Zombie Game: A Game made with Unity 3D

Prepared by
Sunny Kalariya(16IT040)
Vatsal Javia(16IT034)

Under the supervision of

Prof. Kamlesh Makwana

A Report Submitted to

Charotar University of Science and Technology

for Partial Fulfillment of the Requirements for the

Degree of Bachelor of Technology

in Information Technology

IT345 Software Group Project-II(5^{th} sem)

Submitted at

DEPARTMENT OF INFORMATION TECHNOLOGY

Chandubhai S. Patel Institute of Technology

At: Changa, Dist: Anand – 388421

October 2018

CERTIFICATE

This is to certify that the report entitled "Zombie Game: A Game made with Unity 3D" is a bonafied work carried out by **Mr. Sunny Kalariya (16IT040), Mr. Vatsal Javia (16IT034)** under the guidance and supervision of **Prof. Kamlesh Makwana** for the subject **Software Group Project-II (IT345)** of **5th** Semester of Bachelor of Technology in **Information Technology** at Faculty of Technology & Engineering – CHARUSAT, Gujarat.

To the best of my knowledge and belief, this work embodies the work of candidate **himself**, has duly been completed, and fulfills the requirement of the ordinance relating to the B.Tech. Degree of the University and is up to the standard in respect of content, presentation and language for being referred to the examiner.

Under supervision of,

Prof. Kamlesh Makwana
Assistant Professor
Dept. of Information Technology
CSPIT, Changa, Gujarat.

Prof. Parth Shah
Head & Associate Professor
Department of Information Technology
CSPIT, Changa, Gujarat.

Chandubhai S Patel Institute of Technology

At: Changa, Ta. Petlad, Dist. Anand, PIN: 388 421. Gujarat

ABSTRACT

Player is to survive through the level by destroying the zombie using gun. We have used C# language to prepare control and handling movement,animations and sound effects. An in depth report of scripting and Monodevelop class explained further The Infinite Survivor is created using Unity3D Game tool. It is a First person 3D Game where a in the report.

ACKNOWLEDGEMENT

I would like to thank the Almighty God without which this project would not have been possible. We hereby take this opportunity to thank each and every one who has helped us in creating and formulating this project. I thank our Head of Department **Prof. Parth D. Shah** who gave me this golden opportunity to do this project. We should also like to take this opportunity to express a deep sense of gratitude to our teacher **Prof. Kamlesh Makwana** for his cordial support, exemplary guidance, monitoring and constant encouragement. I am obliged to my friends, parents and importantly the faculties for their valuable guidance and co-operation during the period of this task. The blessing help and guidance was a deep inspiration to me.

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Project Summary :

Background: By Learning about the Unity tool for developing games,
We developed 3D game for Windows/Android platform.Here we explor the basics about Game development, and a few common problems faced in its development.

Problem Statement: Learning C# for coding in Unity3D as well as learning about game physics and animation may turn out to be time consuming and as its consequences, the development time gets shorted.

Approach: Unity3D is very popular and powerful software for making 3D games.It provides a versatile working space and is also supported by multiple platforms.As this tool is highly popular, it also provides a large responsive community. Therefore we will use Unity 3D for making this project.

During the making of this project, we have explore 3D tools and also learn more about textures,character physics,level objects and much more in the field of game development.

1.2 Scope:

This Project open gateway to various methods of learning graphics rendering,mathematics, physics and Game development. Also it provides good means of entertainment.

The hot trending topics of Virtual Reality and Augmented Reality further extend the scope of this project.

1.3 Objective:

The objective of this project is to demonstrate the knowledge and scope of Game Development using Unity 3D.

Chapter 2 System Analysis

2.1 User Characteristics

- Hobby of playing games, no specific knowledge about computers or software is required.

2.2 Tools & Technology

- Unity 3D version 2018.2.4f1
- C# text editor (Visual studio)

❖ HARDWARE REQUIREMENTS

- For smoothly use of software, high specific PC or Laptop

2.3 User System Requirements

- Android Version 6.0 or above
- 512MB RAM

Chapter 3 Implementation

3.1 Implementation Environment

- Single User
- GUI

3.2 Program / Modules Specification

- Environment setup
- Lighting setup
- Objectimovement and actions
- Camera setup
- Debugging

3.3 Coding Standards

- We have used Object Oriented approach for programming in this project.
- We have followed general coding conventions used in Object Oriented Programming(OOP) our project which include:
  - Lower camel casing for naming conventions.
  - Frequent commenting which helps us easily identify the work of Each and every statement of code, to understand the functioning of a script.
  - As we have used C# language for scripting, we use common object Object oriented programming terms like:

1. Class: to define a class for creating objects.
2. Public, private: access specifiers to variables.
3. Awake(), Update(), Start(): method names which are called during runtime to perform various tasks.
4. Void: return type to methods(return type can also be char, int, etc)
5. Float, bool, vector3: various data types available in C# language to Declare variables and store various values.

