A Major Project on

**KRISHNA SAFARI**

**(The Magical Voyage of Murliwala)**

Submitted in partial fulfillment of the requirements

for the award of the degree of

**Bachelor of Computer Application (BCA)**

To

Ansal University, Gurgaon

Guide:

Ms. Kiran Khatter

Submitted by:

Sonali Chawla (AU124060202002P)

Vinisha Sharma (AU120060301018P)



Ansal University, Gurgaon

Batch (2012-2015)

DECLARATION BY STUDENT

We, Ms. Sonali Chawla (AU124060202002P) and Ms. Vinisha Sharma (AU120060301018P) students of BCA-VIth Semester (2012-2015) batch declare that the Project Report/Dissertation entitled “KRISHNA SAFARI-THE MAGICAL VOYAGE OF MURLIWALA” is done by us and it is an authentic work carried out by us at Ansal University. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of our knowledge and belief.

**SONALI CHAWLA**

**(AU124060202002P)**

**VINISHA SHARMA**

**(AU120060301018P)**

**CERTIFICATE BY SUPERVISOR**

It is Certified that the Project Report/Dissertation entitled “KRISHNA SAFARI-THE MAGICAL VOYAGE OF MURLIWALA” done by Ms. Sonali Chawla (AU124060202002P) and Ms. Vinisha Sharma (AU120060301018P) is completed under my guidance.

**Ms. Kiran Khatter**

**Assistant Professor**

**Ansal University, Gurgaon**

**ACKNOWLEDGEMENTS**

We take this opportunity to express our profound gratitude and deep regards to our guide Ms. Kiran Khatter for her exemplary guidance, monitoring and constant encouragement throughout the development of the project. Her aspiring guidance, invaluably constructive criticism and friendly advice during project development is appreciated..The blessing, help and guidance given by her time to time will carry us a long way in the journey of life on which we are about to embark. We are also sincerely grateful to our family and friends for sharing their truthful and illuminating views on a number of issues related to the project.

**SONALI CHAWLA**

**(AU124060202002P)**

**VINISHA SHARMA**

**(AU120060301018P)**

**Executive Summary**

**1. INTRODUCTION**

This project is an extension to our project **KRISHNA SAFARI-A 3D GAME** for android users which was done in the fifth semester of our BCA program.

In today’s world, people are running behind their desires and dreams. They just want to earn money at any cost, even if they have to compromise on their morals and self esteem. The various factors such as corruption, selfishness, materialistic thinking and money mindedness have made this world a wicked place, which is continuously moving towards the end of this vicious circle. One of the main reasons behind the same is the increasing gap between this so called “modern generation” and Indian culture and mythology. This ignited the idea for developing a game based on the life of a colossus from the past, who has shown the correct path to the humanity earlier in the times of Varnveda i.e. ”***LORD KRISHNA***”.

The almighty ***Krishna*** possesses all opulent qualities in the infinite degree. Being naughty in childhood, to being brave and fighting against the evil powers of that time, makes his life a bunch of inspirational events. He was one such fine leader, who was blessed with the virtue of simplicity, intelligence, determination, forgiveness and love. He was the protector of the masses. All these inspiring characteristics and events in Lord Krishna’s life led to the thought of creating a game, which could project out all of these.

The game ***Krishna Safari*** will be designed to bring alive the various events in Lord Krishna’s life. In this, a player gets a chance of playing the entire game by being ***Lord Krishna*** himself and therefore understanding and experiencing something of what life in the spiritual world of ***Lord Krishna*** would be, when we go there.

The thought behind choosing specifically a game to demonstrate the life of ***Lord Krishna*** is that games are interesting to play. Game-based learning facilitates the process of meaning making, i.e. understanding the implication among facts. The simulation of the same era can be projected in a 3D game. It will help the user in skill building, getting motivated, attentive and most importantly transmission of knowledge.

Unity 3D, a cross-platform open source game creation system, which includes a game engine and integrated development environment (IDE) will be used for designing and developing the complete game. Character models and object models will be designed in Blender and Make human as they make the task of prototyping photo realistic humanoids very easy. For the database connectivity Oracle is going to be used.

In this game, a player will need to cross certain levels to win. These levels will be based on the facts related to Lord Krishna’s life. The transition of levels will be based on a player’s skills. He can switch to the next level only after completing the previous one. If he is not successful in clearing any particular level, the next level remains locked. All of these levels will challenge the player’s gaming as well as mental skills. A player needs to be focused while playing the game. This game can be played all over the world.

As the game is going to be based on the various episodes of Lord Krishna’s life, it’s going to teach the players a chapter of courage, honesty, morals and selflessness throughout the whole gaming experience, taking them closer to Indian culture and mythology.

**2. OBJECTIVE**

The main objective of the game ***Krishna Safari*** is to teach people that ”burai par hamesha acchai ki jeet hoti hai” i.e. in a fight between good and evil, good always win. The game will be designed in such a way that the entire Krishna saga can be projected as a series of levels, which are not only interesting to play but got various faces of life, helping a person to develop as an individual. Some of the specific objectives are observed as below:

* 1. **Promoting Indian mythology**: As the game is entirely based on Lord Krishna’s life, and promotes the same so as to make its players to get a chance to know the Indian cultures and traditions and hence learn them side by side.
  2. **Challenging and improving mental and gaming skills**: The various levels in the game will be designed in such a way that they will challenge the gaming skills of the player.
  3. **Competition to the players of all ages**: The game doesn’t target players of any specific age group. It will design in such a way that, players of all ages can play it.
  4. **Self development**: The game will teach some moral values of life, which will help in self development of the player as an individual.
  5. **Skill development:** Skills like focus, speed, hand-eye coordination and concentration can be developed by playing this game.
  6. **Providing a joyful learning experience**: Along with the knowledge, the game will also be fun to play. The various levels will be designed so that a player finds them interesting and amusing.

**3. FEASIBILITY STUDY**

* 1. **Economical Feasibility**

The project includes the use of open source softwares such as Unity3d, Blender and Makehuman. The game can be played on any version of Android with Graphics Driver and Audio Driver. Therefore, this project is economically feasible.

* 1. **Technical Feasibility**

All the inventories like interfaces and characters will be made by using softwares such as Unity3d, Blender, Makehuman and Android. These softwares are open source softwares and are easy to use. The task of prototyping of humanoid and creation of a 3D environment is made simpler by these softwares. Therefore, this project is technically feasible.

* 1. **Legal Feasibility**

The project ***Krishna Safari*** is legally feasible. It uses only the open source software and easily available hardware such as Windows Operating System and Android Operating System. The characters used in this game are not copied from anywhere. They have been created on the basis of imagination. All the scenes which will be designed in the game are based on the real events of Lord Krishna’s Life. If there is any kind similarity to any person living or dead, it is purely coincidental. The development of this game doesn’t use the copyright of others. Therefore, it is legally feasible. The project doesn’t conflicts any legal requirements which make it.

**4. METHODOLOGY**

The methodology of this project provides a detailed overview of all the guidelines, practices, procedures and working methods that will be followed in the development of the game at various levels.

* 1. **Surveying current scenario**: Now days, people are busy in their own world and forgetting the importance of Indian culture and mythology. This led the idea to create something which could take them a few steps close to the Indian mythology. Therefore, we will be designing a game which could which will be interesting to play and will help the player to develop himself as an individual, taking him a step close to Indian culture.
  2. **Game story**: A leader from the Indian mythology whose life journey was interesting and inspiring was chosen. This led to the decision of designing the plot of the entire game on the life of Lord Krishna. His life is full with inspiring and motivating events. The game will be designed so as to bring alive all of them one by one.
  3. **Softwares**: Unity 3D is organized, easily accessible software which makes tasks like modeling, rigging, animation etc easy for the developer. Blender provides Photorealistic Rendering, Fast Modeling, Realistic Materials, Fast Rigging and various animation toolset. Makehuman makes prototyping of humanoids very easy, simple and interesting. All these features led to the usage of the above mentioned softwares for the development of the game.
  4. **Feasibility study**: A feasibility study was conducted on various aspects to check whether our project is feasible or not. This study led to a conclusion that the product is technically, economically and legally feasible.
  5. **Designing characters and Elements for the game:** The characters and elements will be made in Blender and Makehuman. The design and structure of the characters and models will be based on the various incidences of Lord Krishna’s Life.
  6. **Conceptualizing Scenes and Positioning Elements and Characters:** Unity3d software will be used to make scenes. The scenes will be designed in such a way so that they can to showcase the era belonging to Lord Krishna’s life exactly as it was.

