TECHNISCHE UNIVERSITÄT CHEMNITZ

Informatik

Dr.Thomas Kronfeld

Research Project

Implementation of a sample application for the Augmented Reality glasses Microsoft HoloLens

Xiangyu Tong

Chemnitz, den 1. September 2021

Supervisor: Dr. Kronfeld

Tong, Xiangyu
Implementation of a sample application for the Augmented Reality glasses Microsoft HoloLens
Research Project, Informatik
Technische Universität Chemnitz, September 2021

Thanksgiving

First of all, I want to thank my supervisor Dr. Kronfeld, who gave me a lot of valuable opinions on my research project, so that I have a goal and direction in developing and writing this report.

Abstract

Microsoft HoloLens, known under development as Project Baraboo, is a pair of mixed reality smartglasses developed and manufactured by Microsoft. HoloLens was the first head-mounted display running the Windows Mixed Reality platform under the Windows 10 computer operating system. The tracking technology used in HoloLens can trace its lineage to Kinect, an add-on for Microsoft's Xbox gaming console that was introduced in 2010.

This report is about developing a simple and intuitive game-application for HoloLens by using sensual and natural interface commands. In addition, we also give solutions for some tricky problems in Implement-Phase.

Inhaltsverzeichnis

1 Introduction

1.1 Problem Statement

Microsoft HoloLens is a mixed reality eyewear which allows the user to display interactive 3D projections in the direct environment with the support of a Natural User Interface. The device uses sensual and natural interface commands like gaze, gesture and voice. audience access to this technology, a simple and intuitive application needs to be developed. Hence, this report aims to show the potential of Microsoft HoloLens in Mixed-Reality-Development and problems & issues while in the phase of implementation.

1.2 Research Topic

According to the statement above, a “Whac-A-Mole” like application should be developed within the context of the research project. So the goal is the development of a game with the following features:

1. Determine a flat region in front of the user (for example the surface of a table)
2. Display virtual 3D model on top of the determined region, at random locations for a random amount of time.
3. If the user reaches or taps on one of these models, the model disappears and an internal counter is increased.
4. After 90 seconds the game is over and the score of the user should be displayed
5. A restart can be triggered with the tap gesture. The environment for testing along with the Microsoft HoloLens device will be provided by the supervisor.

1.3 Prerequisites

To get started with the Mixed Reality Toolkit, the following tools are necessary:

- Visual Studio 2019
- Unity 2018.4.x or higher
  MRTK supports both IL2CPP and .Net scripting backends on Unity 2018
- Windows SDK 18362+
  This is necessary if you are building a UWP app for WMR, Hololens 1&2
- Hololens Emulator
  The Hololens Emulator lets you test holographic applications on your PC without a physical HoloLens

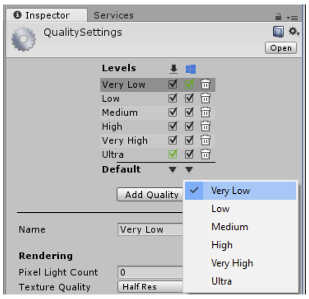
Get the latest MRTK Unity packages and import MRTK packages into your Unity project. To create a HoloLens application,Switch your Unity project to the Universal Windows Platform. Last, add MRTK to a new scene or new project.

1.4 Settings

1.4.1 Setup the project settings for HoloLens-App

Quality settings for HoloLens

1. Select Edit > Project Settings > Quality
2. Select the dropdown under the Windows Store logo and select Very Low.



Levels	⬇	⊞	
Very Low	☑	☑	🗑
Low	☑	☑	🗑
Medium	☑	☑	🗑
High	☑	☑	🗑
Very High	☑	☑	🗑
Ultra	☑	☑	🗑
Default	▼	▼	

Name: Very Low

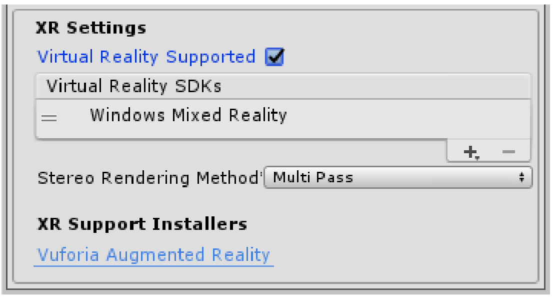
Rendering

Pixel Light Count: 0

Texture Quality: Half Res

Target Windows 10 SDK

1. Edit > Project Settings > Player.
2. select the Windows Store icon.
3. Expand the XR Settings group.
4. In the Rendering section, check the Virtual Reality Supported checkbox to add a new Virtual Reality SDKs list.
5. Verify that Windows Mixed Reality appears in the list.



XR Settings

Virtual Reality Supported ☑

Virtual Reality SDKs

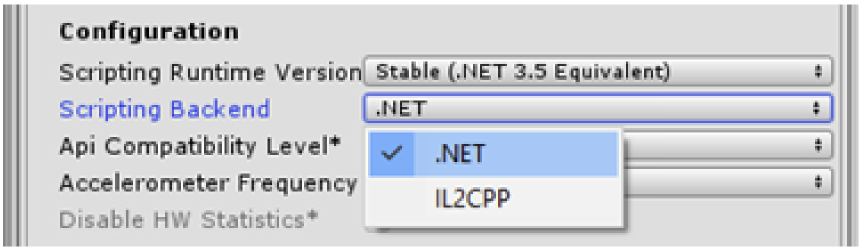
Windows Mixed Reality

Stereo Rendering Method: Multi Pass

XR Support Installers

Vuforia Augmented Reality

Verify .NET Configuration



Configuration

Scripting Runtime Version: Stable (.NET 3.5 Equivalent)

Scripting Backend: .NET

Api Compatibility Level*

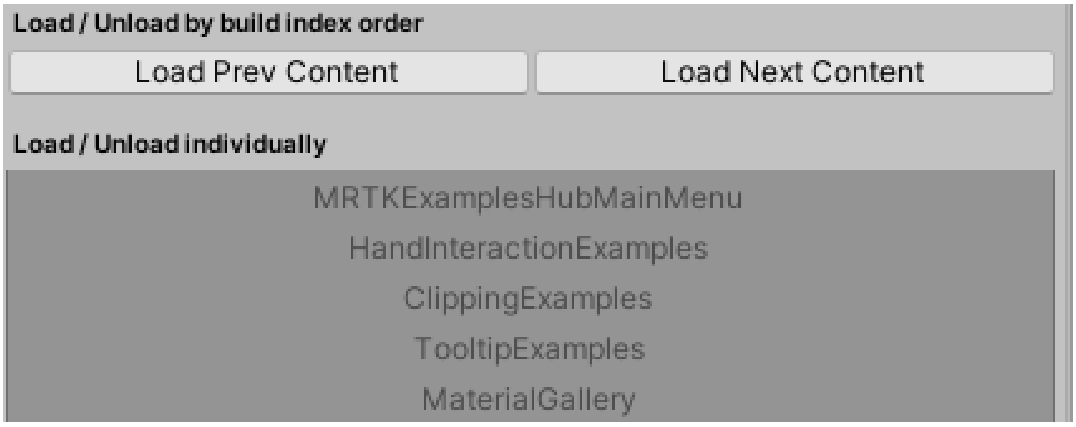
Accelerometer Frequency

Disable HW Statistics*

make sure that Scripting Backend is set to .NET
Notice: Unity 2019 the .NET backend was deprecated and only IL2CPP remains

1.4.2 MRTK Scene Management

If your project consists of a single scene, the Scene System probably isn't necessary.



