

Harry Potter RPG

Project Title: Harry Potter RPG in Unity 3D.

Group - 5

Group Members: Rajakumari Kothareddy (11553413)

Brahmendra Babu Valeti (11543674)

Yamini Bode(11528183)

Course and Term: INFO 5900 Special Problems.

Abstract:

Need of the Work: The purpose of this project is to create a virtual reality environment that simulates evacuation behavior. In the real world, emergency evacuations can be chaotic and dangerous, especially if people don't know what to do. This virtual reality simulation aims to provide a safe and controlled environment to teach people how to behave during an emergency.

Importance of the Project: This project is important because it can potentially save lives. By simulating emergency scenarios in a virtual environment, people can learn how to respond appropriately in a safe and controlled setting. This can also be used to test and optimize evacuation plans for buildings and other public spaces.

Target Audience: The target audience for this VR application is anyone who wants to learn how to behave during an emergency evacuation, including building owners, safety personnel, and the general public.

Intended Purpose: The intended purpose of this project is to create a realistic and interactive virtual environment that simulates emergency evacuations. This VR application will allow users to experience different scenarios and learn how to respond appropriately.

Introduction: The Harry Potter series of books and movies have captured the imaginations of millions of fans around the world with its magical world filled with spells, potions, mythical creatures, and engaging characters. With the rise of video games, it was only a matter of time before a Harry Potter role-playing game (RPG) would be developed.

This Harry Potter RPG will provide players with the opportunity to immerse themselves in the magical world of Hogwarts School of Witchcraft and Wizardry, where they can create their own characters, attend classes, cast spells, participate in magical duels, and explore the richly detailed world of the Harry Potter universe.

The game will offer players the chance to choose their own adventure and to experience the magic of Hogwarts in their own unique way. Whether they choose to follow in the footsteps of Harry, Hagrid and Draco or forge their own path, the game will offer countless hours of fun and entertainment.

Through its richly detailed environments, engaging characters, and immersive gameplay, the Harry Potter RPG will transport players to a world unlike any other, where magic and adventure await around every corner.

Goals and Objectives: The goals of this project are to create a realistic and interactive virtual environment that simulates emergency evacuations and to teach users how to behave during an emergency. The objectives are to implement different aspects of the environment, including geometry, textures, animations, and functionality.

Designed Environment: The designed environment is a building with multiple rooms and floors. The environment includes furniture, landscaping elements, and people.

Different Aspects of the Environment: The different aspects of the environment that will be included are:

Geometry: The geometry of the building will be modeled in 3ds Max.

Textures: Textures will be applied to the building and objects in the environment to make them look realistic.

Animations: At least three objects in the environment will be animated, such as doors opening and closing or people walking.

Images: Images will be included to show the different aspects of the environment.

Programming: The project was implemented using Unity 3D, a cross-platform game engine. The programming language used was C#. The programming was divided into several sections, including vision, sound, animation, interactivity, characters/avatars, sensors, player controller, and AI implementation.

```

using System;
using UnityEngine;

namespace UnityStandardAssets.Characters.ThirdPerson
{
    [RequireComponent(typeof (UnityEngine.AI.NavMeshAgent))]
    [RequireComponent(typeof (ThirdPersonCharacter))]
    public class AICharacterControl : MonoBehaviour
    {
        public UnityEngine.AI.NavMeshAgent agent { get; private set; }
        // the navmesh agent required for the path finding
        public ThirdPersonCharacter character { get; private set; } // the
        character we are controlling
        public Transform target; // target to aim for

        private void Start()
        {
            // get the components on the object we need ( should not be null
            // due to require component so no need to check )
            agent = GetComponentInChildren<UnityEngine.AI.NavMeshAgent>();
            character = GetComponent<ThirdPersonCharacter>();

            agent.updateRotation = false;
            agent.updatePosition = true;
        }

        private void Update()
        {
            if (target != null)
                agent.SetDestination(target.position);

            if (agent.remainingDistance > agent.stoppingDistance)
                character.Move(agent.desiredVelocity, false, false);
            else
                character.Move(Vector3.zero, false, false);
        }

        public void SetTarget(Transform target)
        {
            this.target = target;
        }
    }
}