All the scenes, characters and object models made will be positioned according to their requirement in this step.

* 1. **Technical stride:**
* **Animation:** Animation to the characters and elements in scenes will be added in this step. This will be done by using unity’s animation window.
* **Audio:** Audio will be added in this step. The audio will make the scene more realistic and interesting.
* **Scripting to Scenes:** The script can be added to the scenes using unity’s monoscripting feature. The language followed will be unity’s JavaScript. The script will be added for the player’s action and elements movement.
* **Connection between Scenes:** In this last step, all the scenes so made will be connected together to form a game that will attract the player’s attention and representing the whole saga of Lord Krishna’s life.
  1. **Switching between the levels:** The next level will be unlocked only if the previous one is finished. If the player is not successful in completing any particular level, the succeeding level will remain locked and it cannot be played.
  2. **Connectivity to database:** SQL server- SQLite database will be used for data connectivity. Records of all the players are going to be stored in a SQLite database.

**5. PROPOSED TECHNOLOGY**

A wide range of softwares will be used for development of this game like software for designing models, for designing scenes, conceptualizing the whole game and for database connectivity. The softwares that will be used are listed below:

**5.1 UNITY 3D:**

Unity 3D is a [cross-platform](http://en.wikipedia.org/wiki/Cross-platform) [game creation system](http://en.wikipedia.org/wiki/Game_creation_system) developed by [Unity Technologies](http://en.wikipedia.org/wiki/Unity_Technologies), including a [game engine](http://en.wikipedia.org/wiki/Game_engine) and [integrated development environment](http://en.wikipedia.org/wiki/Integrated_development_environment) (IDE). With an emphasis on portability, the graphics engine targets the following APIs: Windows, and Linux; [OpenGL ES](http://en.wikipedia.org/wiki/OpenGL_ES) on Android and iOS; and proprietary APIs on [video game consoles](http://en.wikipedia.org/wiki/Video_game_console).

**5.2 BLENDER:**

Blender is a free and open source 3D animation suite. It supports the entirety of the 3D pipeline—modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. Blender is well suited to individuals and small studios that benefit from its unified pipeline and responsive development process.

**5.3 MAKEHUMAN:**

Makehuman is an [open source](http://en.wikipedia.org/wiki/FOSS) [3D computer graphics software](http://en.wikipedia.org/wiki/3D_computer_graphics_software) middleware designed for the prototyping of photo realistic humanoids. It uses a very simple GUI in order to access and easily handle hundreds of morphing. Makehuman uses simple sliders to alter parameters with which you can create many different types of people.

**5.4 SQL SERVER- SQLite**

SQLite is a software library that implements a [self-contained](https://www.sqlite.org/selfcontained.html), [serverless](https://www.sqlite.org/serverless.html), [zero-configuration](https://www.sqlite.org/zeroconf.html), [transactional](https://www.sqlite.org/transactional.html) SQL database engine. SQLite is the [most widely deployed](https://www.sqlite.org/mostdeployed.html) SQL database engine in the world. The source code for SQLite is in the public.

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
|  | **Page No.** |
| Declaration by student | i |
| Certicate by Supervisor | ii |
| Acknowledgements | iii |
| Executive Summary | iv |
| 1. Chapter1: Introduction | 1 |
| 1.1 Project Description | 2 |
| 1.2 Game Scenario | 2 |
| * + 1. Level 1 Scenario     2. Level 2 Scenario     3. Level 3 Scenario | 2  3  4 |
| 1.3 Objective | 5 |
| 1.4 Scope | 6 |
| 1. Chapter 2: Software Development Life Cycle | 7 |
| 2.1 Model Used | 8 |
| 2.1.1 Waterfall Model | 8 |
| 1. Chapter 3: Software Requirement Analysis | 10 |
| 3.1 Introduction | 11 |
| 3.2 Overall Description | 12 |
| 3.3 Specific Requirements | 13 |
| 1. Chapter 4: Software Design | 16 |
| 4.1 Game Flow Chart  4.2 Level 1 flow chart  4.3 Level 2 flow chart  4.4 Level 3 flow chart | 17  18  19  20 |
| 1. Chapter 5: Software Development (Coding) | 21 |
| 5.1 Coding  5.1.1 Level 1 coding  5.1.2 Level 2 coding  5.1.3 Level 3 coding | 22  22  26  30 |
| 5.2 Screenshots  5.2.1 Level 1 screenshots  5.2.2 Level 2 screenshots  5.2.3 Level 2 screenshots | 33  33  34  36 |
| 1. Conclusion | 37 |
| 1. Chapter 7: Suggestions For Further Study or Further Improvement | 39 |
| Bibliography | 41 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **TITLE** | **PAGE NO.** |
| Fig 2.1 | Water fall model phases | 8 |
| Fig 4.1 | Game flow chart | 17 |
| Fig 4.2 | Level 1 flow chart | 18 |
| Fig 4.3 | Level 2 flow chart | 19 |
| Fig 4.4 | Level 3 flow chart | 20 |
| Fig 5.1 | Start screen-level1 | 33 |
| Fig 5.2 | Sheshnaga | 33 |
| Fig 5.3 | Game over screen –level 1 | 34 |
| Fig 5.4 | Start screen-level 2 | 34 |
| Fig 5.5 | Game over screen-level 2 | 35 |
| Fig 5.6 | Win screen –level 2 | 35 |
| Fig 5.7 | Start screen- level 3 | 36 |
| Fig 5.8 | Game over screen-level 3 | 36 |
| Fig 5.9 | Win screen- level 3 | 36 |

CHAPTER 1

INTRODUCTION

* 1. **PROJECT DESCRIPTION**

The project focuses on the development of a new game that will not only be interesting to play but will also help the player to learn something from it. The game ***Krishna Safari*** isbased on Indian Mythology and can be played on a mobile phone having android version 2.3 or above. The idea behind choosing Lord Krishna’s life for making a game is to teach people that whatever bad we do, in the end good wisdom always win. This game will help to decrease a gap between this so called modern society and Indian culture and mythology.

In this game the player gets an opportunity to experience the life of Lord Krishna. These levels will challenge the player’s gaming as well as mental skills.

The models for the game are designed in Blender and make human and the interface for the game is designed in Unity 3D.

* 1. **GAME SCENERIO**

The game is based on the various life events of Lord Krishna. His life story is divided into series of levels. There are total three levels. To go to next level, a player first needs to finish the preceding level.

**1.2.1 LEVEL 1 SCENERIO**

Lord Krishna was the son of Devaki and Vaasudeva. Devaki was the sister of Kansa and daughter of King Uggarsen of Mathura. King Kansa was a cruel King. He was known for his brutishness. He arranged the marriage of his sister Devaki with one of his deserving friend Vaasudeva.  After their marriage when King Kansa was about to bid farewell to Devaki and Vaasudeva, he decided to ride the marriage chariot till Vaasudeva’s home. While he war ridding their marriage chariot he heard the Aakash Vani (voice coming from Sky i.e. heavens) that he will be killed by Devaki’s eighth Son.

Hearing this, the evil Kansa stepped into the marriage chariot and tried to kill Devaki with his sward but Vaasudeva stops him. Vaasudeva convinced him saying that they will submit their eighth child and he can do anything to him. Kansa imprisoned Devaki and Vaasudeva; he even imprisoned his father Uggarsen.

When Lord Krishna was born, his uncle Kansa wants to kill him to become more powerful. He believed that Lord Krishna’s birth will bring him misery, misfortune and death. To save Little Krishna from his uncle Kansa, Vaasudeva ji rescued him to Gokul to Nanda’s house. To do so, he has to cross the storming Yamuna River.

