# Game Design Document (GDD)

Immersive Gaming Experience in VR-Space

Aleksandar Bankov

## 1. Introduction

### 1.1. Elevator pitch

The main purpose of creating games specifically for VR headsets is to leverage the immersive experience they offer. However, developers tend to focus solely on the visual aspects of these games, neglecting the fact that VR headsets provide not only 360 degrees of vision, but also 360 degrees of audio, which significantly enhances the level of immersion.

## 2. Game Overview

### 2.1. Game concept

With this TechDemo, I aim to showcase a glimpse of the potential of spatialized audio in creating a more immersive gaming experience.

### 2.2. Audience

This Tech Demo is targeted at game developers and individuals interested in exploring how audio cues can elevate the level of immersion in gaming experiences.

### 2.3. Genre

Immersive Experience

### 2.4. Setting

Nature

### 2.5. World structure

Open-World ready to be explored.

### 2.6. Player

Generic VR Controller.

### 2.7. Core loop

Exploring the world, enjoying the ambiance and interacting with provided interactables.

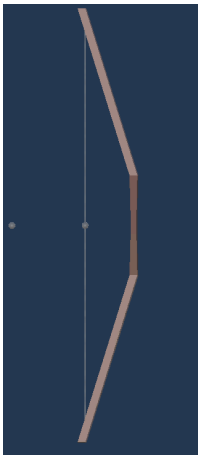
### 2.8. Look & Feel

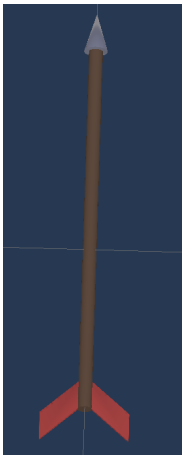
Virtual stroll through nature.

## 3. Mechanics

### 3.1 Bow

Fully interactable bow, taking advantage of the vr controllers, having the same mechanics as a real-world bow, you grab the bow, pull the string and shoot an arrow. When you shoot the arrow you can hear the sound of the pluck being released and when a collision between an arrow and an object occurs the user can hear the arrow landing.





```
private void CheckCollision()
{
    if (Physics.Linecast(_lastPosition, tip.position, out RaycastHit hitInfo))
    {
        if (hitInfo.transform.gameObject.layer != 8)
        {
            if (hitInfo.transform.TryGetComponent(out Rigidbody body))
            {
                PlayHitSound();
                _rigidbody.interpolation = RigidbodyInterpolation.None;
                transform.parent = hitInfo.transform;
                body.AddForce(_rigidbody.velocity, ForceMode.Impulse);
            }else if (hitInfo.transform.TryGetComponent(out Collider collider))
            {
                PlayHitSound();
            }
            Stop();
        }
    }
}
```

### 3.2 Character movement

The player is free to walk in each axis, the character controller checks the position of the user’s head and moves the camera in this direction. Movement in each direction is controlled with the stick on the left controller, rotation with the stick of the right controller, jump with the ”B” button on the right controller.

### 3.3 Assets

- Bow Complete
- Arrow
- XR Origin
- Terrain

# 4. Graphics and audio

## 4.1. Visual system

The visual system is based on Virtual Reality, as the reason for that is to take most of the situation with our ambient audio

#### 4.1.1. Player camera

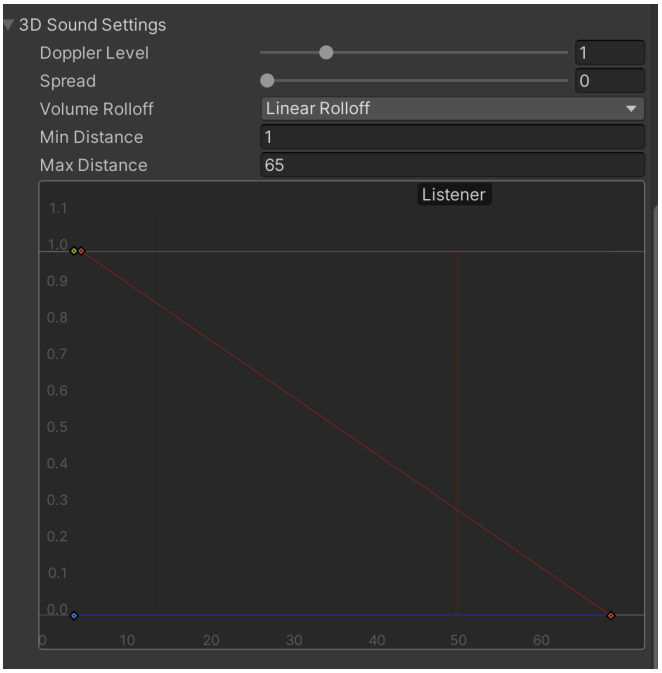
The player sees the games through his eyes from the VR-Headset

## 4.2. Interface

There is no interface as the point of the game is to showcase the capabilities of audio output through VR-HMD.