```

Ccode for player.

```

# NPC Activator (Mono Script)
Assembly Information
Filename Assembly-CSharp.dll
using UnityEngine;
using System.Collections;

/*
Collides with NPCs (using a FOV width sphere) to enable/disable them
*/

public class NPCActivator : MonoBehaviour
{
    void OnTriggerEnter(Collider col)
    {
        if (col.tag != "NPC" || col.isTrigger) {
            return;
        }

        // get the ownership so NPC can move
        if (col.gameObject.GetComponent<PhotonView>().owner == null) {
            col.gameObject.GetComponent<PhotonView>().TransferOwnership(Photon
            Network.player);
        }

        col.gameObject.GetComponent<NPC>().setEnabled(true);
    }

    void OnTriggerExit(Collider col)
    {
        if (col.tag != "NPC" || col.isTrigger) {
            return;
        }

        if (col.gameObject.GetComponent<PhotonView>().isMine) {
            col.gameObject.GetComponent<PhotonView>().TransferOwnership(Photon
            Network.masterClient);
        }

        col.gameObject.GetComponent<NPC>().setEnabled(false);
    }
}

```

Code for non playing characters.

Functionality:

Functionality: The following functionalities were implemented in the VR environment:

Vision: Textures and 3D models were used to provide detailed information about the exhibits.

Sound: Ambient sounds and music were used to enhance the immersive experience.

Animation: Three animated objects were included in the environment to bring exhibits to life.

Interactivity: Five user-triggered events were implemented to allow users to interact with the exhibits.

Characters/Avatars: Animated agents with path-following behavior were included in the environment.

Sensors: Three different types of sensors (proximity, time, touch) were used to create

a more realistic experience.

Player: A player controller was added to the scene to allow users to navigate the environment.

AI Implementation: Various behaviors were implemented for the agents in the environment, including selfish, altruistic, learning, and adaptive.

Interface Elements: The interface was designed to be intuitive and easy to use. Menu items such as buttons were included to allow users to interact with the environment easily. The interface elements were designed to blend seamlessly with the environment and not distract from the immersive experience.

User Manual: The user manual for the VR environment included detailed instructions on how to navigate and interact with the environment. The manual included step-by-step instructions for using the player controller, interacting with exhibits, and triggering events.

Usefulness of Application: The VR environment created by the project demonstrated the potential of VR technology for education and entertainment. The application could be used in various settings, including museums, classrooms, and entertainment venues.

Create a visually appealing and interactive environment that would engage users. Showcase the use of textures and 3D models to provide detailed information about the simulated space.

Incorporate sound and animations to enhance the immersive experience.

Implement interactivity through user-triggered events and characters/avatars with path-following behavior.

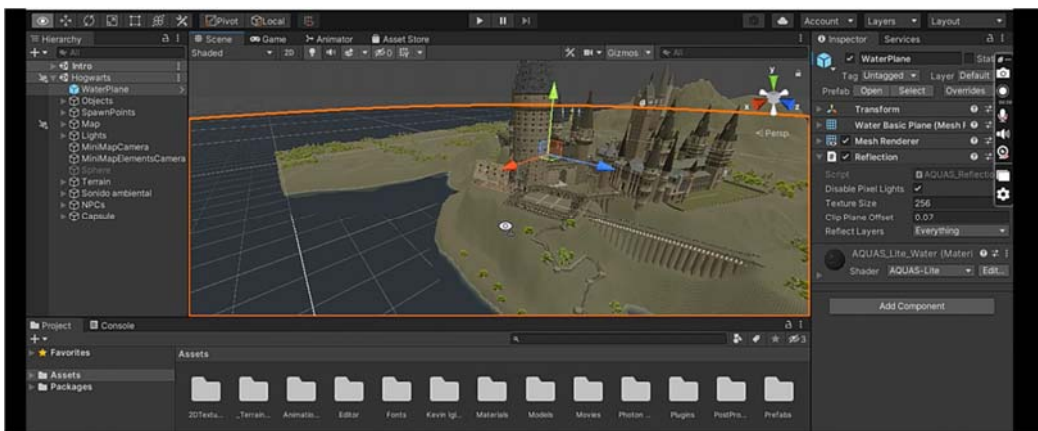
Use sensors to create a more realistic experience.

Implement AI functionality to add more complexity to the environment.