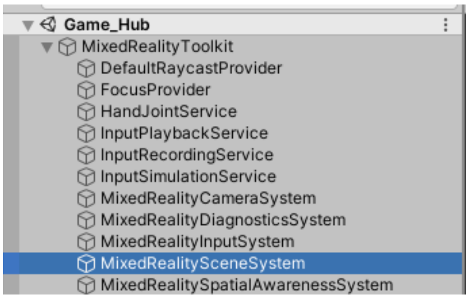
Load / Unload by build index order

Load / Unload individually

It is most useful when one or more of the following are true:

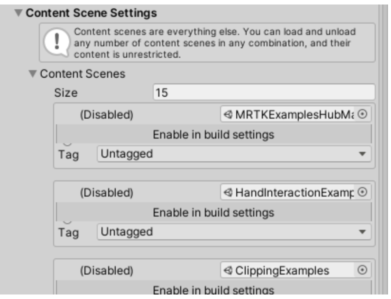
- Your project has multiple scenes.
- You're used to single scene loading, but you don't like the way it destroys the MixedRealityToolkit instance.
- You want a simple way to additively load multiple scenes to construct your experience.
- You want a simple way to keep track of load operations in progress or a simple way to control scene activation for multiple scenes being loaded at once.
- You want to keep lighting consistent and predictable across all your scenes.

MRTK offers SceneSystem to manage and set scenes



```
▼ Game_Hub
  ▼ MixedRealityToolkit
      DefaultRaycastProvider
      FocusProvider
      HandJointService
      InputPlaybackService
      InputRecordingService
      InputSimulationService
      MixedRealityCameraSystem
      MixedRealityDiagnosticsSystem
      MixedRealityInputSystem
      MixedRealitySceneSystem
      MixedRealitySpatialAwarenessSystem
```

Add scene to Content Scenes so that users can load and unload any number of content scenes in any combination.



▼ Content Scene Settings

Content scenes are everything else. You can load and unload any number of content scenes in any combination, and their content is unrestricted.

▼ Content Scenes

Size: 15

(Disabled) MRTKExamplesHubMa
Enable in build settings
Tag: Untagged

(Disabled) HandInteractionExamp
Enable in build settings
Tag: Untagged

(Disabled) ClippingExamples
Enable in build settings

1.4.3 Emulator

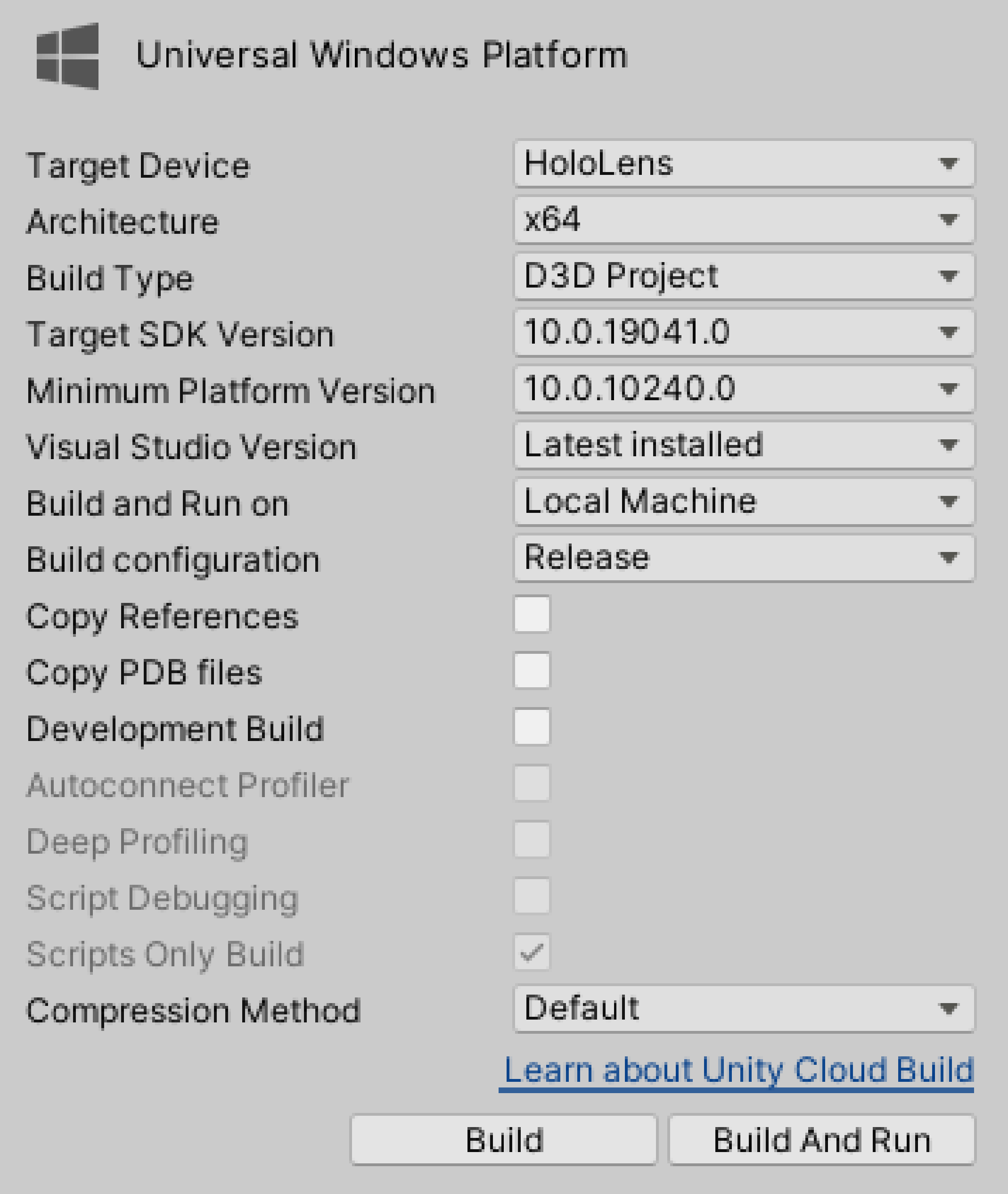
Hardware Reqyurenebts:

- 64-bit CPU
- 8 GB of RAM or more

In the BIOS, the following features must be supported and enabled:

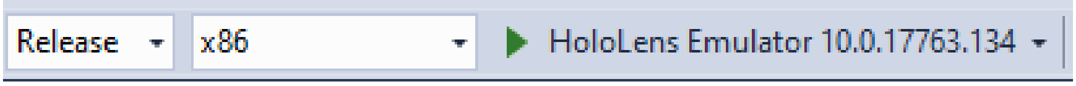
- Hardware-assisted virtualization
- Second Level Address Translation (SLAT)
- Hardware-based Data Execution Prevention (DEP)
- DirectX 11.0 or later
- WDDM 1.2 graphics driver or later (1st gen)

Ensure that the Hyper-V feature has been enabled.
Then build and export to the Visual Studio solution (see the following figures).

