DECLARATION

I hereby declare that the project entitled, “**Escape The Zombies**” is a bona fide work carried out by me i.e. **Tanisha Jeevan Soni** done at **Viva College (Virar West)**, has not been in any case duplicated to submit to any other university for the award of any degree.

The project is done in partial fulfilment of the requirements for the award of degree of **BACHELOR OF SCIENCE (COMPUTER SCIENCE)** to be submitted as final year semester 5 project as part of our curriculum.

Tanisha Jeevan Soni

(Signature)

ACKNOWLEDGEMENT

Apart from the efforts of myself, the success of the project depends largely on the encouragement and guidelines of many others. The gratification and joy that accompanies the successful completion of the project is incomplete without the humble and deep-felt expression of gratitude. So I take this opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project.

I would like to express a deep sense of gratitude to our Head of the department **Prof. Jagruti Raut** who modelled us both technically and morally for achieving greater success in life and also who gave us opportunity to work under her guidance and for her cordial support throughout the project.

I would like to extend my sincerest gratitude to **Prof. Priyanka Brid** my internal project guide**,** who has been source of motivation and inspiration and has helped in bringing out best in me and also for her constant guidance, supervision and support in completion of the project.

Further I also extend my heartiest thanks to my parents and friends for being with me and extending encouragement and co-operation throughout the project.

Table of Contents

1. INTRODUCTION

   1.1-Proposed System 4

2. REQUIREMENT SPECIFICATION

   2.1-Hardware 5

   2.2-Software 5

   2.3-Modules 5

3. GANTT CHART 6

4. UML DIAGRAMS

   4.1-Activity diagram 7

   4.2-Flow chart 8

   4.3-Sequence Diagram 9

5. TEST CASES 10

6. CODING & IMPLEMENTATION 11-13

7. FUTURE ENHANCEMENT 14

8. REFERENCE 14

INTRODUCTION

1.1- Propose system

Escape the Zombies is a fun and engaging 3D game that challenges players to think strategically and use their problem-solving skills to escape from the chasing zombies. The game combines elements of horror and survival to create a unique and immersive gameplay experience. The game is developed using the Unity game engine, which provides high-quality 3D graphics and gameplay.

The player character is chased by a group of zombies and must navigate through a maze-like environment in order to find an escape.

The game includes a range of power-ups and weapons that the player can collect to aid in their survival.

The sound effects and different types of zombies add to the realism of the game and make it a unique and exciting experience.

REQUIREMENT SPECIFICATION

2.1-Hardware requirement:

1. A processor with X64 architecture

2.2-Software requirement:

1. Unity - Unity3D is a powerful cross-platform 3D engine and a user friendly development environment. Easy enough for the beginner and powerful enough for the expert; Unity should interest anybody who wants to easily create 3D games and applications for mobile, desktop, the web, and consoles.

2. Visual studio - Visual studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code.

3. Windows 7 SP1+, 8, 10, 64-bit versions only. – operating system

2.3-Modules:

1. Player :- The player character is chased by a group of zombies and must navigate through a maze-like environment in order to find an escape