In the game ***Krishna Safari*** the player has to help Vaasudeva to cross the river, crossing various wooden logs.

**FORMAT**

* First level is divided into four in four laps.
* In every lap, Vasudeva ji has to cross ten wooden logs.
* Multiplier will increase by 1x after every lap.
* Touch screen to increase speed.
* Collect snake ball after 2nd lap and get the help of Sheshnaga.
* Finish game in less than 125 seconds to get time bonus.

**1.2.1 LEVEL 2 SCENERIO**

Once the people of Vraj started planning for the puja and yagya for Lord Indra. But Lord Krishna, the supreme personality of Godhead, questioned them as to why they are serving Lord Indra. On seeing that the inhabitants of Vrindavan had neglected to worship him, lndra, the King of Heaven, decided to punish them by sending terrible rain clouds to inundate the land of Vrindavan. Within no time torrents of rain as thick as pillars, accompanied by thunder, lighting and howling winds, descended on Vrindavan, causing great misery to its inhabitants. Understanding the situation, Krishna immediately lifted Govardhan Hill with His left hand, just as a child picks up a mushroom, and held it up like an umbrella. Bringing all their household possessions, the inhabitants of Vrindavan, along with their cows, took shelter from the torrential rains under Govardhan Hill.

As we all know, Lord Krishna used to love butter. In this level help Krishna to break pitchers filled with butter as much as he can in the given time, so as to become strong and lift Govardhan Parvat.

**FORMAT**

* Swipe on the pitchers to crack them.
* Hold yourself from swiping on the bombs.
* For each pitcher cracked, a player gets +1 point and for each bomb exploded, -2 points are given.
* Crack more than 20 pitchers in 15 seconds to finish the level.

**1.2.3 LEVEL 3 SCENERIO**

During Mahabharata, Once Arjuna and Duryodhana went to Lord Krishna to seek help for the war between Kaurvas and Pandvas. Duryodhana took the huge army of Lord Krishna (Narayeni Sena) where as Arjuna only wanted Krishna to be with him. So, Lord Krishna took up the work of the charioteer for Arjuna.

**1.3 OBJECTIVE**

The main objective of the game ***Krishna Safari*** is to teach people that ”burai par hamesha acchai ki jeet hoti hai” i.e. in a fight between good and evil, good always win. The game will be designed in such a way that the entire Krishna saga can be projected as a series of levels, which are not only interesting to play but got various faces of life, helping a person to develop as an individual. Some of the specific objectives are observed as below:

* **Promoting Indian mythology**: As the game is entirely based on Lord Krishna’s life, and promotes the same so as to make its players to get a chance to know the Indian cultures and traditions and hence learn them side by side.
* **Challenging and improving mental and gaming skills**: The various levels in the game will be designed in such a way that they will challenge the gaming skills of the player.
* **Competition to the players of all ages**: The game doesn’t target players of any specific age group. It will design in such a way that, players of all ages can play it.
* **Self development**: The game will teach some moral values of life, which will help in self development of the player as an individual.
* **Skill development:** Skills like focus, speed, hand-eye coordination and concentration can be developed by playing this game.
* **Providing a joyful learning experience**: Along with the knowledge, the game will also be fun to play. The various levels will be designed so that a player finds them interesting and amusing.

**1.4 SCOPE**

Krishna Safari is a game that will not only be interesting to play but will also help the player to learn something from it. The game ***Krishna Safari*** isbased on Indian Mythology and can be played on a mobile phone having android version 2.3 or above.

There will be three levels of the game. All these levels will challenge players of all ages. The idea behind choosing Lord Krishna’s life for making a game is to teach people that whatever bad we do, in the end good wisdom always win. Based on this, various versions of the game can be launched in the future.

CHAPTER 2

SOFTWARE DEVELOPMENT

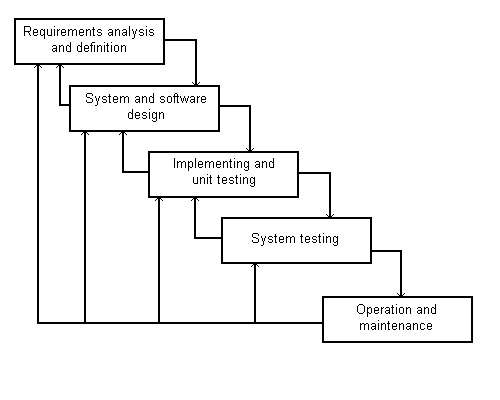
LIFE CYCLE

**2.1 MODEL USED**

The water fall model was followed for the development of the development of the project as all the requirements were known before starting the development of the project.

**2.1.1 WATERFALL MODEL**

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases as shown in figure 2.1. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.



**Fig 2.1 Water fall model phases**

1. **Requirement analysis and specification phase**

It is the most crucial phase for the whole project, in which all the requirements for the software are listed down in detail. The functionality and limitations (if there are any) of the game are explored, in detail. A small amount of top-level analysis and design is also documented

1. **Design phase**

Using SRS as input, system design is done. System design included designing of software and hardware i.e. functionality of hardware and software is separated-out. After separation design of software modules is done.

1. **Implementation and unit testing phase**

The code is developed in this phase after finishing the software design. All the individual modules of the software were tested separately to check their working individually.

1. **Implementation and system testing phase**

Now the units of the software are integrated together and a system is built. So we have complete software at hand which is tested to check if it meets the functional and performance requirements of the customer. Testing is done, as per the steps defined in the test plan, to ensure defined input produces actual results which agree with the required results. A test report is generated which contains test results.

1. **Operation and maintenance phase**

Now that we have completed the tested game, we deliver it to the client. His feed-backs are taken and any changes, if required, are made in this phase. This phase goes on till the software is retired.

CHAPTER 3

SOFTWARE REQUIREMNET   
ANALYSIS

**SOFTWARE REQUIREMENT SPECIFICATION (SRS)**

* 1. **INTRODUCTION**

This document aims at defining the overall software requirements for the development of the game ***‘Krishna Safari’***. Efforts have been made to define the requirements exhaustively and accurately. The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing/ implementing/ testing/ using this product. In this case it is required to have some additional features, a formal change request will need to be raised and subsequently a new release of this document and/or product will be produced.

* + 1. **PURPOSE**

This specification document describes the capabilities that will be provided by the game ‘***Krishna Safari’***. The game will be an interactive mobile game with a graphical interface with three levels. In the first level the player has to help Vaasudeva to safely rescue Lord Krishna to Nanda’s home in Gokul village. In the second level, a player needs to break pitchers in a given time and finally in the third level, win the chariot race. It also states the various required constraints by which the system will abide. The intended audiences for this document are the development team, testing team and end users of the product.

* + 1. **SCOPE**

The requirements specified in this document will be used for designing all the aspects and components of the game. The document will be updated as the requirements grow and change over the design and development process.

* + 1. **REFERENCES**

1. IEEE recommended practice for software requirements specifications- IEEE STD 830-993
   * 1. **OVERVIEW**

There are two main sections in this document:

• Section 2 will provide an overview of our vision of the game including our perspectives of the game, assumptions of user characteristics and interactions, and some design constraints.

• Section 3 will go into detail for each requirement: usability, reliability, performance, supportability, design constraints, documentation, and interfaces.

* 1. **OVERALL DESCRIPTION**

**3.2.1 PRODUCT PERSPECTIVE**

There is only one main component of the system i.e. the game is the mobile screen through which a will player play the game.

**3.2.2 PRODUCT FUNCTIONS**

A summary of the major functions that the game will perform:

* The game will provide entertainment to its users
* It will be a learning experience for the players

**3.2.3 USER CHARACTERISTICS**

This game is targeted directly towards the people of all age groups.

**3.2.4 CONSTRAINTS**

Android versions below 2.3 are not supported by the game

**3.2.5ASSUMPTIONS AND DEPENDENCIES**

It is assumed that the player has an android phone with version 2.3 or above

* 1. **SPECIFIC REQUIREMENTS**
     1. **EXTERNAL INTERFACE REQUIREMENTS**

**(a)USER INTERFACES**

The interface for the game will be entertaining and engaging. The function of the buttons will be easy to understand and simple to use. Menus will be interactive and easily accessible throughout the game. Once the game is in playing mode, everything a player needs will be clearly visible on the screen and easily accessible. Player will find the most basic functions of the game fun to play, from crossing the river to the background music.