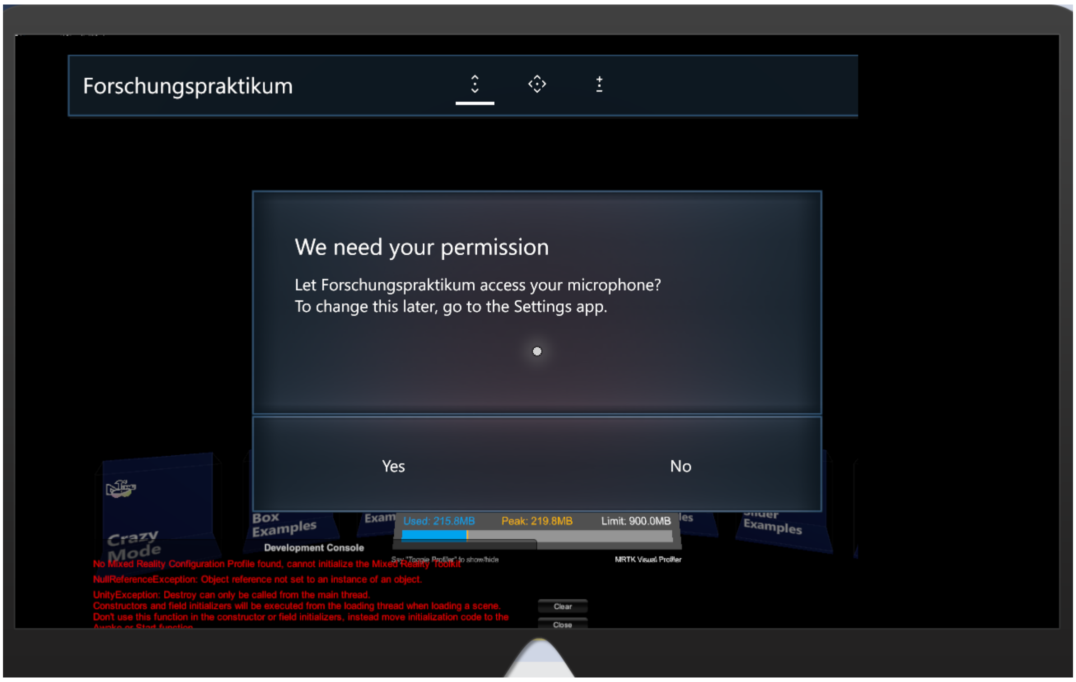


Target Device	HoloLens
Architecture	x64
Build Type	D3D Project
Target SDK Version	10.0.19041.0
Minimum Platform Version	10.0.10240.0
Visual Studio Version	Latest installed
Build and Run on	Local Machine
Build configuration	Release
Copy References	□
Copy PDB files	□
Development Build	□
Autoconnect Profiler	□
Deep Profiling	□
Script Debugging	□
Scripts Only Build	☑
Compression Method	Default

Open the generated Visual Studio solution



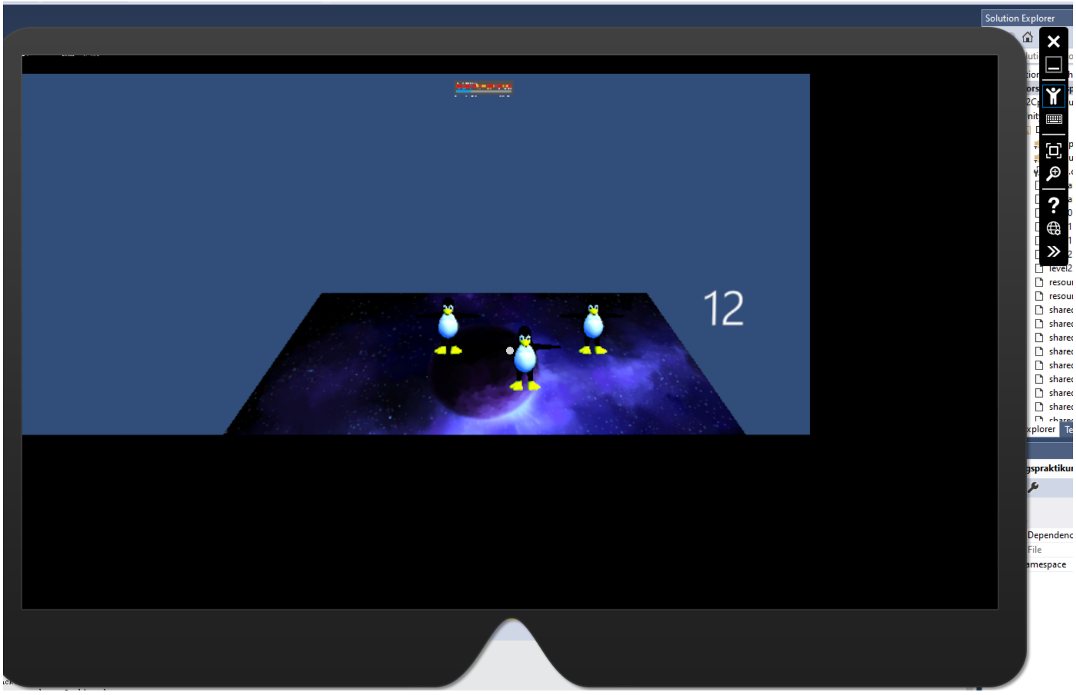
Release | x86 | ▶ HoloLens Emulator 10.0.17763.134



We need your permission

Let Forschungspraktikum access your microphone?
To change this later, go to the Settings app.

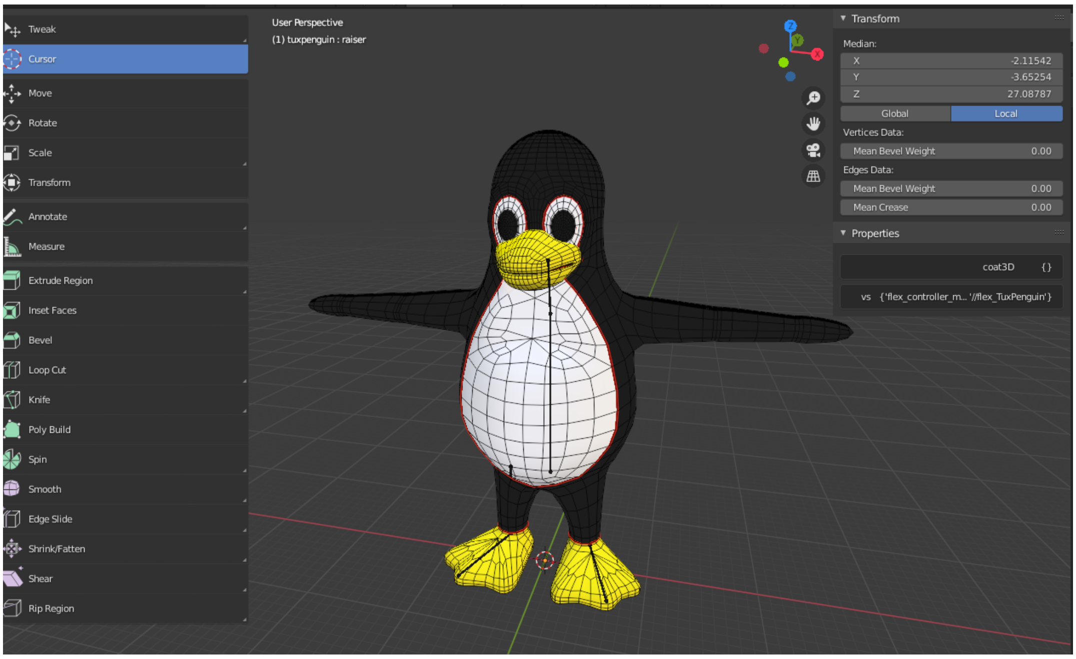
```
No Mixed Reality Configuration Profile found, cannot initialize the Mixed Reality Toolkit
NullReferenceException: Object reference not set to an instance of an object.
UnityException: Destroy can only be called from the main thread.
Constructors and field initializers will be executed from the loading thread when loading a scene.
Don't use this function in the constructor or field initializers, instead move initialization code to the
Awake or Start function.
```



2 Main Section: Methods and Implementation

2.1 Postprecessing of 3D-Models

3D-models can be obtained or constructed by using some softwares like Blender, Sketchup Free, or download 3d-models directly from 3D-model-shop(Here in this report the author used Tux). There are many model type that are compatible with Unity: FBX,BLENDER,gLTF,USDZ.