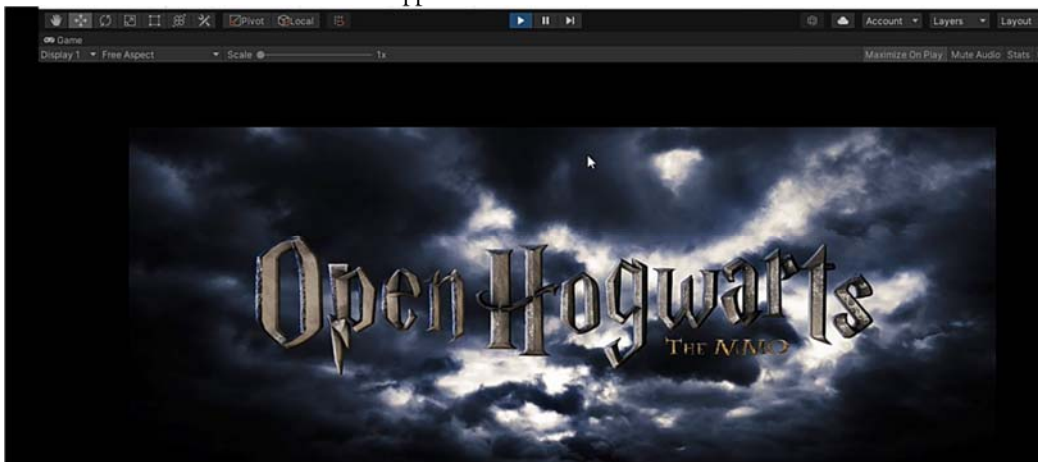
Design an intuitive interface that would allow users to navigate and interact with the environment easily.

Modeling: The envisioned virtual environment for the project was a museum exhibit showcasing various artifacts and exhibits. The environment was designed using 3ds Max, a 3D modeling and animation software. The geometry of the space was carefully planned to ensure that it was visually appealing and allowed for easy navigation. Textures were used to add depth and detail to the exhibits, and animations were used to bring the exhibits to life.

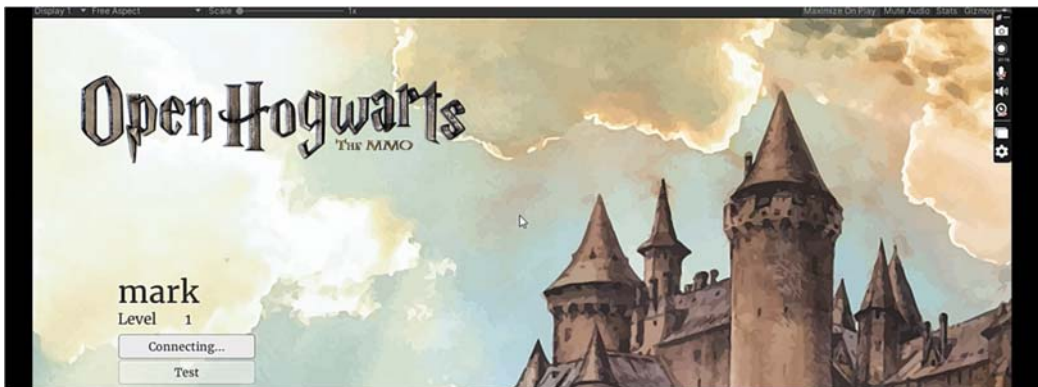
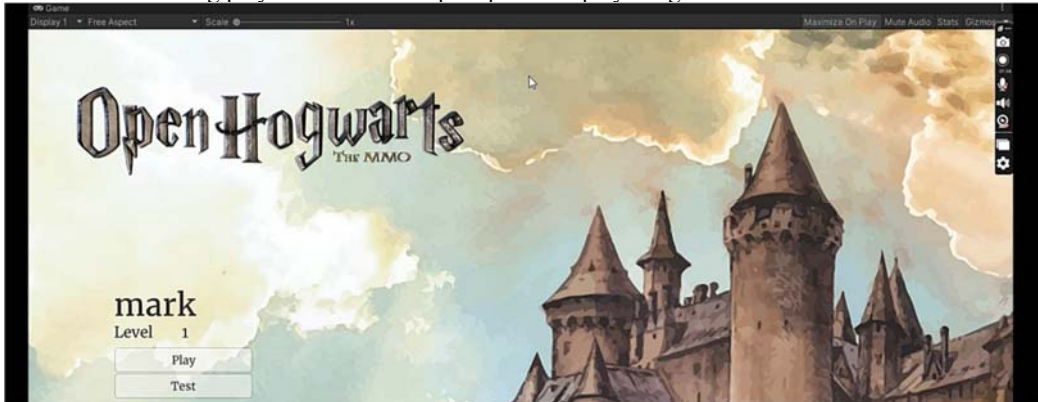
Overview of Architecture