**Game Restart screen**: This screen will appear when the game will get over. It will have a restart button and will show player’s score

**(b)HARDWARE INTERFACES**

1. Screen resolution of at least 800\*600 required for proper and complete viewing of screens. Higher resolution would not be a problem.
2. Standalone system
3. Android based smart mobile phones

**(c)SOFTWARE INTERFACES**

1. Any windows-based operating system
2. UNITY 3D:For developing game interface
3. Blender and Makehuman: For designing human models
4. Android 2.3 or above in the smart phone

**(d)COMMUNICATION INTERFACES**

Communication interface will be the mobile screen through which a player will play the game.

* + 1. **ATTRIBUTES**

**(a)Reliability**

The system should never crash or hang, other than as the result of an operating system error.

**(b)Availability**

There are no specific availability requirements

**(c) Security and Privacy**

There are no specific security and privacy requirements, other than those generally governing use of student login accounts on University of Toronto computer equipment.

**(d) Maintainability**

All code shall be fully documented. Each function shall be commented with pre- and post-conditions. All program files shall include comments concerning authorship and date of last change. The code should be modular, to permit future modifications. Anticipated updates include changes to the sets of objects and their descriptions used during the game. These should be stored in a separate data file, rather than embedded in the program code.

* + 1. **OTHER REQUIREMENTS**

There are no other requirements.

CHAPTER 4

SOFTWARE DESIGN

**4.1 GAME FLOW CHART**

YES

YES

NO

NO

YES

LEVEL 3

LEVEL 2

GAME OVER

LEVEL 1

START

NO

RSTART

COMPLETE

COMPLETE

COMPLETE

YES

END

**Fig. 4.1 game flow chart**

**4.2 LEVEL 1 FLOW CHART**

Start

Initialize

speed=0.5

multiplier=1x

Countdown starts

Logs instantiate()

Game continues with the current multiplier

Game complete

GUIText=”you win”

Scores displayed with time bonus included

Scores displayed without time bonus

Stop

Hit by log

GUIText=”game over”

Restart button= enable

Hit by log

Snake ball

Snake visibility=true

Multiplier+=1x

<125seconds

Yes

Yes

No

No

Collected

Not Collected

Yes

No

Click Restart Button

No

Yes

**Fig. 4.2 level 1 flow chart**

**4.2 LEVEL 2 FLOW CHART**

Time=20

Score=0

Time=Time-1

Swipe Objects

Countdown Start

Objects Instantiate

Instantiate Splash

Score=Score+1

Instantiate Explosion

Score=Score-2

GUIText=”You Win”

Display Score=Score\*10

GUIText=”Game Over”

Enable Restart Button

Object=”pot”

Time=0

Score>=20

Click Restart Button

Yes

No

Yes

Yes

No

No

**Fig. 4.3 level 2 flow chart**

**4.4 LEVEL 3 FLOW CHART**

Score=0

Time=85

Countdown Start

Time=Time-1

Translate player

x=x-1

Translate player

x=x+1

Instantiate Arrow

Score=Score+10

GUIText=”You Win”

Display Score=Score\*10

GUIText=”Game Over”

Enable Restart Button

Press Button

Tap Opponent

Time=0

Hit Opponent

Score>=100

Press Restart Button

Yes

No

Right

Left

Yes

No

Yes

Yes

No

Yes

No

No

**Fig. 4.4 level 3 flow chart**

CHAPTER 5

SOFTWARE DEVELOPMENT

**5.1 CODING**

**5.1.1 LEVEL 1 CODING**

* **creating graphical shader for the ocean waves**

**OceanIndie.shader**

Shader "Ocean/Indie/OceanShader"

{

Properties

{

\_SunPow("SunPow", float) = 256

\_SeaColor("SeaColor", Color) = (1,1,1,1)

\_SkyBox("SkyBox", CUBE) = "" {}

}

SubShader

{

Tags { "Queue"="Transparent" "IgnoreProjector"="True" "RenderType"="Transparent" }

Blend SrcAlpha OneMinusSrcAlpha

LOD 200

CGPROGRAM

#pragma surface surf Lambert vertex:vert

#pragma target 3.0

#pragma glsl

uniform sampler2D \_FresnelLookUp, \_Map0, \_Map1, \_Map2;

uniform float4 \_GridSizes;

uniform float3 \_SunColor, \_SunDir;

uniform float \_MaxLod, \_LodFadeDist;

float \_SunPow;

float3 \_SeaColor;

samplerCUBE \_SkyBox;

struct Input

{

float3 worldPos;

float3 worldRefl;

INTERNAL\_DATA

};

void vert(inout appdata\_full v)

{

v.tangent = float4(1,0,0,1);

float3 worldPos = mul(\_Object2World, v.vertex).xyz;

float dist = clamp(distance(\_WorldSpaceCameraPos.xyz, worldPos) / \_LodFadeDist, 0.0, 1.0);

float lod = \_MaxLod \* dist;

float ht = 0.0;

ht += tex2Dlod(\_Map0, float4(worldPos.xz/\_GridSizes.x, 0, lod)).x\*2.0-1.0;

ht += tex2Dlod(\_Map0, float4(worldPos.xz/\_GridSizes.y, 0, lod)).y\*2.0-1.0;

v.vertex.y += ht;

}

float Fresnel(float3 V, float3 N)

{

float costhetai = abs(dot(V, N));

return tex2D(\_FresnelLookUp, float2(costhetai, 0.0)).a \* 0.7; //looks better scaled down a little?

}

float3 Sun(float3 V, float3 N)

{

float3 H = normalize(V+\_SunDir);

return \_SunColor \* pow(abs(dot(H,N)), \_SunPow);

}

void surf(Input IN, inout SurfaceOutput o)

{

float2 uv = IN.worldPos.xz;

float2 slope = float2(0,0);

slope += tex2D(\_Map1, uv/\_GridSizes.x).xy\*2.0-1.0;

slope += tex2D(\_Map1, uv/\_GridSizes.y).zw\*2.0-1.0;

slope += tex2D(\_Map2, uv/\_GridSizes.z).xy\*2.0-1.0;

slope += tex2D(\_Map2, uv/\_GridSizes.w).zw\*2.0-1.0;

float3 N = normalize(float3(-slope.x, 2.0, -slope.y));

float3 N2 = normalize(float3(-slope.x, 0.5, -slope.y));

float3 V = normalize(\_WorldSpaceCameraPos-IN.worldPos);

float fresnel = Fresnel(V, N);

o.Normal = N.xzy;

float3 skyColor = texCUBE(\_SkyBox, WorldReflectionVector(IN, o.Normal)\*float3(-1,1,1)).rgb;

o.Albedo = lerp(\_SeaColor, skyColor, fresnel) + Sun(V,N2);

o.Alpha = 2.0;

}

ENDCG

}

FallBack "Mobile Diffuse"

}

* **initialize the logs, change speed when screen is touched and increase multiplier by 10 with each lap**

**Control.js**

function Start(){

StartCoroutine (oWaves ());

}

function Update(){

if (Input.GetKeyDown(KeyCode.Escape))

Application.Quit();

if(Input.touchCount > 0 && Input.GetTouch(0).phase==TouchPhase.Began)

{

m.speed+=0.5;

}

if(Input.touchCount > 0 && Input.GetTouch(0).phase==TouchPhase.Ended)

{

m.speed-=0.5;

}

}

function oWaves ()

{

var i:int;

var oRotation : Quaternion = Quaternion.Euler(90,0,0);

var oPosition:Vector3;

for(i=0;i<10;i++)

{

oPosition = new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

oPosition = new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

oPosition = new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

oPosition = new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

score\_no+=mul;

yield WaitForSeconds (3.5f);

}

yield WaitForSeconds(3.0f);

for(i=0;i<10;i++)

