



Department of Computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,

Mathura – 281406

Declaration

We hereby declare that the work which is being presented in the Mini Project “**Title - FIGHTKINIGHT**”, in partial fulfillment of the requirements for Mini-Project LAB, is an authentic record of our own work carried under the supervision of **Anshu Chaudhary , Asst. Prof., GLA University, Mathura.**

Signature of Divyansh

Name of Candidate: Divyansh Gupta

Roll. No: 151500193

Course: B-tech

Year: 3rd

Semester: V

Signature of Lekhraj

Name of Candidate: Lekhraj Singh

Roll. No : 151500275

Course: B-tech

Year: 3rd

Semester: V



Department of Computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

CERTIFICATE

This is to certify that the project entitled “***FIGHTKINIGHT***” carried out in Mini Project – II Lab is a bonafide work done by ***DIVYANSH GUPTA (151500193)*** and ***LEKHRAJ SINGH (151500275)*** and is submitted in partial fulfillment of the requirements for the award of the degree Bachelor of Technology (Computer Science & Engineering).

Signature of Supervisor:

Name of Supervisor:

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Mini Project undertaken during B. Tech. Third Year. This project in itself is an acknowledgement to the inspiration, drive and technical assistance contributed to it by many individuals. This project would never have seen the light of the day without the help and guidance that we have received.

*Our heartiest thanks to **Dr. (Prof). Anand Singh Jalal**, Head of Dept., Department of CEA for providing us with an encouraging platform to develop this project, which thus helped us in shaping our abilities towards a constructive goal.*

*We owe special debt of gratitude to **Anshu Chaudhary**, Asst. Prof. Department of CEA, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. He has showered us with all his extensively experienced ideas and insightful comments at virtually all stages of the project & has also taught us about the latest industry-oriented technologies.*

*Acknowledgement to any other faculty member **Pankaj Sharma** Asst. Prof.*

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind guidance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

AMIT VERMA

AKSHAY CHAUHAN

Abstract

The proposed game will be a computer game that people can play on computers and smart phone . Players will be able to play the first person controller game in the single player style where they have to advance in the game by killing all the enemies and finally have to fight with the boss of all enemies. The game will allow only single player to play at a time. The game deals with artificial intelligence where enemies use their own intelligence to attack the player while he is playing. They will also follow the player so there is no chance of player to run from them. Unless or until player kills the boss he can't win. Due to the nature of the game, the graphics will be in 3D and offer a layout and makes people feel that they are in real world fighting zone.

In order to make the game interactive it is imparted with various animations that gives a good look and feel to the user. While fighting both the player and enemy can use special attacks that are designed to cause more damage to the other. Other than that player can heal himself using certain controls used in game .

The Terrain of the game is designed in 3D form in order to impart good looks to the game. Terrain represents an island where player can control his knight in order to fight with the skeleton enemies. There are various blocks on Terrain where Knight can go for instant power-up and can use his special thunder and fire attacks in order to kill his enemies.

Table of Contents

Declaration	ii
Certificate	iii
Acknowledgments	iv
Abstract	v
Table of Contents	vi
1. Introduction (This chapter must describe introduction about your project)	vii
1.1 Motivation and Overview	vii
1.3 Objective	viii
2. Software Requirement Analysis	ix
2.1 Define the problem	ix
2.2 Define the modules and their functionalities (SRS)	ix
3. Software Design	X
3.1 Use Case Diagram.....	x
3.2 Control Flow Diagram.....	xi
4. Testing	xii
4.1 Controls Testing.....	xii
4.2 Module Integration Testing.....	xii
5. Implementation and User Interface	xiii
5.1 Game Object Knight.....	xiii
5.2 Game Object Enemy.....	xiv
5.3 Front Page.....	xv
5.4 Game In Play Mode.....	xvi
5.5 Game View When Player Attacks.....	xvii
5.6 Enemy attack view.....	xviii
5.7 Game View On Unity.....	xix
References/Bibliography	xxii
6. Appendices	

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION AND OVERVIEW:

Nowadays many people have access to computers on a wide basis and a large number of children play computer games. Playing computer games can develop children's cognitive skills. Many popular games require abstract and high level thinking skills in order to win, skills that may not be taught at school. For example, children need to follow instructions, solve complex problems and use logic in many of the games that are currently popular. Such experience will be beneficial to a child's progression into an adult.