2. Zombies :- There are different types of zombies that will chase the player and have a set of great animations to it.

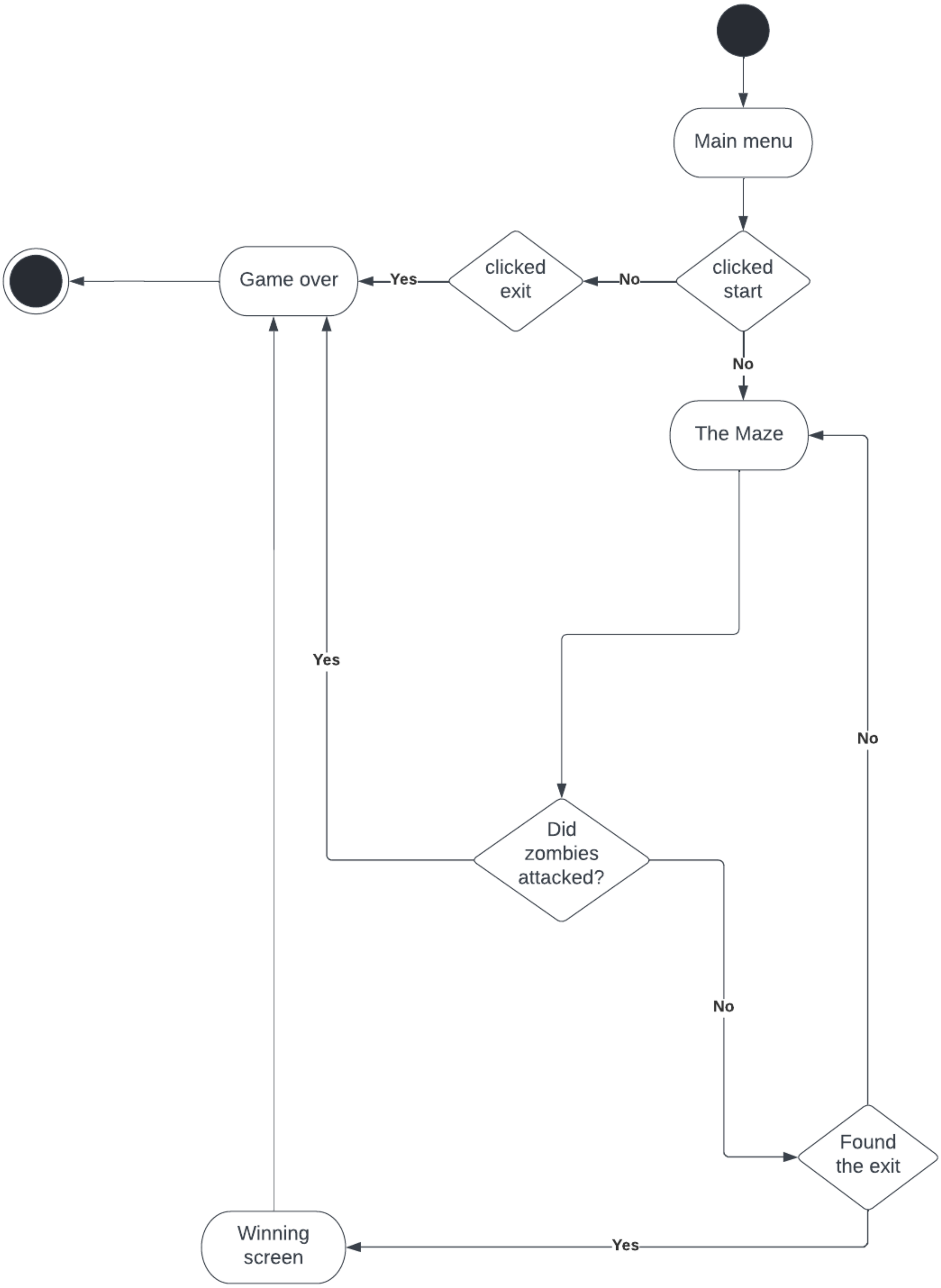
3. The Maze :- The maze is the environment built using Walls in which the player will have to find the correct path to exit.

4. Mini map :- The minimap provides the top view of the maze where the player is located which will help user to find the path and look if the zombies are close.