{

mul+=10;

oPosition= new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

oPosition = new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

oPosition = new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

oPosition = new Vector3 (Random.Range(-oValues.x,oValues.x), oValues.y, -oValues.z);

Instantiate (ob, oPosition, oRotation);

score\_no+=mul;

yield WaitForSeconds (2.0f);

}

}

* **Reload level when restart button is clicked**

**Redolevel.js**

#pragma strict

function OnGUI()

{

if(GUI.Button(Rect (150,100,100,40),"Restart"))

{

Application.LoadLevel("level1");

}

}

**5.1.2 LEVEL 2 CODING**

* **Timer.js //To enable/disable timer clock**

#pragma strict

private var timeTF : GUIText;

public var alertReference : GameObject;

private var scoreSF: GUIText;

public var youWT : GameObject;

public var totalS: GUIText;

private var rdl : redolevel;

function Start () {

Screen.sleepTimeout = SleepTimeout.NeverSleep;

Screen.orientation=ScreenOrientation.LandscapeLeft;

Camera.main.GetComponent(AudioSource).Play();

rdl=Camera.main.GetComponent(redolevel);

rdl.enabled=false;

totalS.enabled=false;

timeTF = gameObject.guiText;

scoreSF=GameObject.Find("score\_text").guiText;

Time.timeScale=1;

InvokeRepeating("ReduceTime",1,1);

}

function Update () {

if (Input.GetKeyDown(KeyCode.Escape))

Application.Quit();

}

function ReduceTime() {

if(timeTF.text=="1")

{

Time.timeScale=0;

if(int.Parse(scoreSF.text)>=20)

{

Instantiate(youWT,new Vector3(0.5f,0.5f,0),transform.rotation);

totalS.text=(int.Parse(scoreSF.text)\*10).ToString();

totalS.enabled=true;

Camera.main.GetComponent(AudioSource).Stop();

}

else{

Instantiate(alertReference,new Vector3(0.5f,0.5f,0),transform.rotation);

rdl.enabled=true;

Camera.main.GetComponent(AudioSource).Stop();

}

}

timeTF.text=(int.Parse(timeTF.text)-1).ToString();

}

* **Pot.js //To destroy pots and bombs when user uses swipe**

#pragma strict

public var splashReference: GameObject;

private var randomPos: Vector3=new Vector3(Random.Range(-1,1),Random.Range(0.3f,0.7f),Random.Range(-6.5f,-7.5f));

private var scoreReference:GUIText;

function Start () {

scoreReference=GameObject.Find("score\_text").guiText;

}

function Update () {

if(gameObject.transform.position.y<-36)

{

Destroy(gameObject);

}

}

function OnTriggerEnter(other:Collider)

{

if(other.gameObject.tag=="Line" && gameObject.tag=="pot")

{

audio.Play();

Destroy(gameObject);

Instantiate(splashReference,randomPos,Quaternion.identity);

scoreReference.text=(int.Parse(scoreReference.text)+1).ToString();

}

else if

(other.gameObject.tag=="Line" && gameObject.tag=="bomb")

{

audio.Play();

Destroy(gameObject);

scoreReference.text=(int.Parse(scoreReference.text)-1).ToString();

}

}

* **MtkiSpawn.js //To initiate pots and bombs at random range**

#pragma strict

private var throwForce:Vector3=new Vector3(0,15,0);

public var mtkiReference:GameObject[];

function Start () {

InvokeRepeating("SpawnMtki",Random.Range(0.1f,0.5f),Random.Range(1.0f,5.0f));

}

function Update () {

}

function SpawnMtki(){

var i:byte;

for(i=0;i<4;i++)

{

var mtki:GameObject=Instantiate(mtkiReference[Random.Range(0,4)],new Vector3(Random.Range(-10,10),Random.Range(-10,-5),5),Quaternion.Euler(270,90,0)) as GameObject;

mtki.rigidbody.AddForce(throwForce,ForceMode.Impulse);

}

}

* **LineHandler.js //To draw line when user uses swipe**

#pragma strict

public var c1 : Color = Color.yellow;

public var c2 : Color = Color.red;

private var lineGo : GameObject;

private var lineRenderer : LineRenderer;

private var i : int = 0;

function Start () {

lineGo = new GameObject("Line");

lineGo.tag = "Line";

lineGo.AddComponent(LineRenderer);

lineRenderer=lineGo.GetComponent(LineRenderer);

lineRenderer.material=new Material(Shader.Find("Mobile/Particles/Additive"));

lineRenderer.SetColors(c1, c2);

lineRenderer.SetWidth(0.3f, 0);

lineRenderer.SetVertexCount(0);

}

function Update () {

if(Input.touchCount > 0)

{

var touch : Touch = Input.GetTouch(0);

if(touch.phase == TouchPhase.Moved)

{

lineRenderer.SetVertexCount(i+1);

var mPosition : Vector3 = new Vector3(Input.mousePosition.x, Input.mousePosition.y, 15);

lineRenderer.SetPosition(i, Camera.main.ScreenToWorldPoint(mPosition));

i++;

var bc : BoxCollider = lineGo.AddComponent(BoxCollider);

bc.transform.position = lineRenderer.transform.position;

bc.size = new Vector3(0.1f, 0.1f, 0.1f);

bc.isTrigger=true;

}

if(touch.phase == TouchPhase.Ended)

{

lineRenderer.SetVertexCount(0);

i = 0;

var lineColliders : BoxCollider[] = lineGo.GetComponents(BoxCollider);

for(var b : BoxCollider in lineColliders)

{

Destroy(b);

}

}

}

}

* **Splash.js //To initiate splash/explosion effect when user swipes pots/bomb**

#pragma strict

private var randomAlpha: Color;

private var currentAlpha: float;

function Start () {

randomAlpha=new Color(1,1,1,Random.Range(0.3f,0.5f));

gameObject.renderer.material.color=randomAlpha;

InvokeRepeating("ReduceAlpha",0.3f,0.3f);

}

function Update () {

}

function ReduceAlpha(){

currentAlpha=gameObject.renderer.material.color.a;

if(gameObject.renderer.material.color.a<=0.1f)

{

Destroy(gameObject);

}

else{

gameObject.renderer.material.color=new Color(1,1,1,currentAlpha-0.1f);

}

}

* **Redolevel.js //To relode level again**

#pragma strict

function OnGUI()

{

if(GUI.Button(Rect((Screen.width/2)-60,100,120,50),"Try Again"))

{

Application.LoadLevel(Application.loadedLevel);

}

}

**5.1.3 LEVEL 3 CODING**

**//Carraige.js //To add forward force to Carriage**

#pragma strict

private var speed:float;

function Start () {

speed=1.0f;

}

function Update () {

transform.Translate(speed,0,0);

}

**//Timer1.js //To enable/disable Timer and display “GameOver”/”YouWin” Text**

#pragma strict

public TimeText: GUIText;

public ScoreText:GUIText;

public TotalText:GUIText;

public Totalt:GUIText;

public var alertReference : GameObject;

public var youWT : GameObject;

private var rdl : redolevel;

function Start () {

Screen.sleepTimeout = SleepTimeout.NeverSleep;

Screen.orientation=ScreenOrientation.LandscapeLeft;

Camera.main.GetComponent(AudioSource).Play();

TotalText.enable=false;

Totalt.enable=false;

rdl=Camera.main.GetComponent(redolevel);

rdl.enabled=false;

Time.timeScale=1;

InvokeRepeating("ReduceTimer",1,1);

}

function Update () {

}

function ReduceTimer() {

if(TimeText.text=="1")

{

Time.timeScale=0;

if(int.Parse(ScoreText.text)>=20)

{

Instantiate(youWT,new Vector3(0.5f,0.5f,0),transform.rotation);

TotalText.text=(int.Parse(ScoreText.text)\*10).ToString();

TotalText.enabled=true;

Camera.main.GetComponent(AudioSource).Stop();

}

else{

Instantiate(alertReference,new Vector3(0.5f,0.5f,0),transform.rotation);

rdl.enabled=true;

Camera.main.GetComponent(AudioSource).Stop();

}

}

TimeText.text=(int.Parse(TimeText.text)-1).ToString();

