Documentatie

Mixed reality project



Inhoudsopgave

[Summary 2](#_Toc151502495)

[Intro 3](#_Toc151502496)

[Games: 3](#_Toc151502497)

[Animations: 3](#_Toc151502498)

[Assets: 3](#_Toc151502499)

[Scripts: 3](#_Toc151502500)

[Scene 1 4](#_Toc151502501)

[Games: 4](#_Toc151502502)

[Animations: 4](#_Toc151502503)

[Assets: 4](#_Toc151502504)

[Scripts: 4](#_Toc151502505)

[Scene 2 5](#_Toc151502506)

[Games: 5](#_Toc151502507)

[Animations: 5](#_Toc151502508)

[Assets: 5](#_Toc151502509)

[Scripts: 5](#_Toc151502510)

[Scene 3 6](#_Toc151502511)

[Games: 6](#_Toc151502512)

[Animations: 6](#_Toc151502513)

[Assets: 6](#_Toc151502514)

[Scripts: 6](#_Toc151502515)

[Outro 7](#_Toc151502516)

[Reference: 8](#_Toc151502517)

[Codes 8](#_Toc151502518)

# Summary

In this immersive VR experience, players dive into a unique world where art comes alive. The game revolves around a painting split into four distinct sections, each serving as a portal into a mesmerizing realm. As players navigate through these segments, they encounter a series of mini-games intricately woven into the artistic landscape.

In the first part, the painting unveils a serene landscape, inviting players to solve puzzles amidst rolling hills and tranquil streams. Each interaction triggers animated responses, revealing hidden paths and unlocking the essence of the scenery.

Moving into the second segment, the painting transforms into an urban scene bustling with life. Here, players engage in challenges that mimic the vibrant cityscape, from rhythm-based tasks to interactive street art, all while experiencing dynamic animations that react to their actions.

The third part transports players into a fantastical realm, where mythical creatures and magical elements intertwine. Through intuitive gameplay, they unravel the secrets of this enchanted world, engaging with spellbinding animations and uncovering surprises hidden within the canvas.

Finally, the fourth section presents a surreal dreamscape, blurring the lines between reality and imagination. Players navigate mind-bending puzzles and illusions, exploring interactive elements that warp perception and challenge their senses, all within the confines of the painting's boundaries.

Throughout the game, the seamless integration of animations and interactive elements within each painting segment creates an enchanting and immersive experience, allowing players to step beyond the frame and become integral parts of the artwork itself.

## Intro

## Games:

* None

## Animations:

* None

## Assets:

* Fog
* Painting
* FadeCanvas

## Scripts:

* Intro manager
  + Move the painting
  + Control the FadeCanvas
  + Teleport to next scene
* PaintingScript
  + Make the painting come to life with different delays

# Afbeelding met verven, kunst, Beeldende kunst, tekening Automatisch gegenereerde beschrijvingScene 1

## Games:

* Playing with a doll
* Bikkelen
* Playing with a tame bird
* Poking a turd
  + With sound
* Hobby Horse

## Animations:

* Playing with a doll
* Blowing bubbles
* Handstand
* Waving
* Bikkelen

## Assets:

* House
* Fence
* Bird
  + Sound
  + Animations
* Child seat

## Scripts:

* Teleport to next scene
* On touch make sound

# Afbeelding met verven, tekening, gebouw, kunst Automatisch gegenereerde beschrijvingScene 2

## Games:

* Tops (tollen)
* Stilts (stelten)

## Animations:

* Dancing
* Climbing a tree
* Petanque
* Swimming

## Assets:

* House
* River
* Bird
  + Sound
  + Animations

## Scripts:

* Teleport to next scene
* Walking on stilts
* Spinning the tops

# Afbeelding met verven, tekening, kunst, persoon Automatisch gegenereerde beschrijvingScene 3

## Games:

* Swing
* Hoepel rollen
* Rinkelhoep

## Animations:

* Haasje over

## Assets:

* Bricks -> Well
* Barrel

## Scripts:

* Teleport to next scene

# Outro

# Reference:

## Codes

**Intro Manager**

|  |
| --- |
| *using System.Collections;*  *using System.Collections.Generic;*  *using UnityEngine;*  *using UnityEngine.SceneManagement;*  *public class IntroManager : MonoBehaviour*  *{*  *public GameObject painting;*  *public float PaintingSpeed = 0.5f;*  *public GameObject CanvasFade;*  *private float i = 0;*  *private float startpos;*  *private bool fadeout = false;*  *void Start()*  *{*  *startpos = painting.transform.position.x;*  *StartCoroutine(DelayPainting());*  *}*  *void Update()*  *{*  *if (painting.active == true)*  *{*  *if (painting.transform.position.x <= -5)*  *{*  *i = (Time.time \* PaintingSpeed) + startpos;*  *painting.transform.position = new Vector3(i, painting.transform.position.y, painting.transform.position.z);*  *}*  *else*  *{*  *StartCoroutine(ActivatePainting());*  *}*  *}*  *if (painting.GetComponent<PaintingMoveBlocks>().endIntro == true && !fadeout)*  *{*  *CanvasFade.GetComponent<FadeCanvas>().StartFadeIn();*  *fadeout = true;*  *}*  *if (fadeout)*  *{*  *if (CanvasFade.GetComponent<FadeCanvas>().GetComponent<CanvasGroup>().alpha == 1f)*  *{*  *SceneManager.LoadScene("Scene1");*  *}*  *}*  *}*  *IEnumerator DelayPainting()*  *{*  *yield return new WaitForSeconds(5);*  *painting.SetActive(true);*  *}*  *IEnumerator ActivatePainting()*  *{*  *yield return new WaitForSeconds(2.5f);*  *painting.GetComponent<PaintingMoveBlocks>().Active = true;*  *}*  *}* |

**Teleport to next scene**

|  |
| --- |
| *using System.Collections;*  *using System.Collections.Generic;*  *using UnityEngine;*  *using UnityEngine.SceneManagement;*  *public class OnTouchChangeScene : MonoBehaviour*  *{*  *public string scene;*  *void OnTriggerEnter(Collider other)*  *{*  *Debug.Log(other.gameObject.tag);*  *if (other.gameObject.tag == "Player")*  *{*  *SceneManager.LoadScene(scene);*  *}*  *}*  *}* |

**ChangeHeight**

|  |
| --- |
| *using System.Collections;*  *using System.Collections.Generic;*  *using UnityEngine;*  *public class ChangeHeigt : MonoBehaviour*  *{*  *public GameObject xrOrigin;*  *public int height = 4;*  *public void changeHeight()*  *{*  *xrOrigin.transform.position = new Vector3(xrOrigin.transform.position.x, xrOrigin.transform.position.y + height, xrOrigin.transform.position.z);*  *}*  *public void resetHeight()*  *{*  *xrOrigin.transform.position = new Vector3(xrOrigin.transform.position.x, xrOrigin.transform.position.y - height, xrOrigin.transform.position.z);*  *}*  *}* |