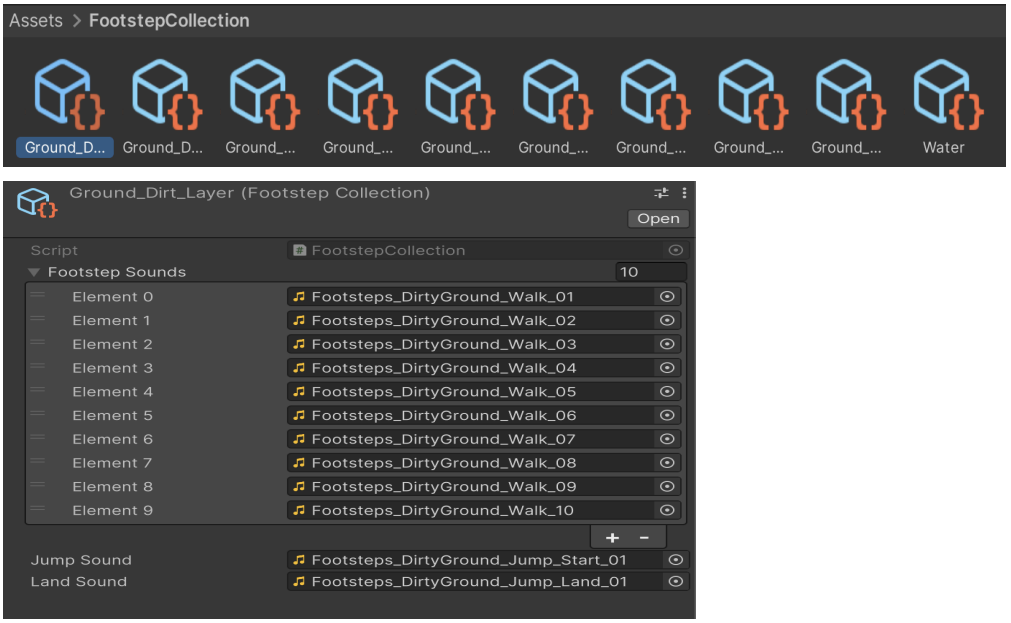
## 4.3. Audio system

The audio system utilizes the Meta XR Audio SDK to convert audio sources into spatial audio sources using the SpatializerMixer. This allows for optimized VR-HMD audio. Additionally, the system considers the distance between the player and the audio source, incorporating linear rolloff to adjust volume based on real-life distance.



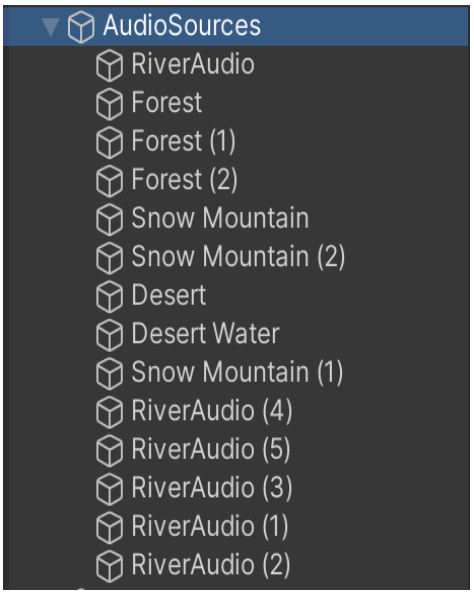
### 4.3.1 Terrain-Specific movement audio