The above paragraph clearly depicts the usability of games , games are not only beneficial for the users but also the developers because they have to think in a broader dimension and try to engulf all their imaginations on one place . They deal with virtual entities that gives them a good understanding of object oriented programming.

Also one of the basic reason for making a project on game is that in upcoming 2018 Global Games Market Report, the researchers estimate that the 2018 games market will jump from \$93.6 billion in 2017, to **\$191.5 billion** in 2018. All this facts also gives a broader overview of what gaming industry is at present.

Later on in future it is predicted that all kinds of cutting edge technologies are going to be used by gaming industry like VR , AR, Cloud Computing , Artificial Intelligence.

1.2 OBJECTIVE:

The main objective of our project is to facilitate people to play an interactive game and develop their cognitive skills and try to use their minds as creatively as possible . People can play this game in order to relax their minds after freaked out by day to day work . Our game enemies have their own intelligence to it is very hard to beat them so people have to think creatively to win the game.

1.3 SUMMARY OF SIMILAR APPLICATIONS:

Although the idea and the user interface of our game is unique but there are various games in the market having more interactive user-interface and gaming experience.

"Temple Run", "Clash of Clans", "Modern Combat 5" are the games that are very trending and are having millions of people playing it. All these games give a very high quality experience to the users, with various additional features like making groups all around the world , chatting with people , battling with different groups of the world. All these games are masters of the virtual world.

Playing these games has become a passion all around the world. All these games have become million dollar company at present and are now working as a leading gaming company and are successfully launching new games every year.

CHAPTER 2:

SOFTWARE REQUIREMENT ANALYSIS

2.1 DEFINE THE PROBLEM:

Design a first person controller game with a 3D interactive terrain where player controls the knight and has to fight with the enemies with their own intelligence in order to win the game.

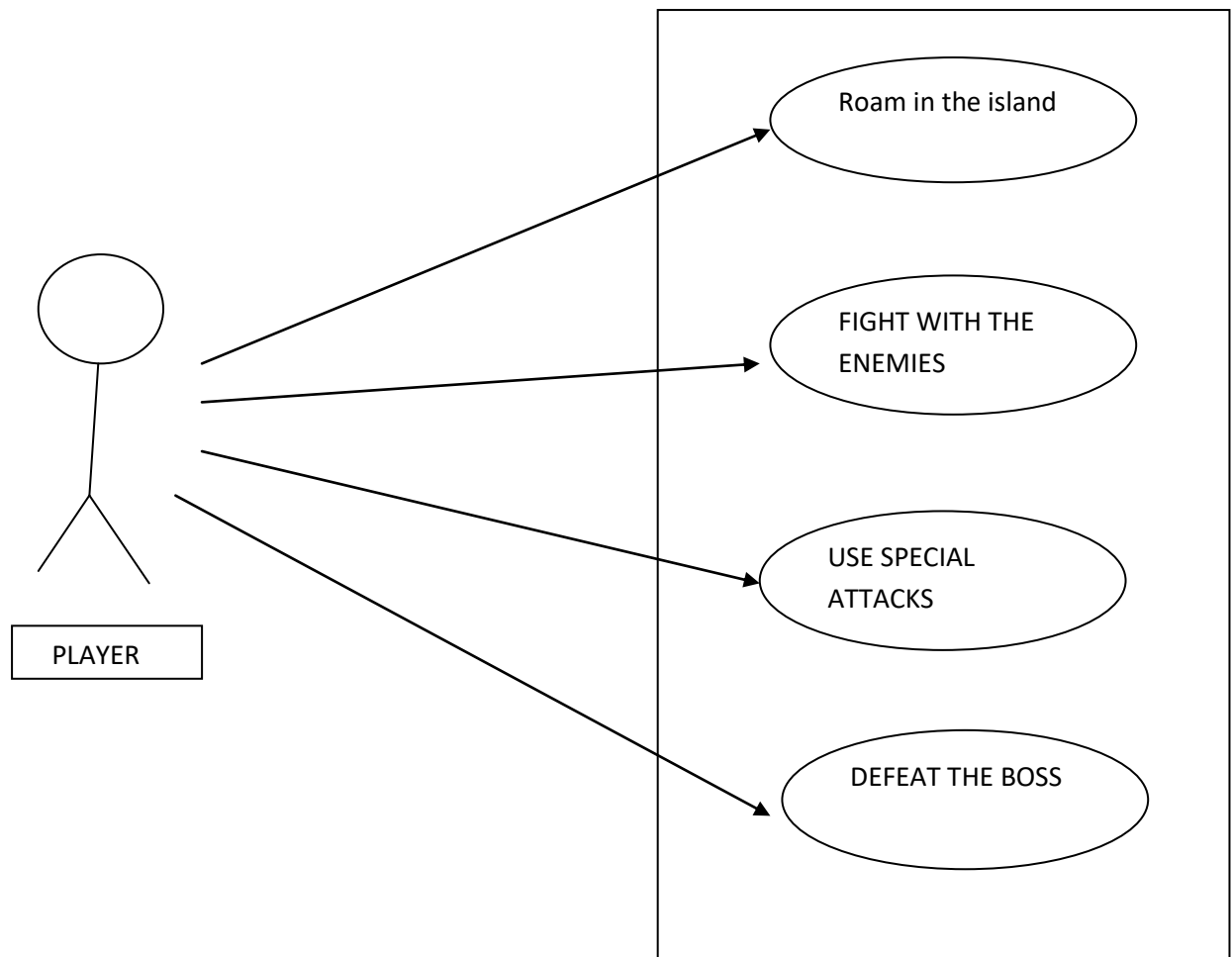
2.2 MODULES AND THEIR FUNCTIONALITY:

- **PLAYER MOVE:** Controls the movement of the player.
- **PLAYER ATTACK:** Maps all the attacks of the player , what type of attack player wants to make.
- **PLAYER ATTACK EFFECTS:** This module triggers the effects that are to be seen on screen when a player makes a particular attack.
- **PLAYER HEALTH:** Maps the current health of the player . If health becomes 0 knight dies.
- **ENEMY CONTROL:** Maps all attacks of enemies.
- **ENEMY HEALTH:** Maps the current health of all the enemies.
- **BOSS CONTROL:** Maps all the attacks of Boss.
- **BOSS SPECIAL ATTACKS :** It controls the special attacks of the boss.
- **BOSS STATE CHECKER:** Returns the current status of the boss.
- **CAMERA FOLLOW:** Control movements of camera. Camera follows the knight.
- **MOUSE POINTER:** This module shows the movement point of the player.
- **DESTROY AFTER TIMER:** Controls time for which special attacks will last.
- **FIRE TORNADO MOVE:** Controls the special attack "fire tornado:

CHAPTER 3:

SOFTWARE DESIGN:

3.1 USE CASE:



3.2 CONTROL FLOW:

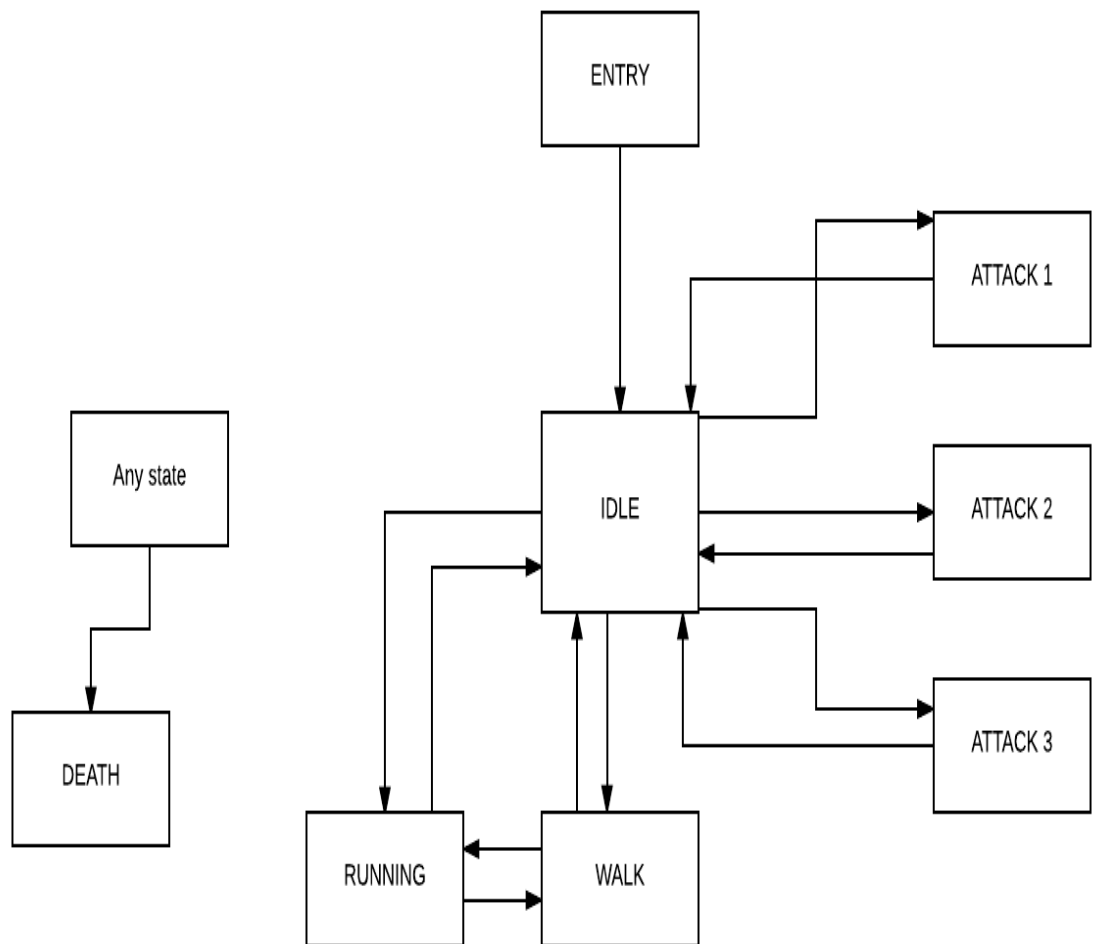


Diagram showing the flow to control between different modules for the game object(knight).

CHAPTER 4

TESTING:

4.1 CONTROLS TESTING:

- POINT: On pointing and clicking the mouse pointer on any part of terrain player moves.
- CAMERA CONTROL: Camera follows the movement of the player.
- ATTACK UNIT: On pressing all the attack keys from 1,2,3,4,5 player attacks.
- ANIMATIONS: All animations gets triggered according to their events.
- ENEMY ATTACK: All enemies attack to the player and advance towards knight according to their intelligence.
- HEALTH: All the game objects health is decreased when attacked.