}

**//Opponent.js**

#pragma strict

private var speed:float;

public ScoreText:GUIText;

function Start () {

speed=1.0f;

}

function Update () {

speed=Random.Range(1.0f,5.0f);

transform.Translate(speed,0,0);

if(Input.touchCount = 1)

{

Instantiate(arrow,new Vector3(GameObject.x,GameObject.y,GameObject.z),Quaternion.Identity);

}

}

function OnTriggerEnter(other : Collider)

{

if (other.tag == "Arrow" && gameObject.tag=="Opponent")

{

ScoreText.text+=10;

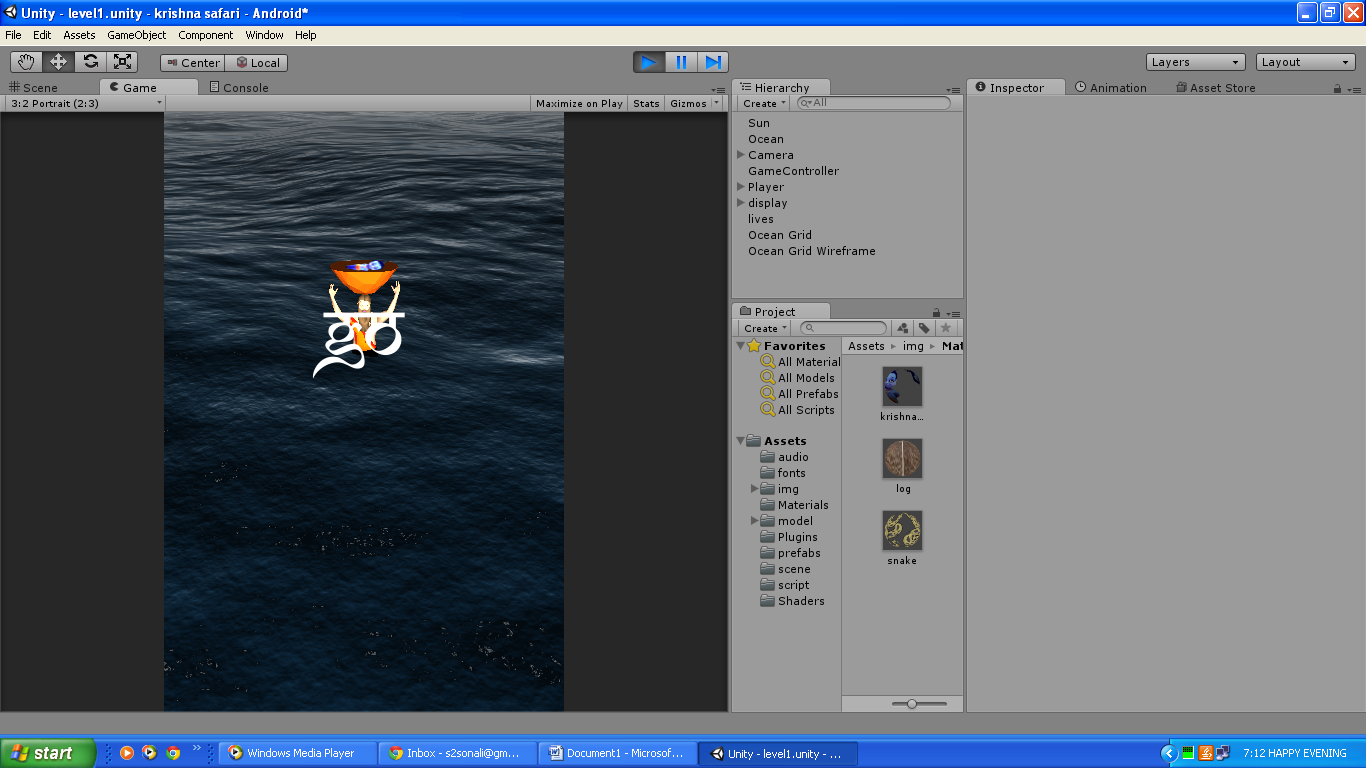
return;

}

}

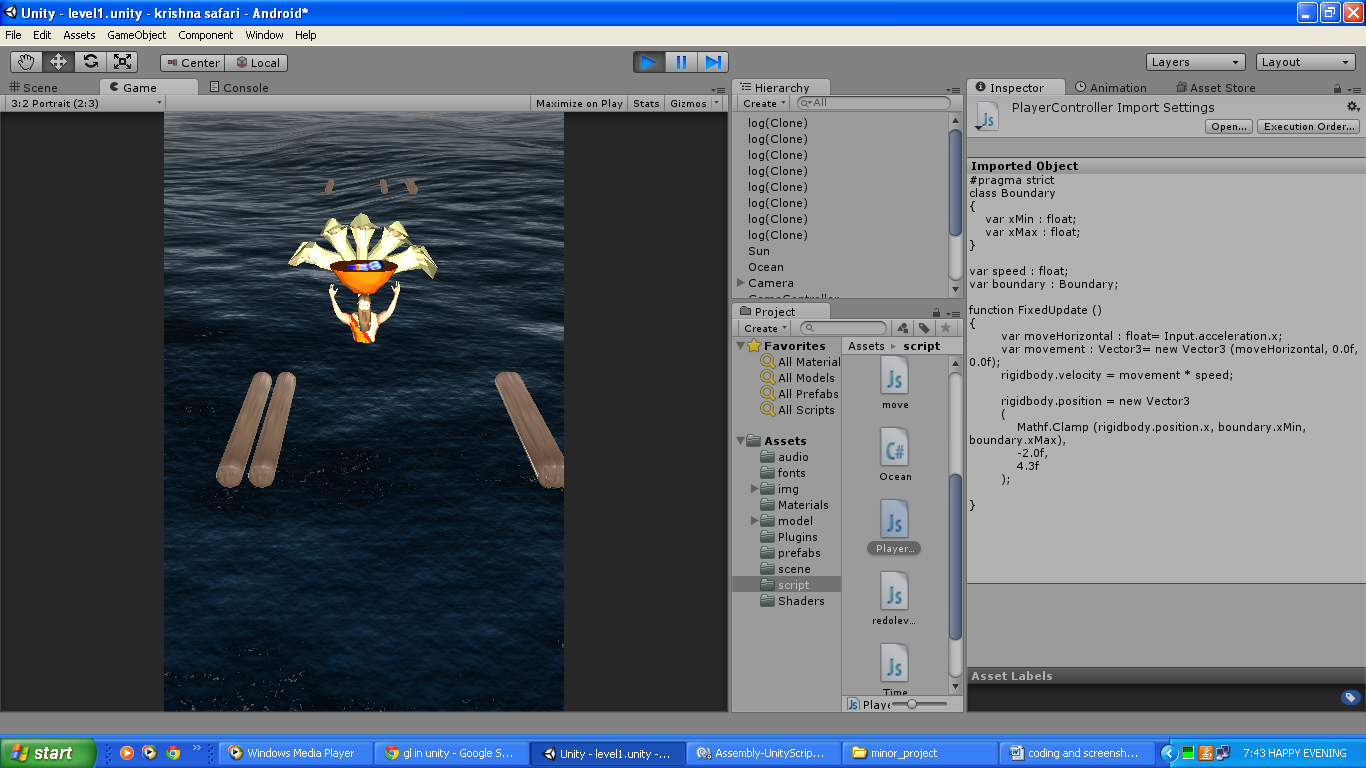
**5.2 SCREENSHOTS**

**5.2.1 LEVEL 1 SCREENSHOTS**

The initial screen will consist of Vaasudeva surrounded by the river as shown in fig.5.1