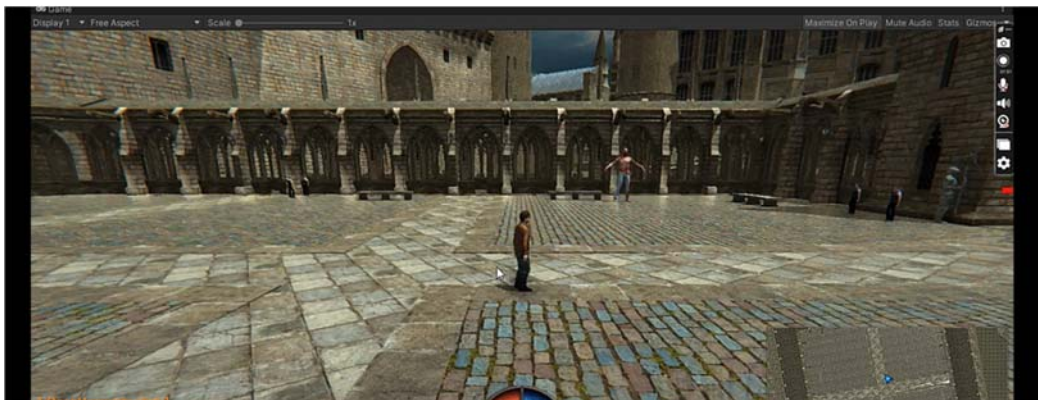
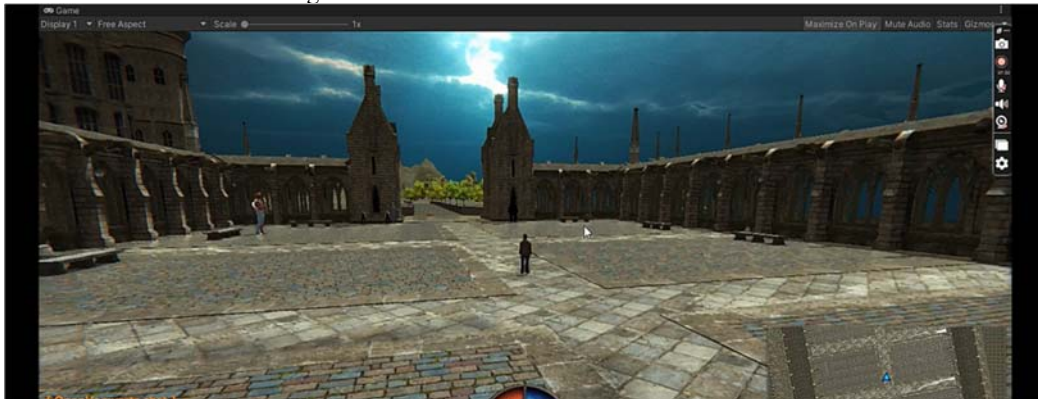
This is how it looks when we start the application



User Interface asking player to select multiple options to play the game



This is how it looks when the game's started



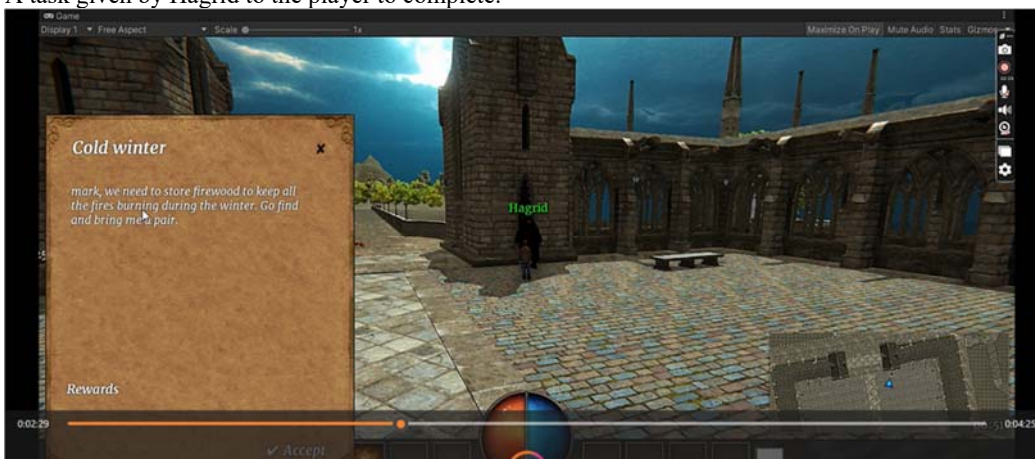
Use multiple controls to make the player react and perform desired activities to complete the game



