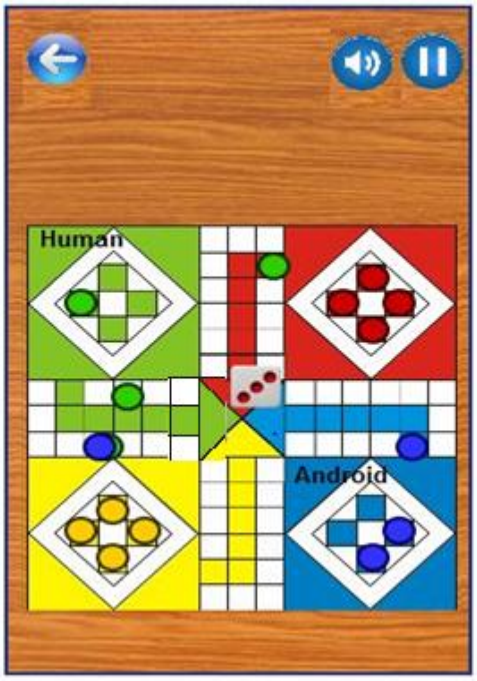

Figure 3. 14 Screen image 5 Beat Token

Figure 3. 35 Screen image 6 Player Wins

Chapter 4:

Implementation

4.1 Introduction

An implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through computer programming and deployment.

In this chapter, we provide you the framework and language selection of our product, Intelligent Ludo. Application screenshots are also given in this chapter.

4.1.1 Framework Selection

- **Unity:**

The game is developed in Unity3D v5.6. It supports Android Software Development Kit (SDK). This development tool is preferred because it is a best game developing engine. It is portable and cross platform which means that the same code, developed via Unity engine, can be ported on many platforms with minimal modifications. It runs on Android Operating System.

4.1.2 Language Selection

- **C-Sharp (C#):**

C# language is used for the development of this game because it is used in Unity. It is a managed language which means that it manages memory for you.

4.1.3 Operating System

- **Android:**

Since this is an Android based game, so Operating system for this game is one of the most widely used these day, Android.

4.1.4 Artificial Intelligence

Nowadays, the games are not just stayed to static environment but also they have been developed to intelligent dynamic environment where system acts as second player. Most people enjoy these types of games only where there is a brainstorming part represented. This part is AI (Artificial Intelligence), where the programming is done in such a way that system also act as an intelligent creature, creating interest of users in games. In this Game, AI can be implemented using algorithm