- Provide below is an example of approach explained above:
Screenshot of Code

EnemySpawner — zombiePrefab

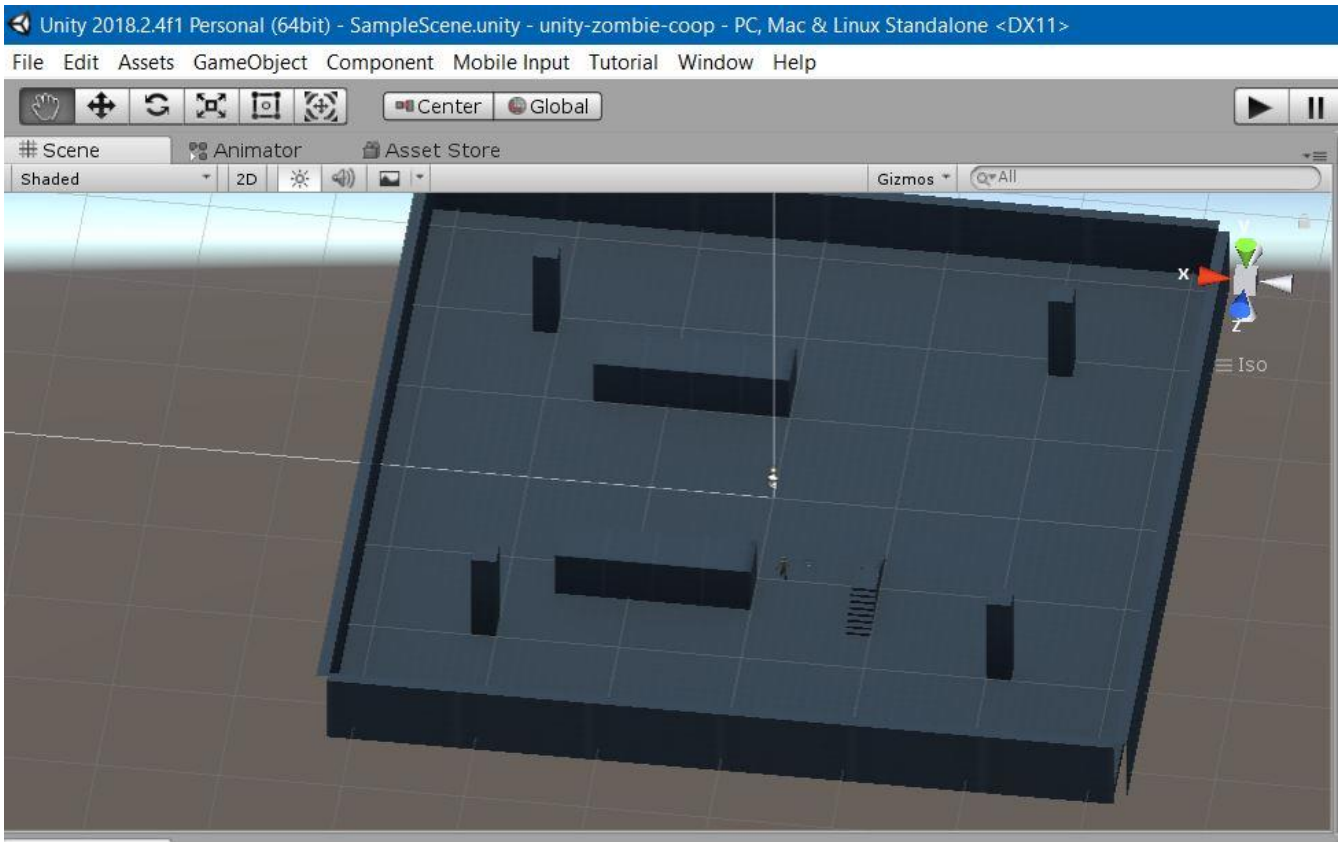
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class EnemySpawner : MonoBehaviour
{
    public GameObject zombiePrefab;
    [HideInInspector] public Transform[] enemySpawnPoints;
    public float spawnDuration = 3f;

    void Start()
    {
        enemySpawnPoints = new Transform[transform.childCount];

        for (int i = 0; i < transform.childCount; i++)
        {
            enemySpawnPoints[i] = transform.GetChild(i);
        }