Median:
X -2.11542
Y -3.65254
Z 27.08787

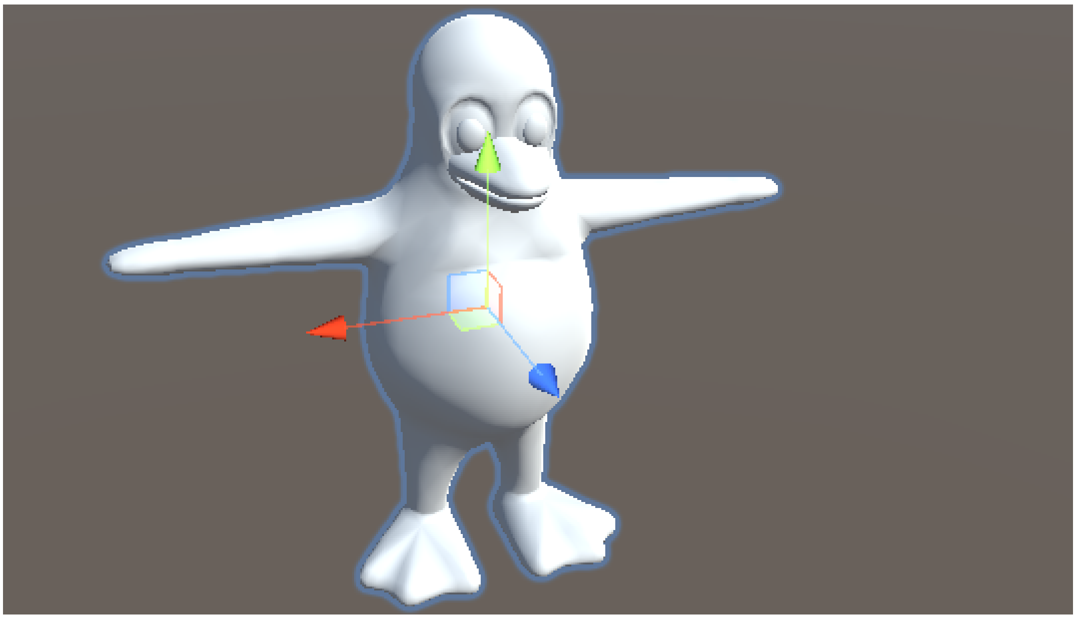
Vertices Data:
Mean Bevel Weight 0.00

Edges Data:
Mean Bevel Weight 0.00
Mean Crease 0.00

coat3D {}
vs {'flex_controller_m... '//flex_TuxPenguin'}

From the figure above, modifying models such as texturing, rendering, even sculpting is possible using Blender. After postprocessing of the 3D-models, then they can be imported into Unity project for further applications. As suggestion by the author, models and their corresponding textures should be in the same folder

Problem Somehow 3d-models were imported into Unity without textures for the reason that Unity does not normally import textures along with meshes.



2 possible solution:

1. First import texture(copy) in Unity project
2. Then, import model(copy) in Unity project

or try

1. Generate a UV map for the mesh
2. Make the texture for the object
3. Save texture file to a separate file. .jpg should do, or .png for textures with a lot of similar colors/few colors overall.
4. Save it as either a .blend or .fbx file to import to Unity
5. Import object to Unity
6. Browse to find the target object

   Find your material options for your mesh, and drag your texture file from the file viewer into the texture/albedo slot

2.2 Basic Play View

Following are the main steps building a basic View in Unity:

1. Change the Near Clip Plane field to the HoloLens recommended 0.85.
2. Determine a flat region in front of the user.(As shown in Figure 2.1, here a cube with height 0.01). It can be vertical or horizontal, which depends on the specific situation of usage.
3. Drag the 3d-models(Tux) into the scene and modify the parameters of them(position,rotation, scale, see Figure 2.2) to fit the scene.

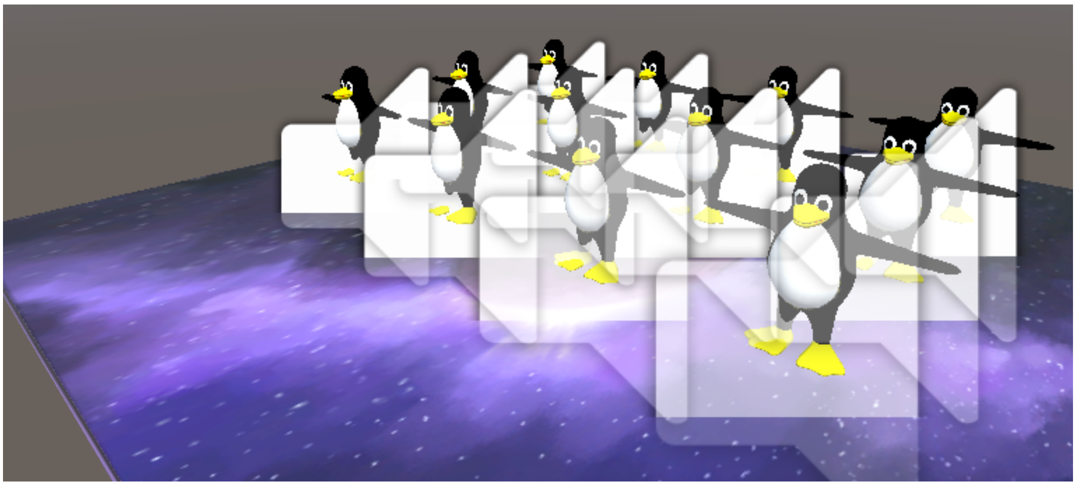
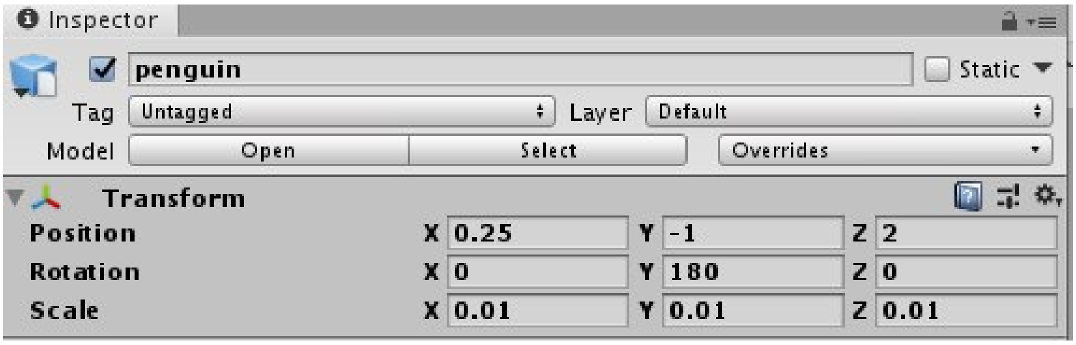


Abbildung 2.1: Basic game view



Inspector

☑ penguin □ Static

Tag Untagged Layer Default

Model Open Select Overrides

Transform

Position	X	0.25	Y	-1	Z	2
Rotation	X	0	Y	180	Z	0
Scale	X	0.01	Y	0.01	Z	0.01

Abbildung 2.2: Parameters of the 3D model

2.3 Physics

2.3.1 Collider

Add a collider to each Tux(Box Collider recommended) to implement Hand-Touch Interactive.