4.2 Application Screenshots

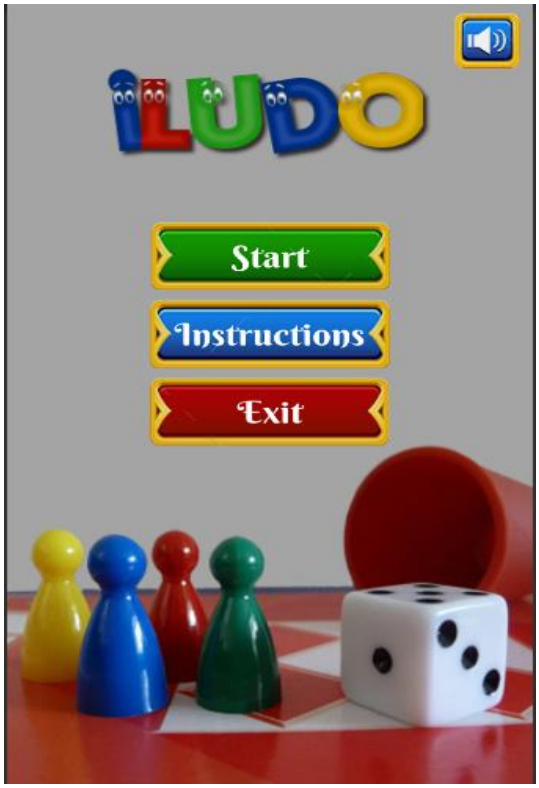

Figure 4. 1 App Screen shot 1

Figure 4. 2 App Screen shot 2

Figure 4. 3 App Screen shot 3

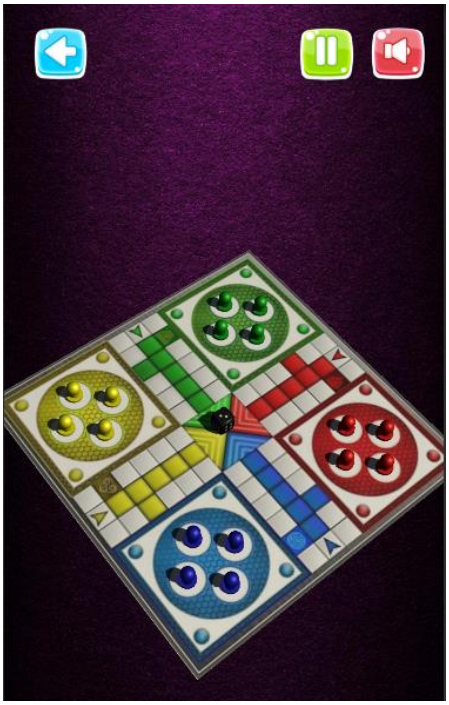

Figure 4. 4 App Screen shot 4

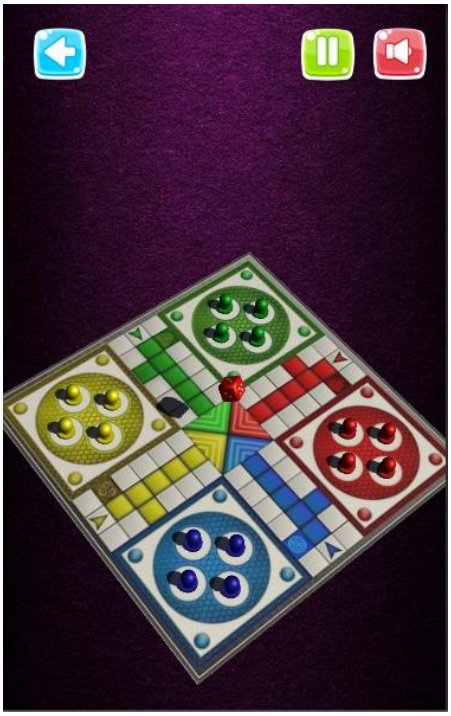

Figure 4. 5 App Screen shot 5

Figure 4. 6 App Screen shot 6

Figure 4. 7 App Screen shot 7

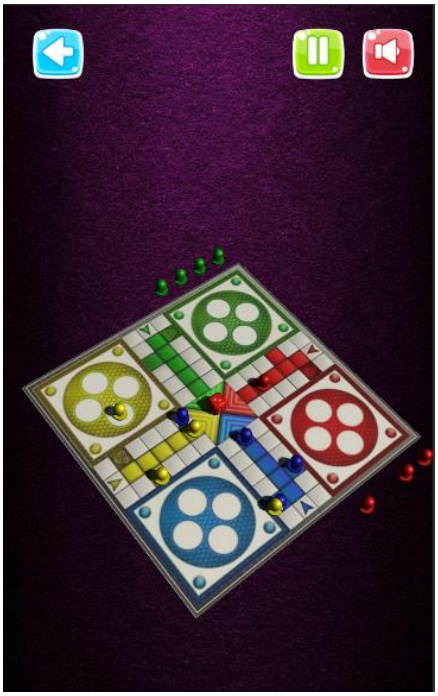

Figure 4.8 App Screen shot 8

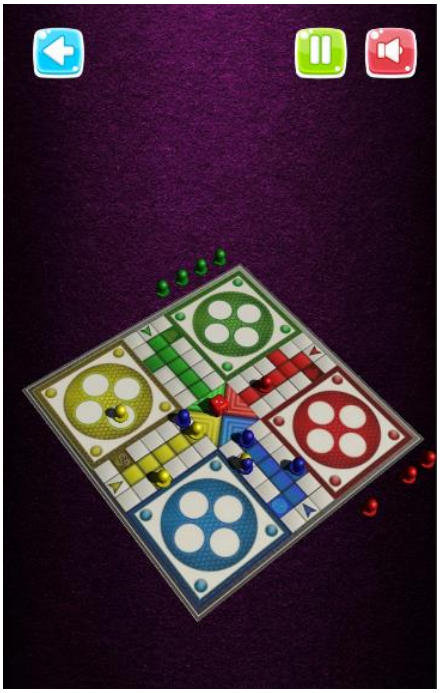

Figure 4.9 App Screen shot 9

Figure 4. 10 App Screen shot 10

Chapter 5:

Software Test Document

5.1 Introduction STD

A document describing the scope, approach, resources, and schedule of intended testing activities. It identifies test items, the features to be tested, the testing tasks, who will do each task, and any risks requiring contingency planning. Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not.