        StartCoroutine(CoStartSpawning());
    }

    IEnumerator CoStartSpawning()
    {
        while (true)
        {
            for (int i = 0; i < enemySpawnPoints.Length; i++)
            {
                Transform enemySpawnPoint = enemySpawnPoints[i];
                Instantiate(zombiePrefab, enemySpawnPoint.position, enemySpawnPoint.rotation);
            }
            yield return new WaitForSeconds(spawnDuration);
        }
    }
}
```

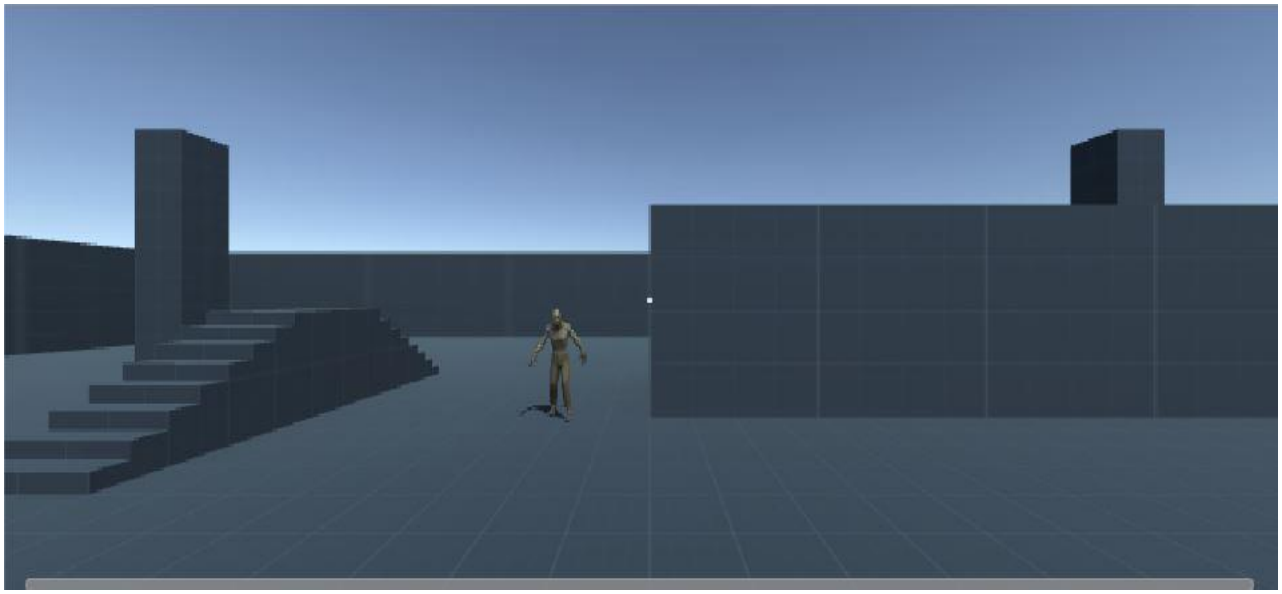
WeaponSway — Update()

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

public class WeaponSway : MonoBehaviour {
    public float amount = 0.1f;
    public float maxAmount = 0.3f;
    public float smoothAmount = 6.0f;

    private Vector3 initPos;

    private void Start()
    {
        initPos = transform.localPosition;
    }

    private void Update()
    {
        float moveX = -Input.GetAxis("Mouse X") * amount;
        float moveY = -Input.GetAxis("Mouse Y") * amount;
        moveX = Mathf.Clamp(moveX, -maxAmount, maxAmount);
        moveY = Mathf.Clamp(moveY, -maxAmount, maxAmount);

        Vector3 finalPosToMove = new Vector3(moveX, moveY,0);
        transform.localPosition = Vector3.Lerp(transform.localPosition, finalPosToMove + initPos, Time.deltaTime * smoothAmount);
    }
}
```