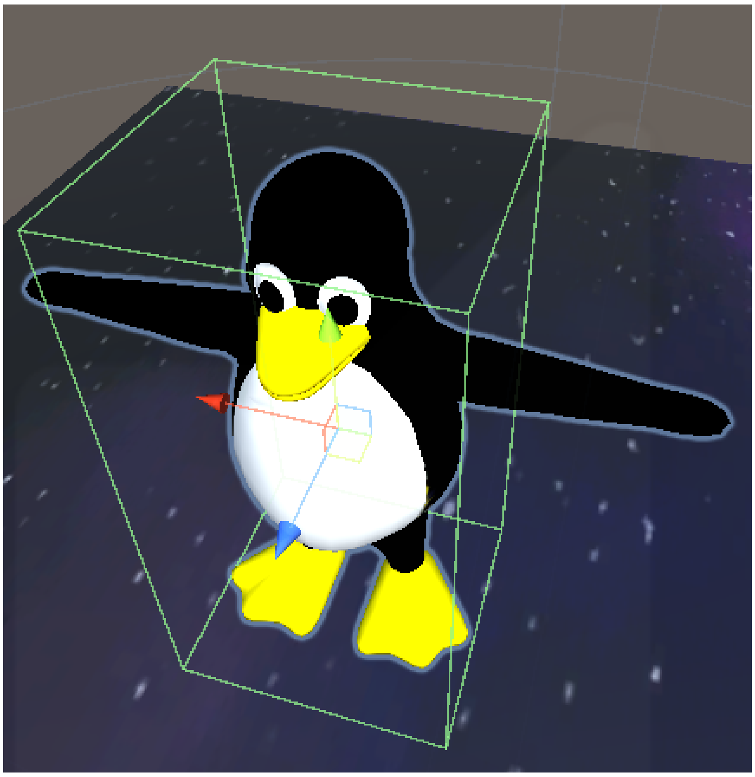
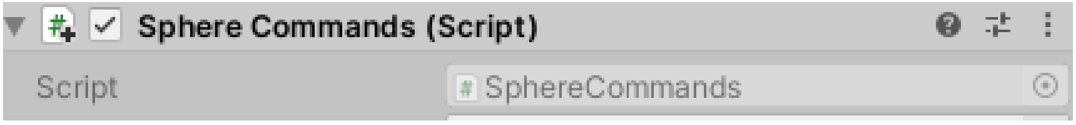


Abbildung 2.3: collider:green lines

Collider defines the valid area involves interaction in gaze and tap-gesture. Hence, the size of collider depends on the specific situation. Furthermore, a group of colliders can be used for one model, so that various physical effect can be implemented in different sub-colliders.

2.3.2 Motion

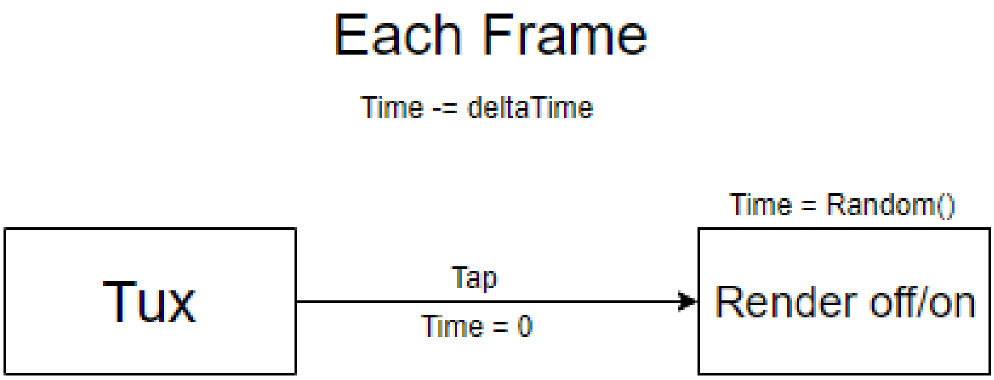
For developing a "Wahc-A-Mole" like application, the "Tuxes" should be displayed at random locations for a random amount of time. Thus, the author adds a script-component for each Tux.
The scripts initializes objects and updates the behavior of objects once per frame:



Sphere Commands (Script)

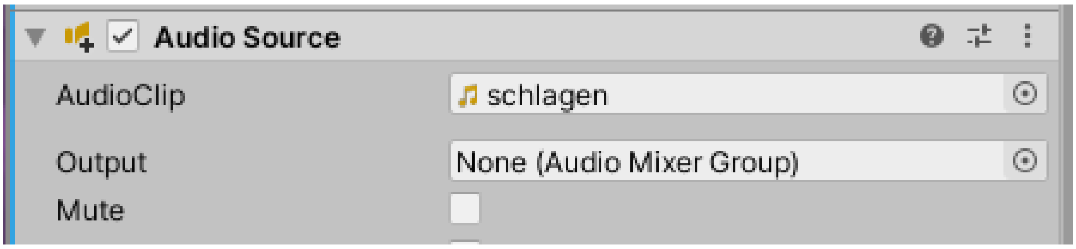
Script: SphereCommands

The basic idea from the author is that each Tux has its own "remaining time" and disappears by Tap-Gesture or remaining time reduces to $0s$.



2.3.3 Sound Effect

Audio-Source-Component is applied in this project so that Sound effect is enabled when Tux is on select and tapped.