In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements. According to ANSI/IEEE 1059 standard, Testing can be defined as - A process of analyzing a software item to detect the differences between existing and required conditions (that is defects/errors/bugs) and to evaluate the features of the software item.

5.1.1 System Overview

Ludo, a famous old traditional game, is basically a board game for two to four players. In this game, every player have to race their four tokens from starting to the finishing point according to a die roll. The game is played with some rules. It is an interesting game with which many kids have spent their childhood. Everyone can play this game.

The purpose of the project is to develop an intelligently made Android based game, Intelligent Ludo. It consists of a board with tokens and die on it. It supports portability, so that user can play this game anywhere on a hand held device instead of carrying a board. User can give Touch input and receive output on the Screen.

It is stand-alone game. It also provides instructions to let user know about the rules of the game.

5.1.2 Test Approach

At this stage, test approach would be User acceptance testing (UAT) also called beta testing or end user testing. It consist of a process of verifying that a solution works for the user. It is not system testing (ensuring software does no crash and meets documented requirements), but it is a test that user acceptance the solution.

This should be undertaken by a subject-matter expert (SME), preferably the owner or client of the solution under test, and provides a summary of the findings for confirmation to proceed after trail or overview. In software development, UAT as one of the final stages of a project often occurs before a client or customer accepts the new system. Users of the system perform test in line with what occur in real life scenarios.

5.1.3 Testing Objectives

To check whether the requirement are fulfilled or not, we have to make test on these cases.

User Acceptance Test has following objectives.

- It makes test case against the requirements.
- It checks actual function, input, expected result, actual result, procedure to make test case, pass/fail status against each test case.

5.2 Test Plan

We plan to identify those features which are to be tested and those which are not.

5.2.1 Features to be tested

Following are the Features to be tested, according to client/player prospective:

- Start game
- Play game
- Pause game
- Resume game
- Select Player
- Select rules
- View instructions
- Exit game
- Change sound
- Roll dice
- Move token
- Beat token
- Return Home

5.2.2 Features not to be tested

Following are the Features not be tested, which are from developer's point of view:

- Power used by the processor
- Memory consumed by the game.
- Frame rate per second of the game
- Maintainability of the game
- Software risk factor.

5.2.3 Testing Tools and Environment

As mentioned above that it is a Beta testing, so there is no need of testing tools and environment.

5.3 Test Cases

Following are the test cases for our game, Intelligent Ludo.

Table 5. 1 Test Case 1

TC 1:	
Purpose	Start Game
Setup	Not defined
Instructions	1. Open game app 2. Main Menu displayed 3. Select Start option 4. Start Menu displayed
Expected Results	Start Menu has been appeared successfully.
Observed Results	Start Menu has been appeared successfully.
Verdict	Pass

Table 5. 2 Test Case 2

TC 2:	
Purpose	Exit Game
Setup	1. Select Home option from Game environment
Instructions	1. Go to Main Menu 2. Select Exit Game.
Expected Results	Game has been closed successfully.
Observed Results	Game has been closed successfully.
Verdict	Pass.

Table 5. 3 Test Case 3

TC 3:	
Purpose	Select Player
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select Player option.
Expected Results	Player is selected successfully.
Observed Results	Player is selected successfully.
Verdict	Pass.

Table 5. 4 Test Case 4

TC 4:	
Purpose	View Instructions
Setup	1. Open Game app
Instructions	1. Go to Main Menu 2. Select Instructions
Expected Results	Game instructions are displayed successfully.
Observed Results	Game instructions are displayed successfully.
Verdict	Pass.