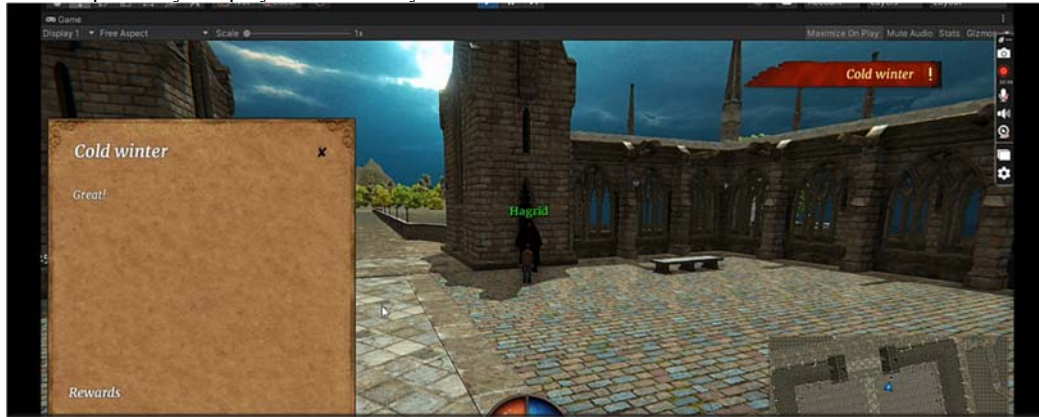
Reach Hagrid and get a task to complete successfully to win the game



A task given by Hagrid to the player to complete.



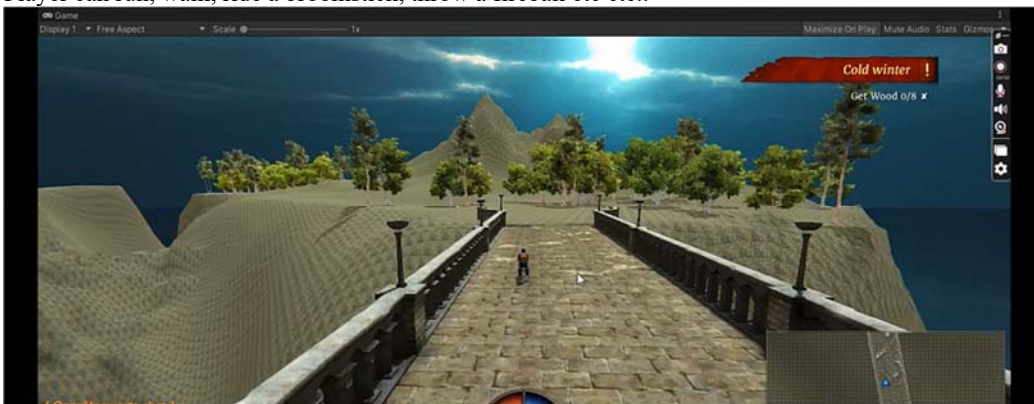
Task completed by the player successfully.