Audio Source

AudioClip: schlagen

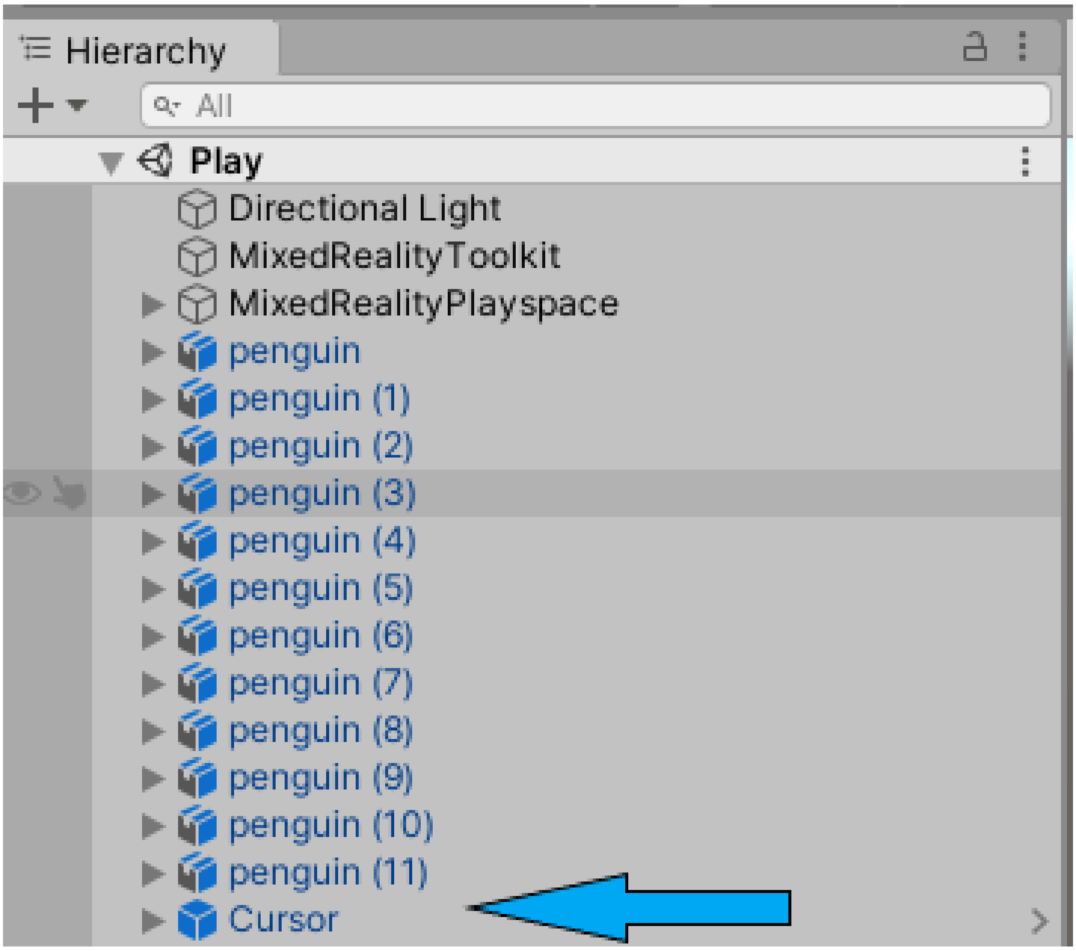
Output: None (Audio Mixer Group)

Mute: □

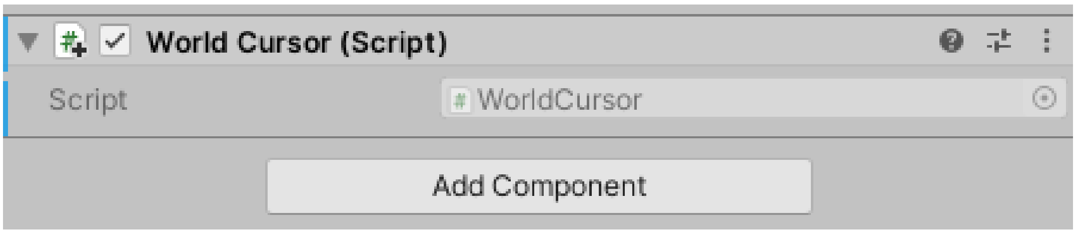
2.4 Gaze-Tap-Gesture Interactive

2.4.1 Gaze and Cursor

One of the interactivities to holograms is Gaze. Gaze is really the user's attention, where they'are looking in the world around them. By using Gaze the user can determine which hologram he or she intends to interact with.
First, a cursor-asset is added to the scene by draging it up to the Hierarchy pane at the root level.



To make this cursor move around every frame, we need to add a script to it.



World Cursor (Script)

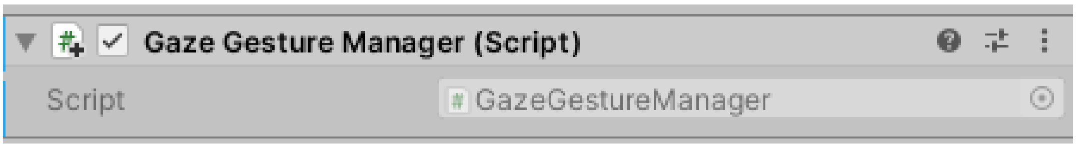
Script WorldCursor

The goal of author here is a Raycast into the scene each frame to figure out where the user's Gaze is. When player figures out where that Gaze point hits one of the holograms(objects), then he or she wants to take the cursor and place it there to make sure that it is oriented outwards so that it looks like it's touching the surface of the object.

2.4.2 Gesture:Tap

After that, the gesture is then implemented in the Application. Specifically, when the user gazes at one of the Tuxes in the scene and does the air-tap gesture, the Tux disappears.

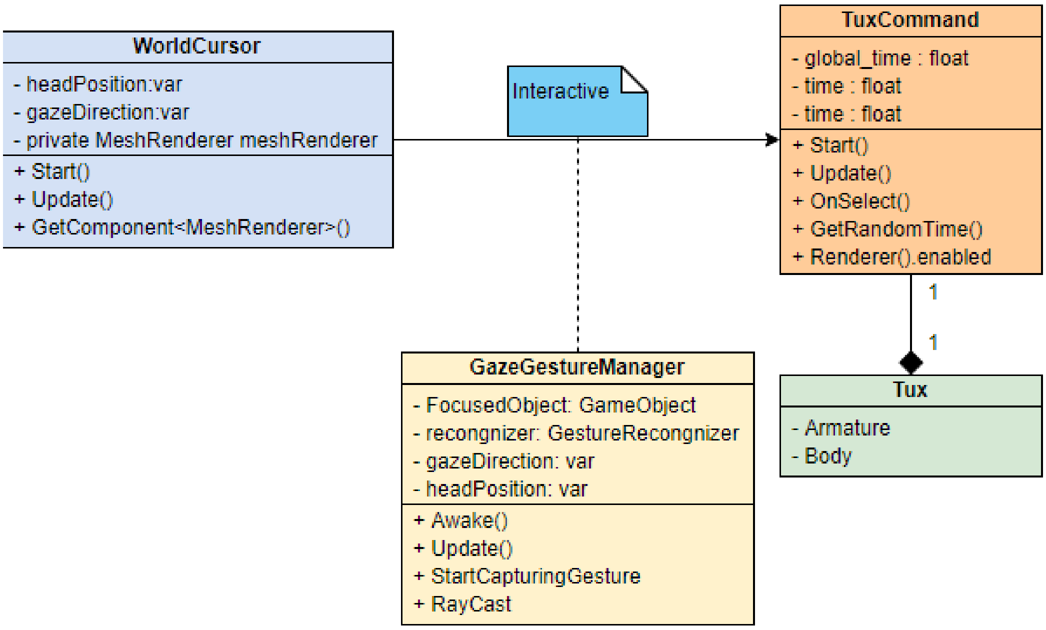
First, add a script that manages the Gaze and Gestures for the user, which ensure that player can use gaze to target a specific Tux and route gestures right to that hologram.



Gaze Gesture Manager (Script)

Script GazeGestureManager

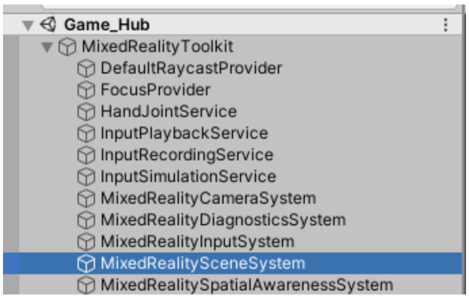
What this script will do is to track each frame where the user is looking at and store that Tux.