Table 5. 5 Test Case 5

TC 5:	
Purpose	Select Rules
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select rules option.
Expected Results	Rules has been selected successfully.
Observed Results	Rules has been selected successfully.
Verdict	Pass.

Table 5. 6 Test Case 6

TC 6:	
Purpose	Play Game
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select Player 2. Select Rules 3. Select Play
Expected Results	Game mode has been displayed successfully.
Observed Results	Game mode has been displayed successfully.
Verdict	Pass.

Table 5. 7 Test Case 7

TC 7:	
Purpose	Move Token
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game 4. Roll Dice
Instructions	1. Select Token
Expected Results	Token has been moved according to dice roll successfully.
Observed Results	Token has been moved according to dice roll successfully.
Verdict	Pass.

Table 5. 8 Test Case 8

TC 8:	
Purpose	Roll Dice
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select Dice
Expected Results	Dice has been animated successfully.
Observed Results	Dice has been animated successfully.
Verdict	Pass.

Table 5. 9 Test Case 9

TC 9:	
Purpose	Pause Game
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select Pause option
Expected Results	Game has been stops successfully.
Observed Results	Game has been stops successfully.
Verdict	Pass.

Table 5. 10 Test Case 10

Test Case 10:	
Purpose	Resume Game
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select Resume icon
Expected Results	Game has been continues successfully.
Observed Results	Game has been continues successfully.
Verdict	Pass.

Table 5. 11 Test Case 11

TC 11:	
Purpose	Change Sound
Setup	1. Open game app 2. Go to Main Menu
Instructions	1. Select Sound icon
Expected Results	Sound has been switched successfully.
Observed Results	Sound has been switched successfully.
Verdict	Pass.

Table 5. 12 Test Case 12

TC 12:	
Purpose	Go Back
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select Back option
Expected Results	Main Menu has been displayed successfully
Observed Results	Main Menu has been displayed successfully
Verdict	Pass.

Table 5. 13 Test Case 13

TC 13:	
Purpose	Beat Token
Setup	4. Open game app 5. Go to Main Menu 6. Select Start Game 7. Roll Dice
Instructions	1. Select Token
Expected Results	Token has been moved back to staring point successfully.
Observed Results	Main Menu has been displayed successfully
Verdict	Pass.

Table 5. 14 Test Case 14

TC 14:	
Purpose	Return Home
Setup	1. Open game app 2. Go to Main Menu 3. Select Start Game
Instructions	1. Select Home icon
Expected Results	Game has been returned to Main Menu successfully
Observed Results	Game has been returned to Main Menu successfully
Verdict	Pass.

Chapter 6:

Conclusion and Future Enhancements

6.1 Summary

This is a project of an Android based game, Intelligent Ludo. It is a turn based game in which one player takes its turn while other waits. The game consists of a board with a set of four tokens and die on it. It includes four players, at least one Human and Android players. Each player has four tokens and they are differentiated by four different colors (red, blue, yellow and green). The game is done playing by rolling a die. It is stand-alone game which means that making it Android based let anyone play this game alone with Android player. This is a unique features with differs this game from previous games, that is AI (Intelligence factor). Android player plays this game with the help of artificial intelligence. The player races all of his four tokens to the finishing point first, wins the game.

6.2 Conclusions

This game is an entertaining game for all. Anyone can play this game anywhere on a hand held device instead of carrying a board. As our lives turned very fast, so this game cannot be played with speed and young people do not have much time and interest in it, so making it Android based helps young people to enjoy it by challenging Android player.

6.3 Future Enhancements

In future, this game can be enhanced by:

- For future enhancements, making it network based, in which user can play this game with socially connected friends, like on Facebook.
- Also it can be improved by making user's profile for maintaining his win or lose record.

6.4 Additional Material

There are no additional materials.

7 References

Following is the list of documents and Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document.

1. IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998 for the documentation.
2. **Book:** C. Larman, APPLYING UML AND PATTERNS An Introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd ed., Massachusetts: Pearson Education, 2005 for the documentation
3. **Website:** www.gameart2D.com for game UI design
4. **Website:** www.cooltext.com and www.fontsquirell.com for game fonts
5. **Website:** www.zapsplat.com, www.freesounds.org and www.soundbible.com for game sounds