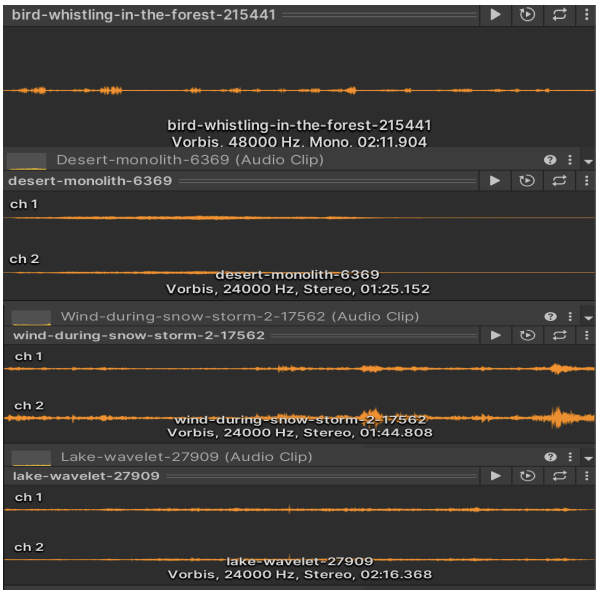
As the player moves, the system checks the type of terrain being traversed and matches it with predefined containers for each terrain type. These containers store specific footstep sounds, as well as jump and landing sounds. Multiple footstep sounds per type are included to closely emulate real-world walking. The sounds are configured to be output by the VR headset as if they are originating from beneath the player.




### 4.3.2. Zone-Specific ambient audio

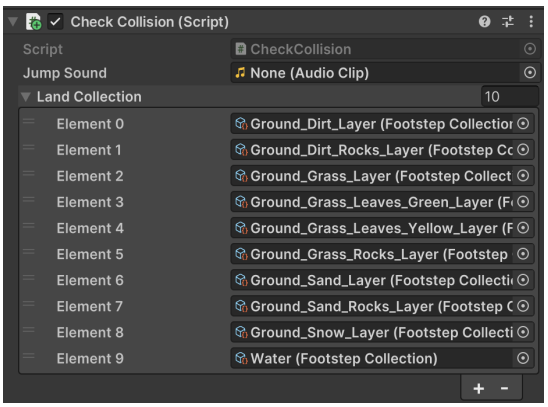
The world features four distinct zones: Forest, Desert, Snow Mountain, and Water Bodies. Each zone is associated with specific ambient audio: - Forest: Bird chirping and wind through the trees - Desert: Approaching sandstorm - Snow Mountain: Strong winds and heavy snowfall - Water Bodies: Flowing water currents



### 4.3.3. Terrain-Specific interactables audio output

Players are able to pick up small rocks and throw them. Similar to the movement audio, the system checks the terrain type and plays a specific landing sound based on the terrain type

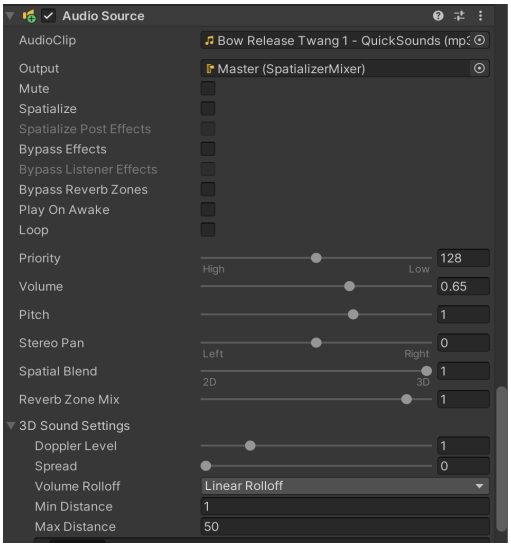



```
public void CheckLayer()
{
    RaycastHit hit;
    if (Physics.Raycast(origin: _rigidbody.transform.position, direction: Vector3.down, out hit, maxDistance: 1))
    {
        if (hit.transform.GetComponent<Terrain>() != null)
        {
            Terrain t = hit.transform.GetComponent<Terrain>();
            if (currentLayer != checker.GetLayerName(_rigidbody.transform.position, t))
            {
                currentLayer = checker.GetLayerName(_rigidbody.transform.position, t);
                foreach (FootstepCollection collection in landCollection)
                {
                    if (currentLayer == collection.name)
                    {
                        _audioSource.clip = collection.landSound;
                    }

                }
```

### 4.3.4 Bow and Arrow Audio

When releasing the string of the bow an audio is played, as well when an arrow hits an object.





# 5. Game world

## 5.1. Look & Feel of the world

The experience can be described as aimlessly wandering through nature and simply enjoying the ambiance.

## 5.2. Locations

Each location is significant, as it has unique characteristics and specific sounds for walking, jumping, landing, and ambient noise. This is designed to simulate a real-time experience for the player.