4.2 MODULES AND THEIR FUNCTIONALITY TESTING:

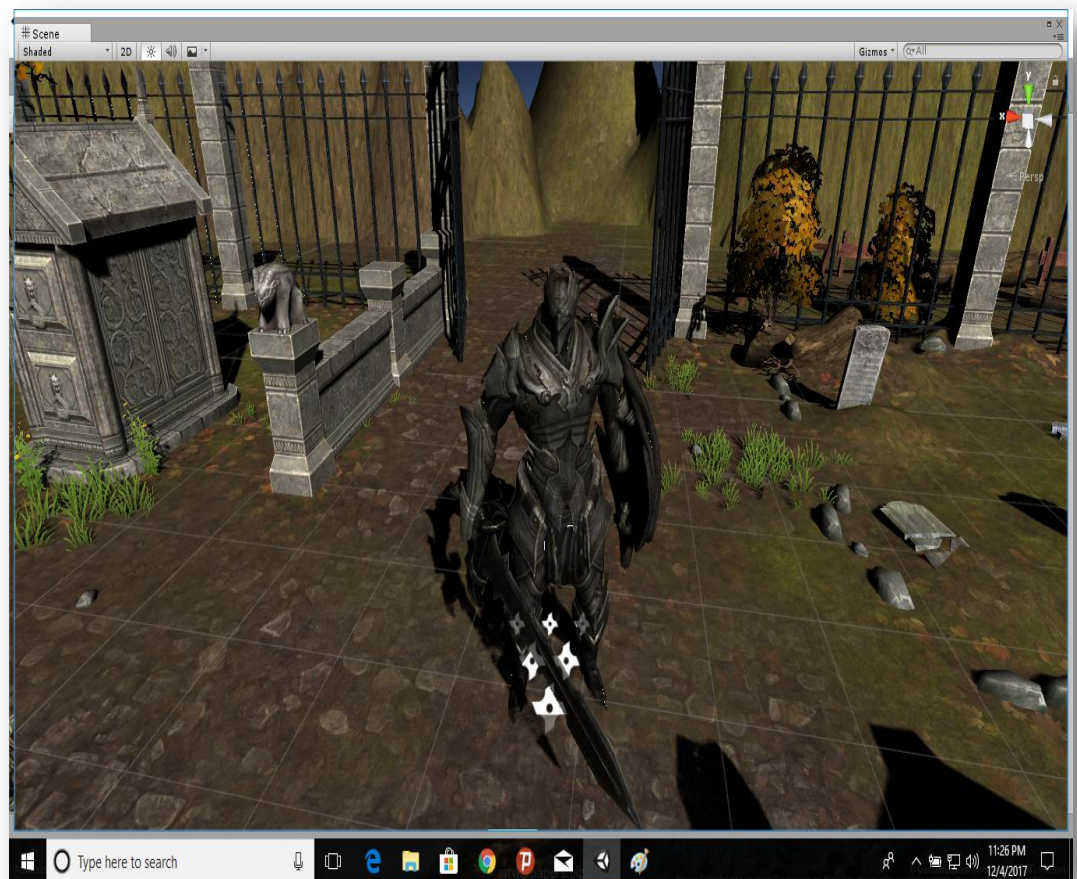
- PLAYER MOVE: Player moves as per the requirements.
- PLAYER ATTACK: Maps all the attacks of the player , what type of attack player wants to make.
- PLAYER ATTACK EFFECTS: This module triggers the effects that are to be seen on screen when a player makes a particular attack.
- PLAYER HEALTH: Maps the current health of the player . If health becomes 0 knight dies.
- ENEMY CONTROL: Maps all attacks of enemies.
- ENEMY HEALTH: Maps the current health of all the enemies.
- BOSS CONTROL: Maps all the attacks of Boss.
- BOSS SPECIAL ATTACKS : It controls the special attacks of the boss.
- BOSS STATE CHECKER: Returns the current status of the boss.
- CAMERA FOLLOW: Control movements of camera. Camera follows the knight.
- MOUSE POINTER: This module shows the movement point of the player.
- DESTROY AFTER TIMER: Destroy the special attacks after 2 sec.
- FIRE TORNADO MOVE: Fires the tornado when requested.

CHAPTER 5

IMPLEMENTATION AND USER INTERFACE:

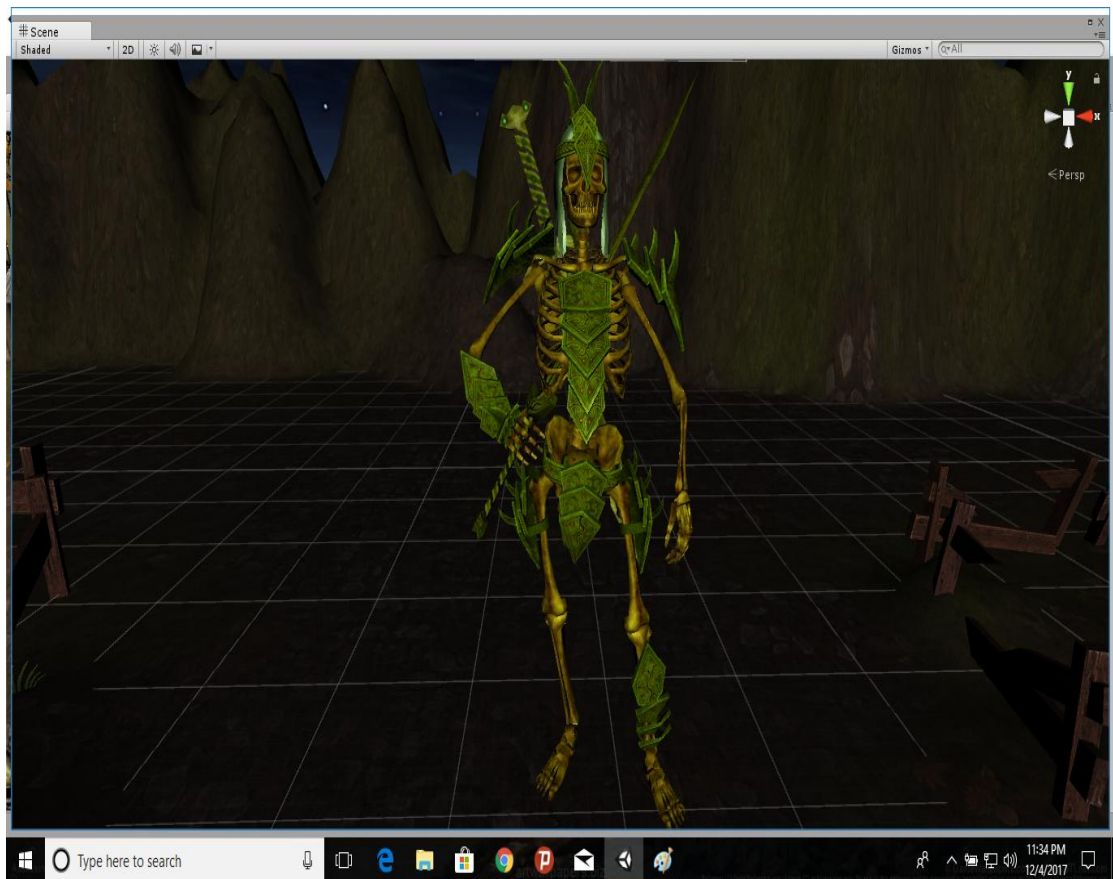
5.1 GAME OBJECT KNIGHT:

This is the game object which user controls and uses it to fight with enemies.