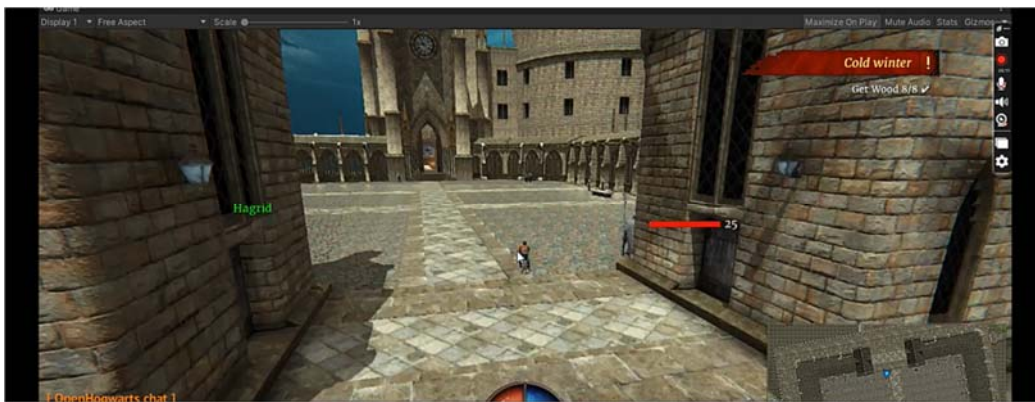
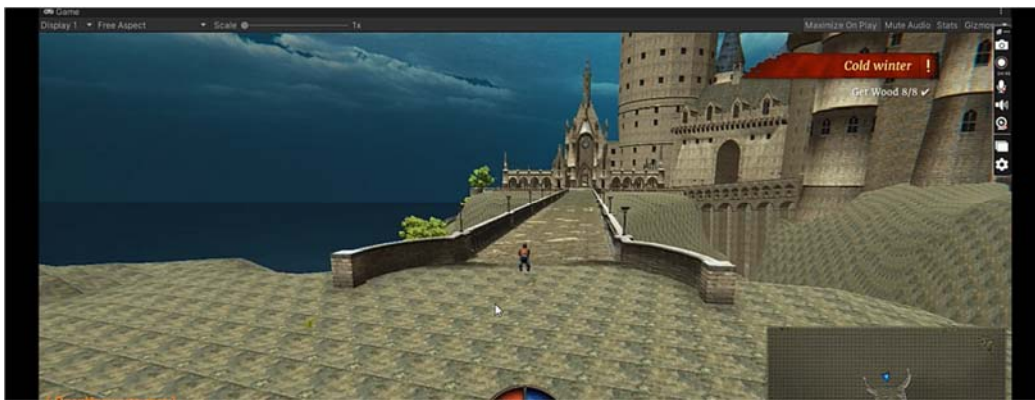
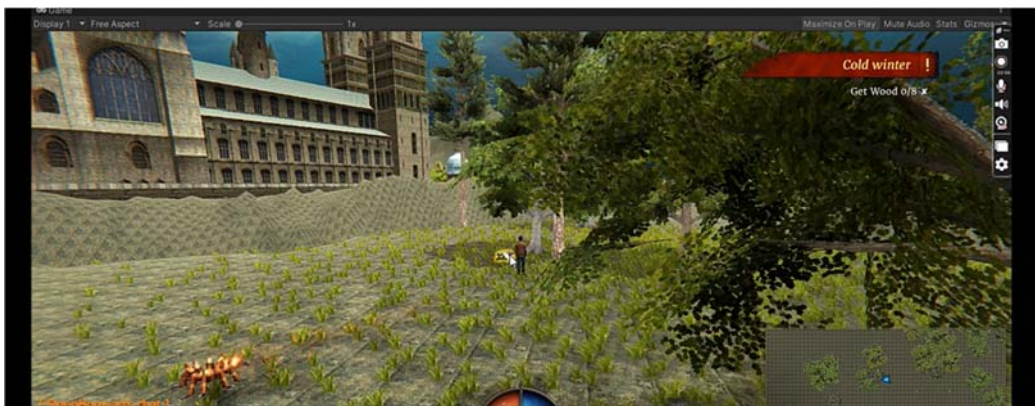
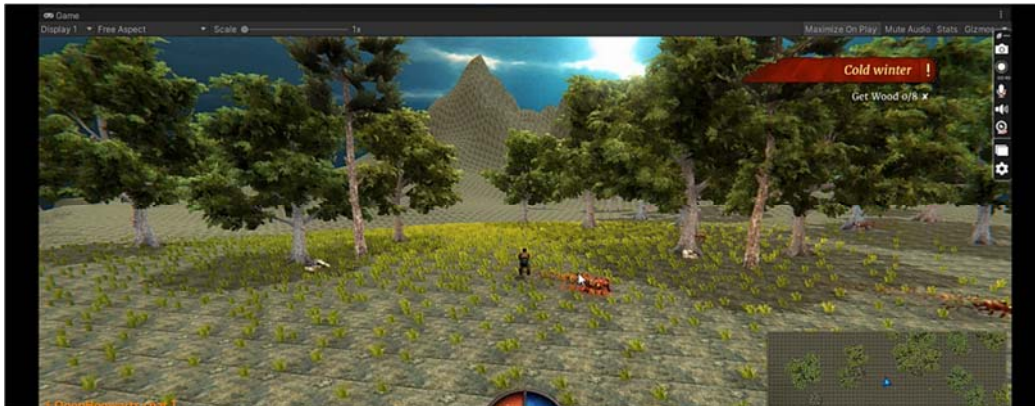


Snapshots of the player completing a given task by Hagrid.



Player can run, walk, ride a broomstick, throw a fireball etc..





Player interacting with other NPC's.