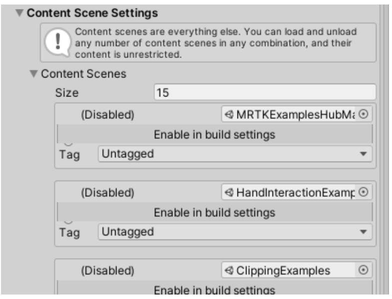


In Unity, objects that have physics will have a component called “Rigidbody”. We can also add one of these Rigidbody components to our Tux when it receives that select message and cause it to fall, or other physic effects.

2.5 Scene-Management

MRTK offers SceneSystem to manage and set scenes





Content Scene Settings

Content scenes are everything else. You can load and unload any number of content scenes in any combination, and their content is unrestricted.

Content Scenes

Size: 15

(Disabled) MRTKExamplesHubMa
Enable in build settings
Tag: Untagged

(Disabled) HandInteractionExamp
Enable in build settings
Tag: Untagged

(Disabled) ClippingExamples
Enable in build settings

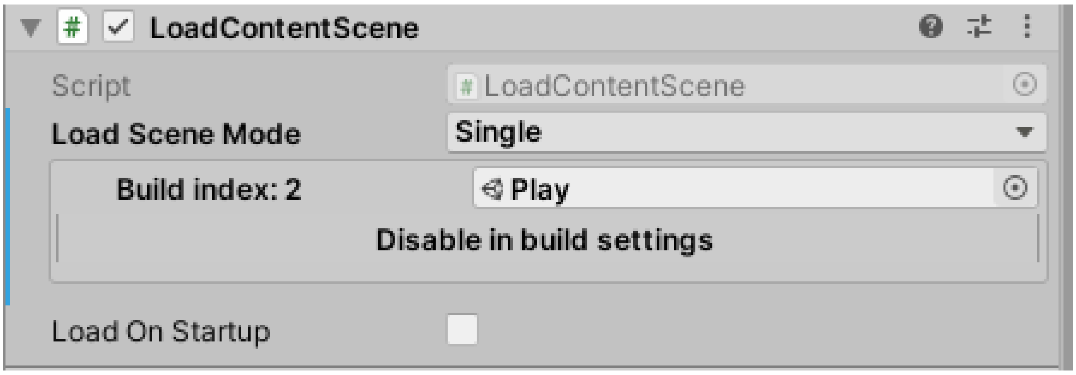
Add scene to Content Scenes so that users can load and unload any number of content scenes in any combination.

2.5.1 Additive Scene loading

All content load operations are asynchronous, and by default all content loading is additive(Adds the Scene to the current loaded Scenes). Manager and lighting scenes are never affected by content loading operations.

2.5.2 Single Scene Loading

The equivalent of a single scene load can be achieved via the optional **mode** argument by using the function **SceneManager.LoadScene(Scene ID, mode= single)**. It will first unload all loaded content scenes before prceeding with the load.



☑ LoadContentScene

Script: LoadContentScene

Load Scene Mode: Single

Build index: 2 — Play

Disable in build settings

Load On Startup: □

2.5.3 Scene Type

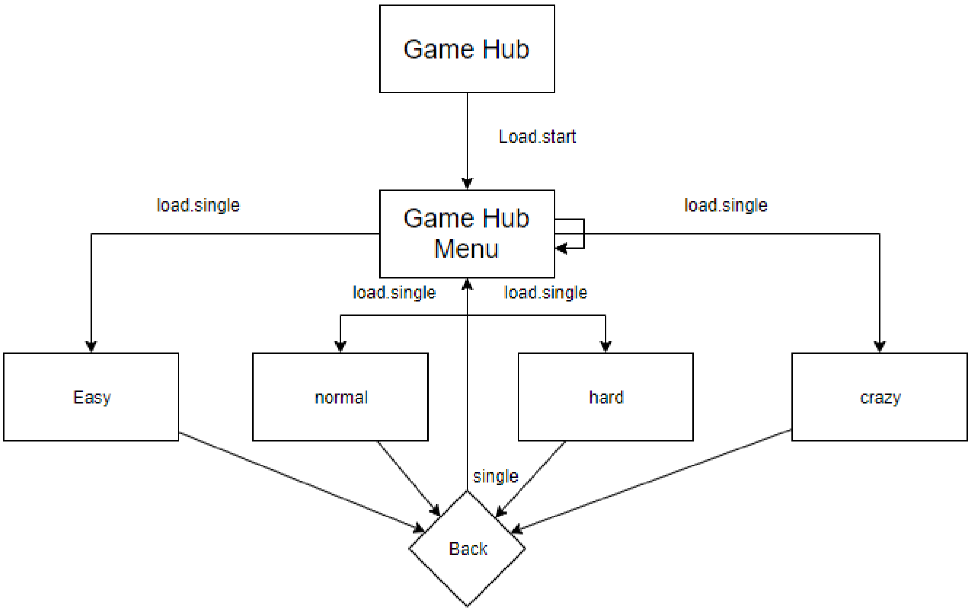
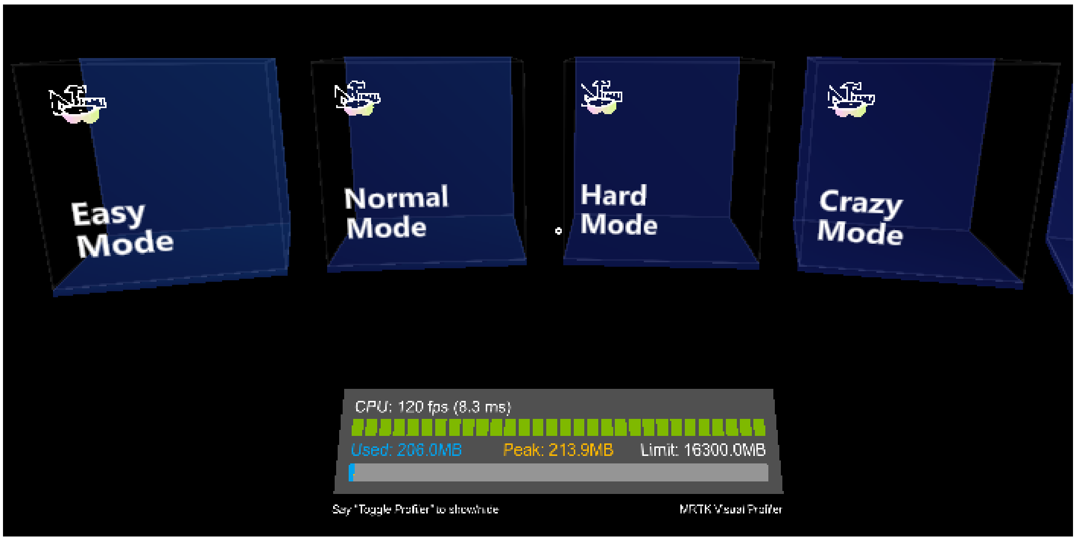