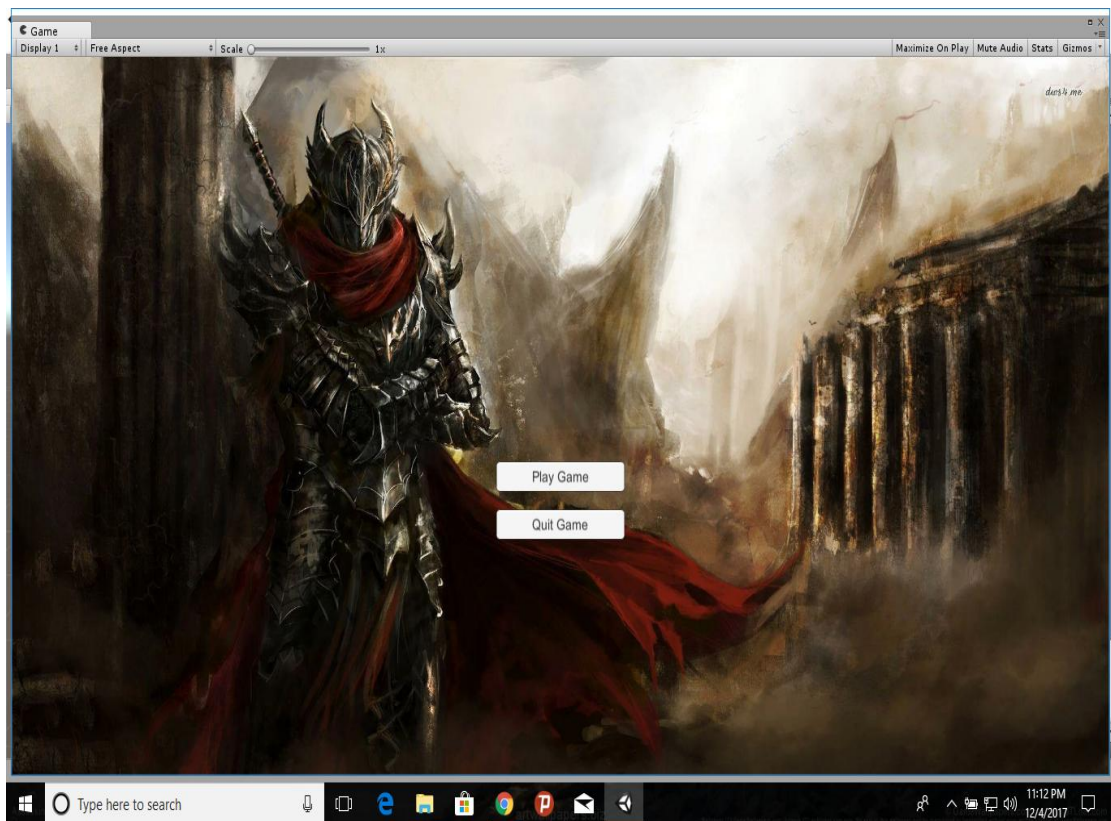
5.2 ENEMY:

This is the game object having its own artificial intelligence used to attack the knight.



5.3 FRONT PAGE:

Page which opens when we start the game.



5.4 GAME IN PLAY MODE(Camera View):