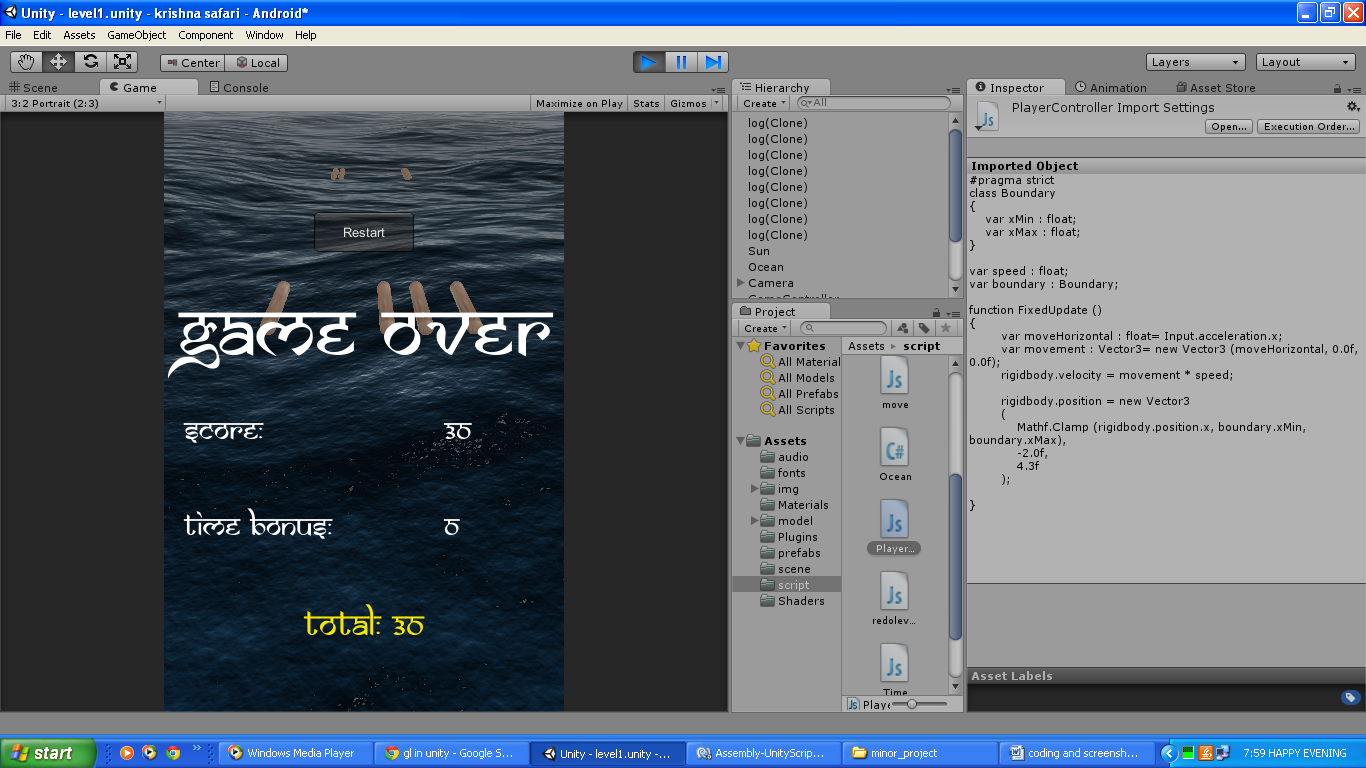
**Fig 5.1 Start Screen- level 1**

When a user collects the snake ball, a snake appears on the top of Vaasudeva as shown in fig. 5.2



**Fig 5.2 Sheshnaga**

If player hits any of the logs, game over screen will appear as shown in fig 5.3



**Fig 5.3 Game Over Screen-level 1**

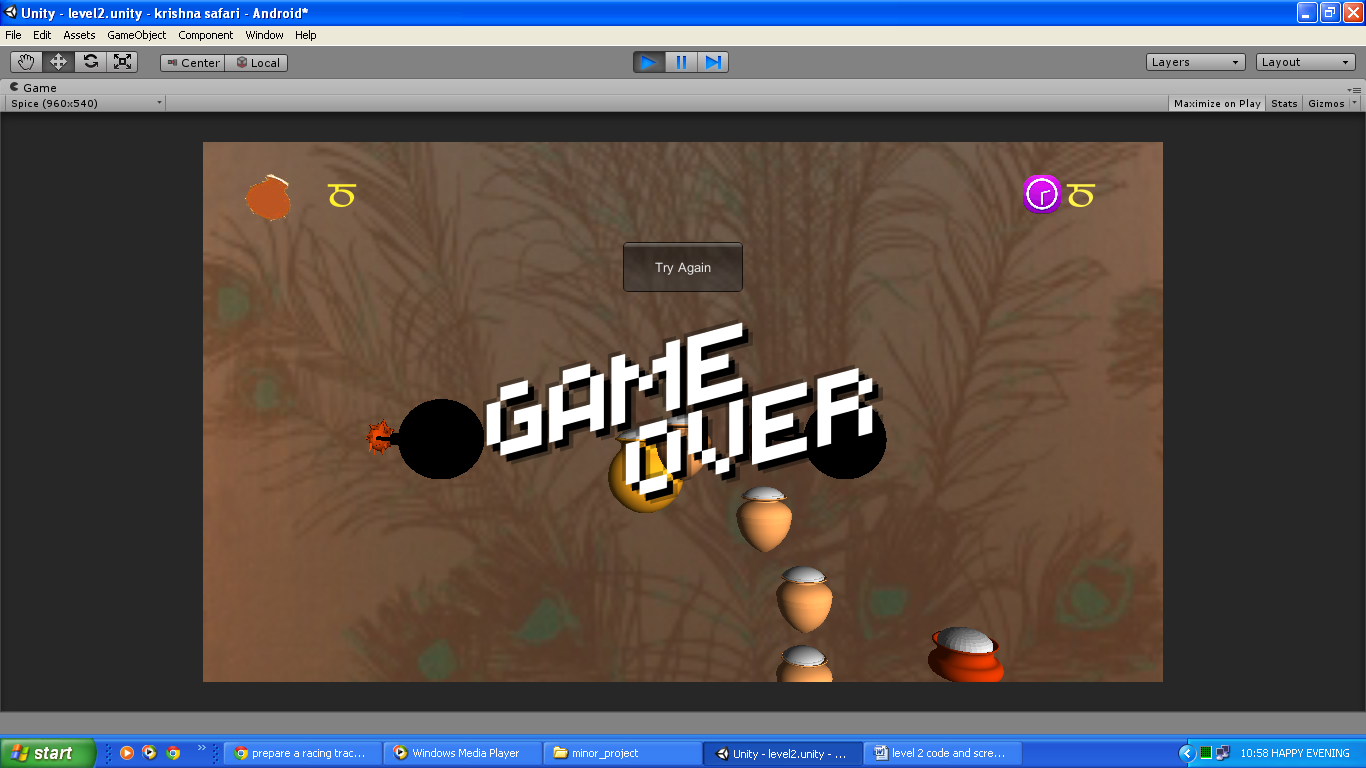
**5.2.2 LEVEL 2 SCREENSHOTS**

The initial screen for level two will consist of pitchers filled with butter as shown in fig.5.4