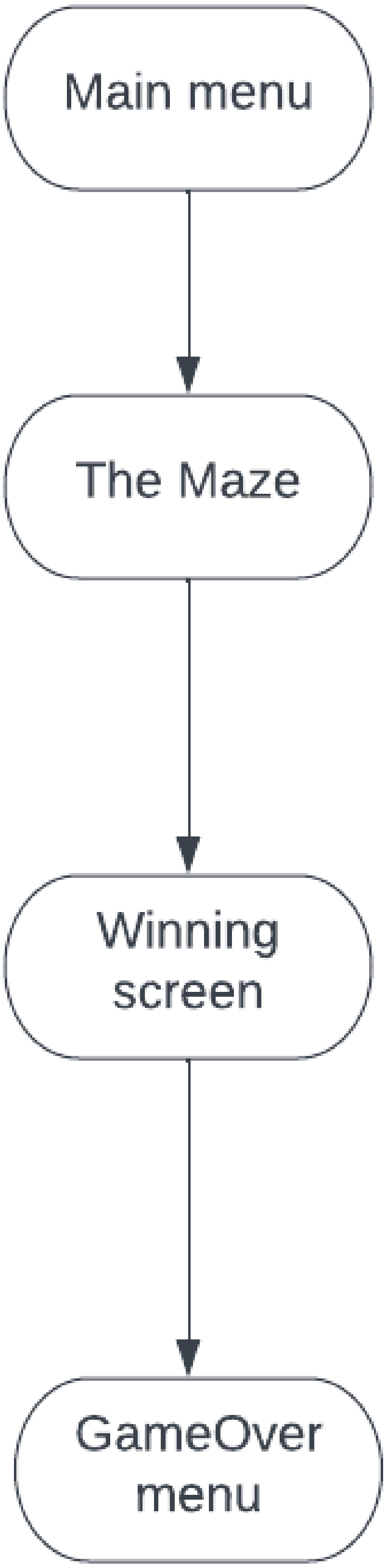
GANTT CHART

UML DIAGRAMS

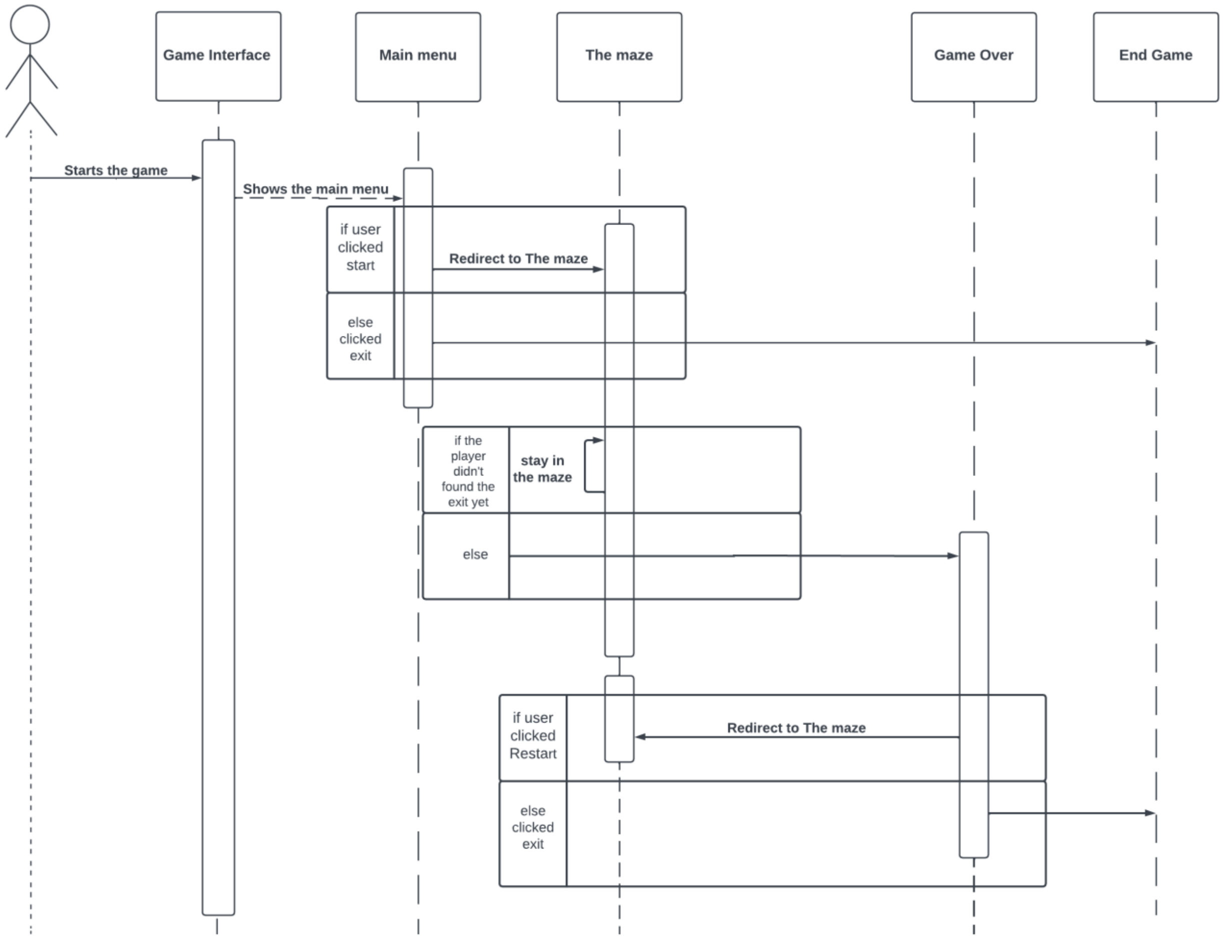