This is what game looks like in play mode.



5.5 GAME VIEW WHEN PLAYER ATTACKS:

This is what game looks like in play mode when player uses special attacks.



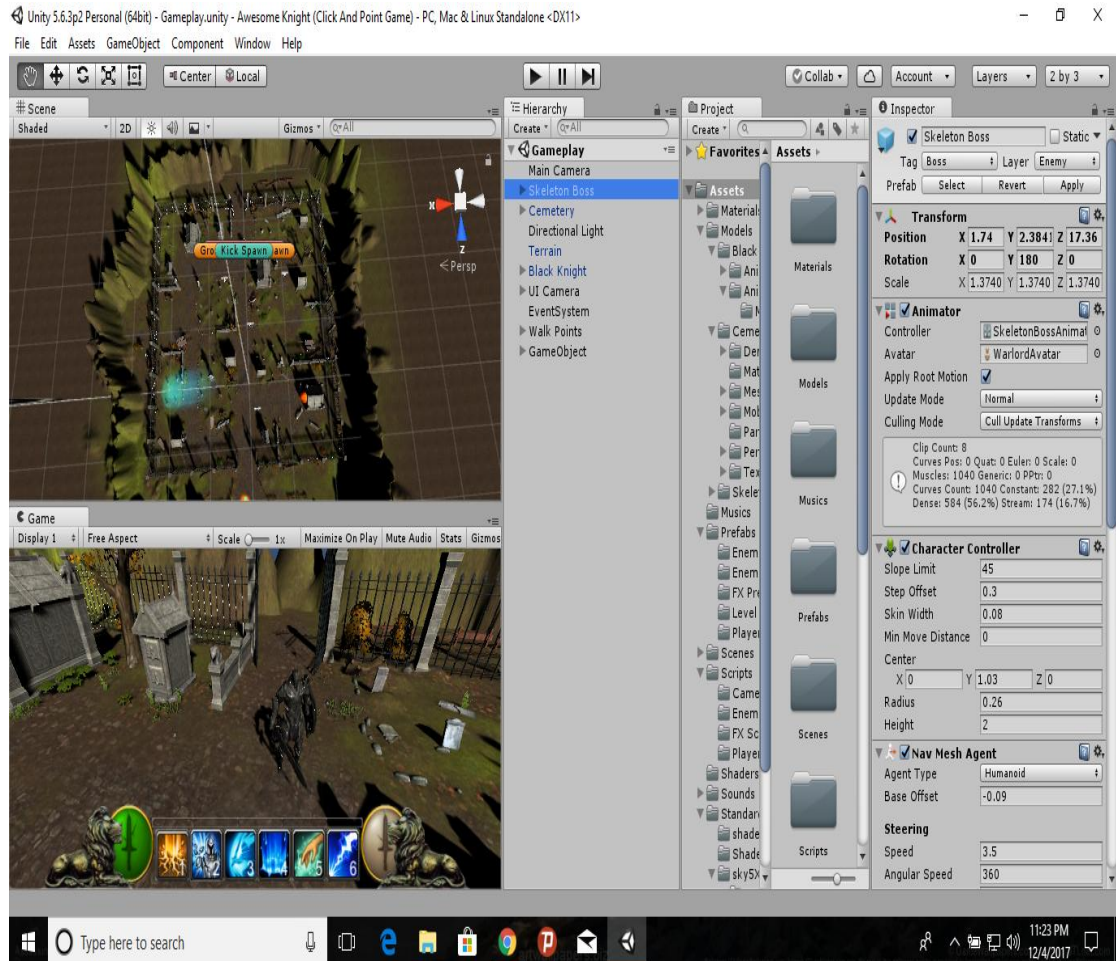
5.6 GAME VIEW WHEN ENEMY APPROACH:

This is what game looks like in play mode when enemies approach.



5.7 GAME VIEW ON UNITY:

This is what game looks like in unity from top view.



REFERENCES/BIBLIOGRAPHY:

* UNITY : UNITY USER MANUAL (2017.2)

* www.YOUTUBE.com

* www.GOOGLE.com

* GAMING TUTORIALS ON UNITY OFFICIAL.

* LUCID CHART

* TUTORIALS POINT(C#)

* JAVATPOINT(C#)

* UNITY LEARN MODULES

CHAPTER 6

APPENDICES:

* PLAYER HEALTH SCRIPT:

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class PlayerHealth : MonoBehaviour {

    public float health = 100f;

    private bool isShielded;

    private Animator anim;

    private Image health_Img;

    void Awake() {
        anim = GetComponent<Animator> ();

        health_Img = GameObject.Find ("Health Icon").GetComponent<Image>();
    }

    public bool Shielded {
        get { return isShielded; }
        set { isShielded = value; }
    }

    public void TakeDamage(float amount) {
        if (!isShielded) {

            health -= amount;

            health_Img.fillAmount = health / 100f;

            if (health <= 0f) {
                anim.SetBool ("Death", true);

                if (!anim.IsInTransition (0) && anim.GetCurrentAnimatorStateInfo
                (0).IsName ("Death")
                && anim.GetCurrentAnimatorStateInfo (0).normalizedTime >= 0.95) {
                    // PLAYER DIED
                    // DESTROY PLAYER
                }
            }
        }
    }
}
```

```

}
}
}
}

public void HealPlayer(float healAmount) {
health += healAmount;

if (health > 100f)
health = 100f;

health_Img.fillAmount = health / 100f;
}
}

```

*** CAMERA MOVEMENT SCRIPT:**

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class CameraFollow : MonoBehaviour {

    public float follow_Height = 8f;
    public float follow_Distance = 6f;

    private Transform player;

    private float target_Height;
    private float current_Height;
    private float current_Rotation;

    void Awake () {
        player = GameObject.FindGameObjectWithTag ("Player").transform;
    }

    void Update () {
        target_Height = player.position.y + follow_Height;

        current_Rotation = transform.eulerAngles.y;

        current_Height = Mathf.Lerp (transform.position.y, target_Height,
0.9f * Time.deltaTime);

        Quaternion euler = Quaternion.Euler (0f, current_Rotation, 0f);

        Vector3 target_Position = player.position - (euler *
Vector3.forward) * follow_Distance;

        target_Position.y = current_Height;

        transform.position = target_Position;
        transform.LookAt (player);
    }
}

```