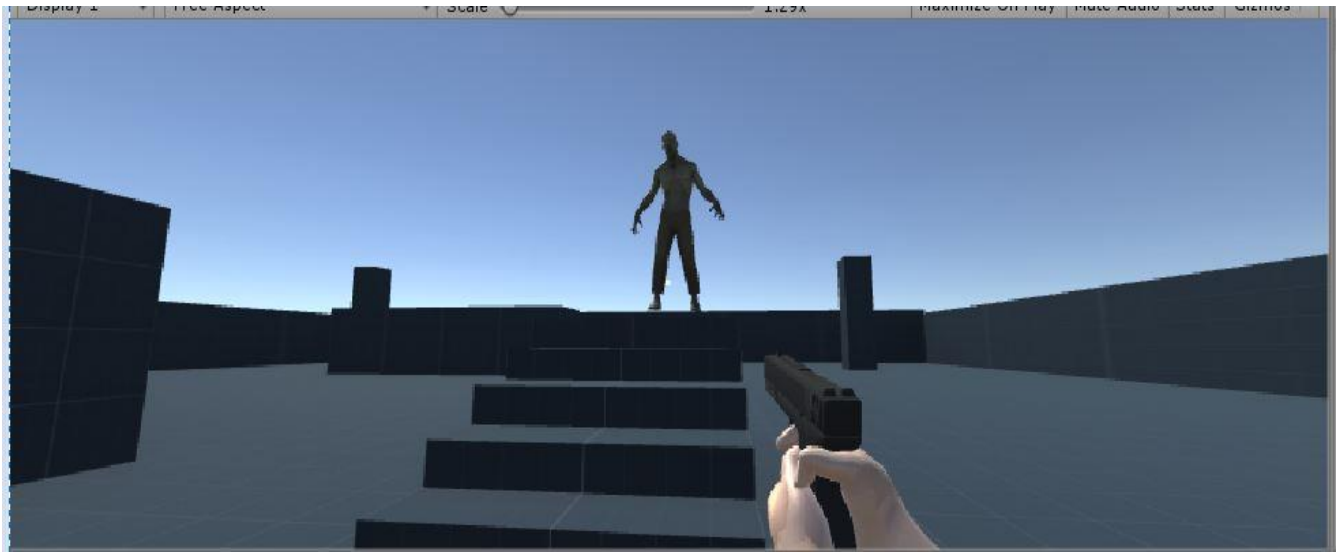
```
Health                                                    parentRef
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Health : MonoBehaviour {
    public Health parentRef;
    public float value = 100f;
    public float damageMultipler = 1.0f;

    public void TakeDamage(float damage)
    {
        damage *= damageMultipler;
        if(parentRef != null)
        {
            parentRef.TakeDamage(damage);
            return;
        }
        value -= damage;
        if(value <= 0)
        {
            value = 0;
        }
    }
}
```

3.4 Snapshots of Project






Display 1
Free Aspect
Scale
1.29x
Maximize On Play
Mute Audio
Stats
Gizmos




YOU ARE DEAD!!

Chapter 4 Limitations and Future Enhancement

4.1 Limitation

- Learning and implementing C# to code in Unity3D is very tedious and time consuming process. Although there are lot of tutorials available online but the user still needs good amount of time to actually be able to thoroughly use the tools available in Unity3D.

4.2 Future Enhancement

- We would like to solve the bugs that are currently occurring during the gameplay and would consider this project to be in our plans of turning it into a Virtual Reality Game.

Chapter 5 Conclusion

Developing this project was quite a challenging task for us. But it enable us to learn how to work as a team and how to compensate for each other's work. We also learned about project distribution and management. While developing the project, we realized the value of time management.

This project enable us to learn about C# programming and use of Unity3D Tool for game development.

REFERANCES

- https://www.assetstore.unity3d.com/
- https://unity3d.com/learn/tutorials/
- https://www.udemy.com/unity-game-deveopment-make-professional-3d-games/learn/v4
- https://docs.unity3d.com/Manual/index.html
- https://www.youtube.com/watch?v=xcUNXBny3RA&list=PLKJB2gde1jbpI5GPHT-Epazq1ywVy0UJ2
- https://www.youtube.com/watch?v=G9BdFZ2MCXc&list=PLZ1b66Z1KFKik2g8D4wrmYj4yein4rCk8