4.1-ACTIVITY DIAGRA



4.2-FLOW CHART



4.3-SEQUENCE DIAGRAM

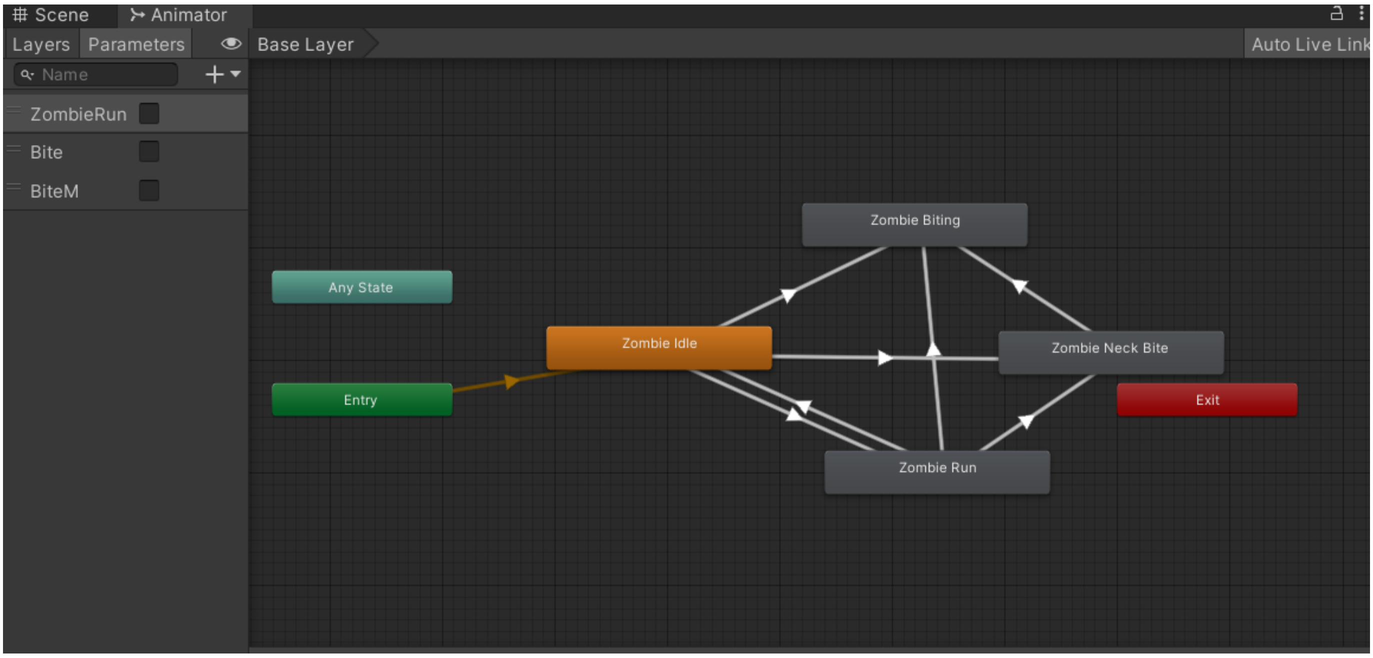


Test Case:

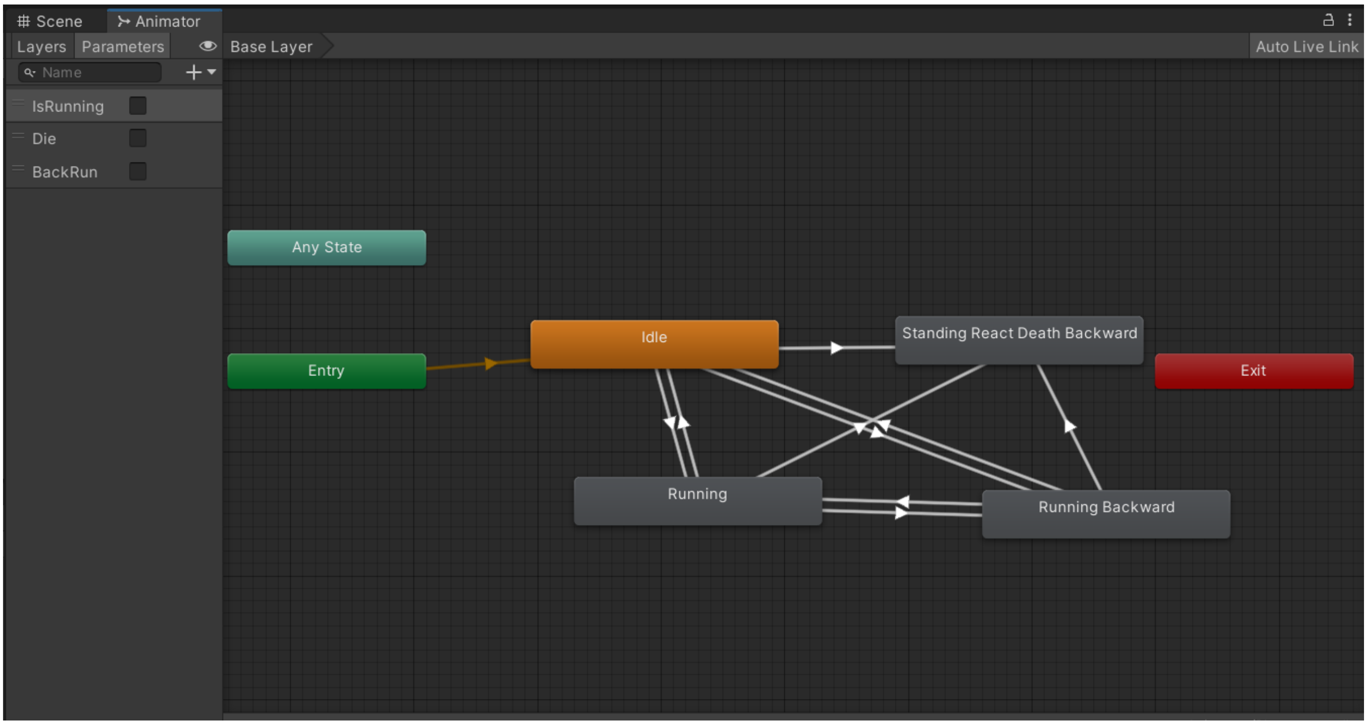
Sr No.	Action	Input	Expected Output	Actual Output	Test Result
Test-1	Start the game	Click on start Button	Redirect to Maze	Redirected to Maze	Pass
Test-2	Exit the game	Click on exit Button	End the Game	End the Game	Pass
Test-4	Game Over	Check if Zombie attacked	Redirect to Game Over menu	Redirect to Game Over menu	Pass
Test-5	Attack the player	Checks if the player collides with zombies	Attack the player	Zombies Attacks the player	Pass
Test-6	Winning screen Appears	Checks if the Player reaches/collided with the Exit	The winning screen will appear	The winning screen appears	Pass

CODING & IMPLEMENTATION

Zombie's Animator:



Player's Animator:



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

public class PlayerController : MonoBehaviour
```

```
{
    float speed = 50f;
    private Animator animator;
    private CharacterController c_controller;
    float z;
    Vector3 move;
    public AudioSource audioSource;
    public AudioClip footstep;
    public AudioClip backfoot;
    public bool gameover = false;

    //float jumpvelocity = 30f;
    float gravity = -19f;
    // Start is called before the first frame update
    void Start()
    {

        animator = GetComponent<Animator>();
        c_controller = GetComponent<CharacterController>();
    }

    // Update is called once per frame
    void Update()
    {
        if (gameover == false) {
            z = Input.GetAxis("Vertical");

            move = transform.forward * z;



            if (z == 1)
            {

                animator.SetBool("IsRunning", true);
            }
            else
            {
                animator.SetBool("IsRunning", false);
            }
            move.y += gravity * Time.deltaTime;
            c_controller.Move(move * speed * Time.deltaTime);
        }


        if (z == -1)
        {
            animator.SetBool("BackRun", true);
        }
        else
        {
            animator.SetBool("BackRun", false);
        }
    }
    void Footsound()
    {
        audioSource.clip = footstep;
        audioSource.Play();
    }
    void Backsound()
    {
        audioSource.clip = backfoot;
        audioSource.Play();
    }
    private void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.CompareTag("Zombies"))
        {
            Debug.Log("player collided");
            gameover = true;
            animator.SetBool("Die", true);
            c_controller.enabled = false;
            other.gameObject.GetComponent<CapsuleCollider>().enabled = false;

other.gameObject.GetComponent<Animator>().SetBool(other.gameObject.GetComponent<ZombieCh
asing>().anim_name, true);
            other.gameObject.GetComponent<ZombieChasing>().bite = true;
            other.gameObject.GetComponent<ZombieChasing>().anim_name = "BiteM";


}

}
```

FUTURE ENHANCEMENT

I will introduce new levels to make the game more interesting.

REFERENCES

https://docs.unity3d.com/Manual/index.html