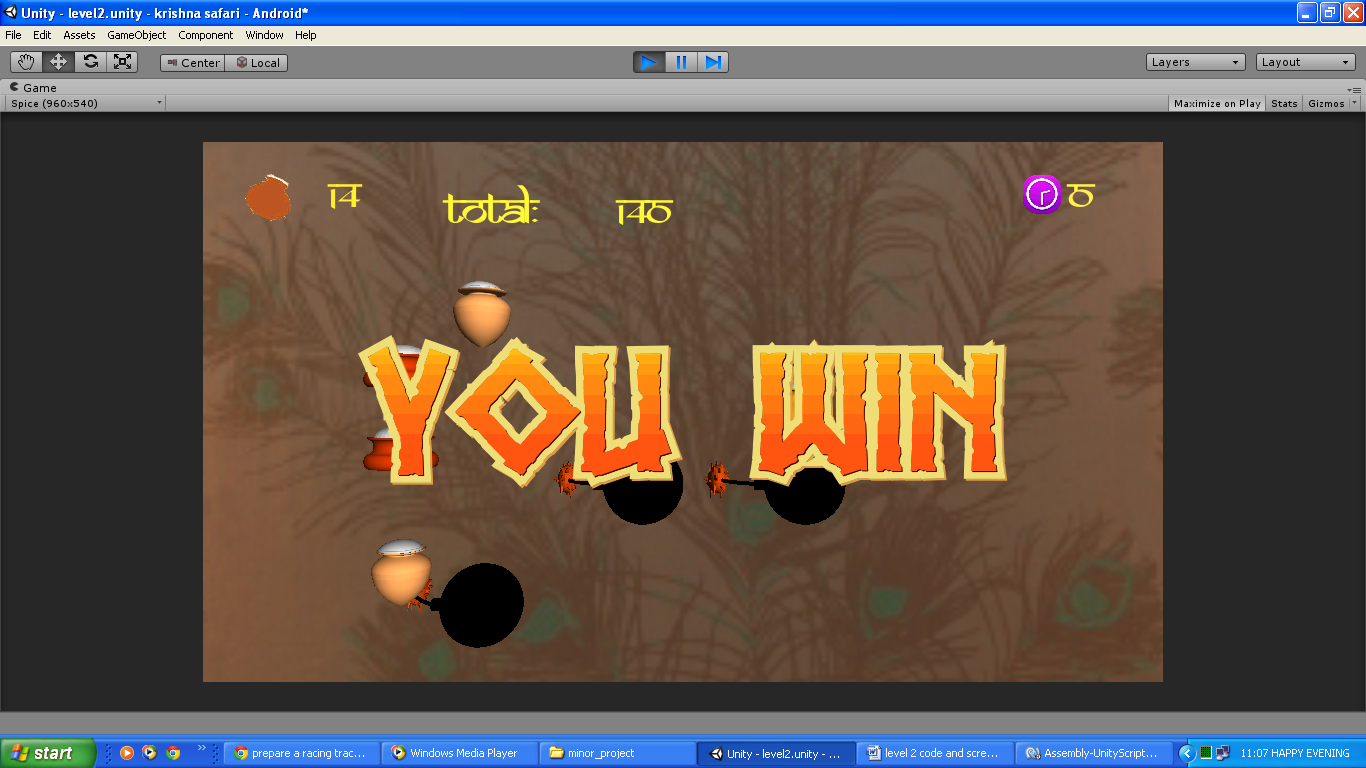
**Fig 5.4 start screen- level 2**

Game over screen will appear if a player swipes on a bomb as shown in Fig 5.5

****

**Fig 5.5 Game Over Screen- level 2**

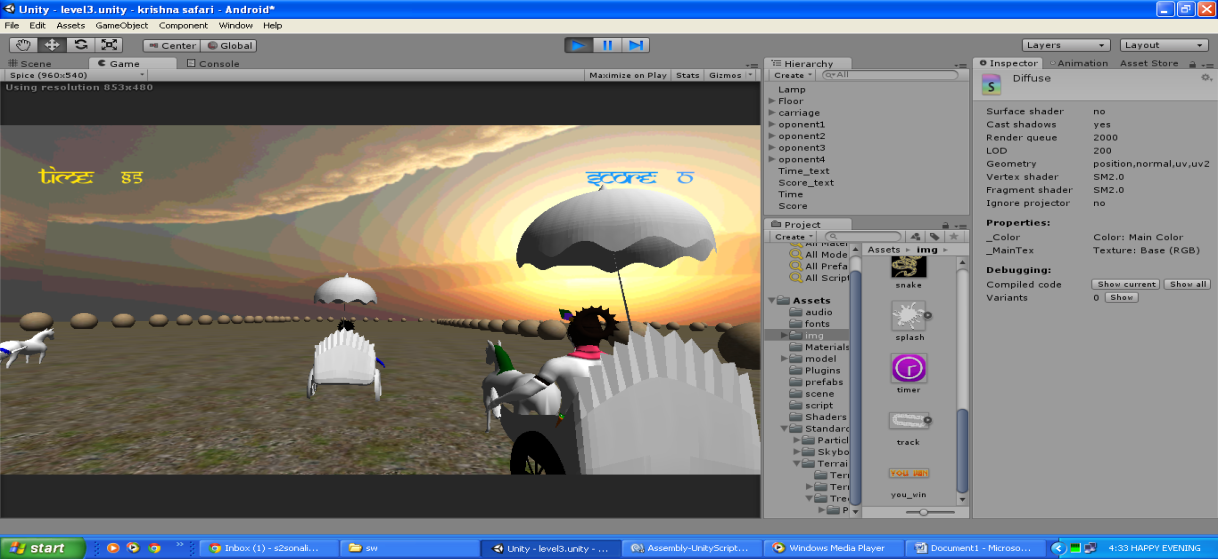
Level 2 winning screen will appear if a player completes level 2 as shown in Fig. 5.6

****

**Fig 5.6 win screen-level 2**

**5.2.3 LEVEL 3 SCREENSHOTS**

Level three starts up screen is shown in fig. 5.7

****

**Fig 5.7 Start Screen-level 3**

If the player is not able to win the race, game over screen will come as shown in fig. 5.8

****

**Fig 5.8 Game over Screen-level 3**

You win screen will appear if a player wins level 3 as shown in fig 5.9

****

**Fig 5.9 You Win Screen-level 3**

CHAPTER 6

CONCLUSIONS

For concluding the game, current scenario and ideology of the people were studied and then the entire game story was plotted. A feasibility study is conducted on various aspects to led a conclusion that the product is technically, economically and legally feasible. The characters and elements were made in Blender and Makehuman.

The designs of the characters are based on the various incidences of Lord Krishna’s Life. Unity 3D software is used for conceptualizing the scenes for the game. Unity’s animation window is used to add animation to all the characters and elements.

The scripts are added to the scenes using unity’s monoscripting feature. Unity’s JavaScript is used for scripting. The script is added for the player’s action and elements movement. To make the game working, all the scenes so made are connected together to attract the player’s attention and representing the whole saga of Lord Krishna’s life.

The game is available in .apk format and can be installed in the mobile phones supporting android version 2.3 and above.

CHAPTER 7

SUGGESTIONS FOR

FURTHER STUDY

There are many possibilities as to how the mobile version of the game ***Krishna Safari*** can be taken forward. Lord Krishna’s life was full of motivational incidences. To further enhance the project more levels can be added to the game showcasing various scenarios is Lord Krishna’s life.

Also, the game can be connected to a centralized database so that players all over the world can store their scores on a central database and could compete for the highest scores with each other. This will encourage more and more players to play the game increasing the feeling of competitiveness among them.

Effort can also be made for storing sessions of the game, so that a player can continue the game, whenever he wants. He need not have to start from the beginning.

As for now, the game ***Krishna Safari*** is only designed for android users. To increase its popularity, the game can also be designed for IOS users.

Another possibility is to introduce small missions in the game, completing which will give user bonus points and a big multiplier. Carefully designing these missions would make the gaming experience very enjoyable.

BIBLIOGRAPHY

* <http://unity3d.com/learn/tutorials/modules/beginner/editor/game-objects-and-components>
* <http://unity3d.com/learn/tutorials/modules/beginner/editor/build-player-settings>
* <http://unity3d.com/learn/tutorials/modules/beginner/editor/the-game-view>
* <http://cgcookie.com/blender/lessons/3-modeling/>
* <http://cgcookie.com/blender/lessons/interface-and-navigation/>
* <https://www.youtube.com/watch?v=lb0EG02Xn50/Moddelling>-a-base-mesh
* <https://www.youtube.com/watch?v=KWC6Pw63BAY/unity>introduction
* <http://www.makehuman.org/blog/a_new_tutorial_an_outfit_for_maiko.html>
* <https://www.youtube.com/watch?v=m3SdEt5hDWI/creating>-a-lifelike-character
* <http://unity3d.com/learn>/rigging
* <http://www.digitaltutors.com/software/Unity-tutorials>
* <http://www.ishafoundation.org/blog/yoga-meditation/history-of-yoga/krishna-stories/>
* <http://www.123janmashtami.com/lord-krishna/shri-krishna-and-kansa.html>
* <http://indianmythology.com/finish/seestory.php?storyID=77>
* <http://fonthindi.blogspot.in/2013/10/font-that-look-like-hindi.html>
* <http://www.bhagavad-gita.org/>krishnaLife