Abbildung 2.4: Scene Loading

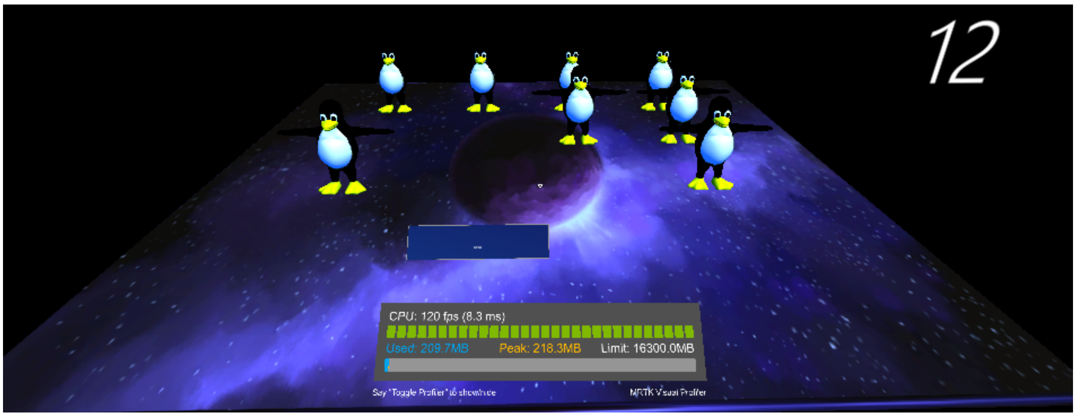
Manager Scene: A single scene with a required MixedRealityToolkit instance. This scene will be loaded first on launch and will remain loaded for the lifetime of the app. The manager scene can also host other objects that should never be destroyed. This is the preferred alternative to DontDestroyOnLoad.
In App: Game Hub
Content Scene: These are the scenes you're used to dealing with. Any kind of content can be stored in them, and they can be loaded or unloaded in any combination.
In App: Game Hub Menu, Easy, Normal, Hard, Crazy



CPU: 120 fps (8.3 ms)
Used: 206.0MB Peak: 213.9MB Limit: 16300.0MB
Say "Toggle Profiler" to show/hide MRTK Visual Profiler

Abbildung 2.5: Menu

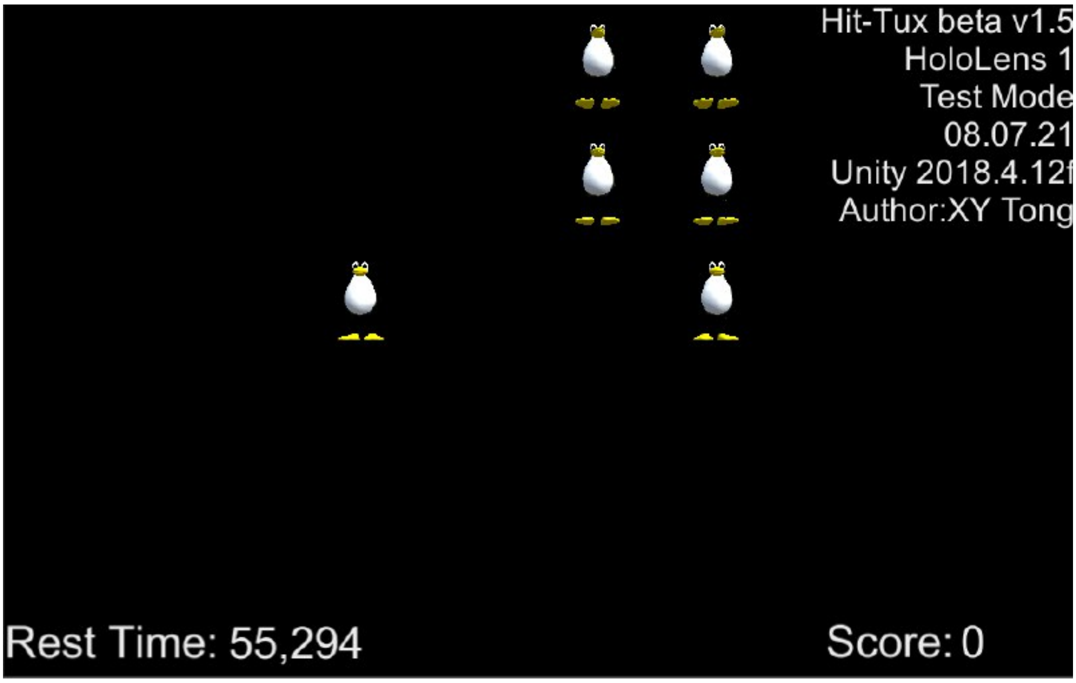
The SceneOperationProgress value is the average of all current async scene operations. At the start of a content load, SceneOperationProgress will be zero. Once fully completed, SceneOperationProgress will be set to 1 and will remain at 1 until the next operation takes place. Note that only content scene operations affect these properties.



CPU: 120 fps (8.3 ms)
Used: 209.7MB Peak: 218.3MB Limit: 16300.0MB
Say "Toggle Profiler" to show/hide MRTK Visual Profiler

2.6 Visualization of the application in HoloLens

This chapter is related to how to implement the iconic interface for the HoloLens, which involves various information and performs an analysis of objects in the room.



Once the project and corresponding scene are properly configured, the first thing to add is a Canvas object to the scene to use as a surface to write on. By default, the Canvas will be locked to the world space. To configure a camera-locked view, select the Canvas and examine its properties in the inspector window. Then in the render mode filed of the Canvas, select **Screen Space - Camera** in the drop down menu and drag the **Main Camera** into the Render Camera filed of the Canvas. This tells the Canvas which camera perspective it is locked to. As a result, the terminator vision will be permanently locked to the display of HoloLens.

2.7 Issues and possible solutions

1. Debug error DEP6100 in Visual Studio
The best fix is to remove all of the SDKs you installed, including the HoloLens Emulator, then install the Windows 10 SDK first.
2. Unity Build Error: "Cannot build player while editor is importing assets or compiling scripts"
build project with the latest SDK version.
if not working, try to delete the files within **using UnityEditor** in your scripts folder

3.Error:MSB3774 Could not find SDK "WindowsMobile, Version=10.0.17763.0"
This issue belongs to Unity (Unity is in charge of generating those project files when you export your game).Please go to Unity-Bug-Reporting to continue tracking the issue.
As a workaround, we identified that removing the reference to the WindowsMobile assembly (directly from the csproj or using Visual Studio) or reinstalling Visual Studio without Unity running could help solving the issue.(By MSFT)

3 Conclusion

The Unity engine is powerful to develop HoloLens-applications, and most processes are similar with developing other UWP (Universal Windows Platform) projects. However, there are still many minefields and special requirements that need to be noticed. The following are some of the parts to be noted by the author:

1. Main Camera must be configured according to official requirements, with solid background color and RGBA value (0, 0, 0, 0), so as to avoid blocking the actual content;

2. Gaze feature requires us to use Raycast to realize, pay attention to deal with the situation where the ray misses the target. The default gaze distance is 15 meters. If the object is not hit, a null reference exception may occur when used;

3. Gesture recognition, taking pictures and voice commands all need to use Windows-specific APIs, and spatial mapping and scene matching need to use HoloLens-specific APIs;

4. For many other details, such as visual gaze component, target area visual guide component, use them to prompt users and help users understand the application operation method, improve the user experience.

There are still many bugs and issues, which have been not solved. It is recommended to follow the latest Update-Information of Hololens by Microsoft while developing for HoloLens. Anyway, the author is look forward to seeing the new generation of Hololens and the open-source community of MRTK by Microsoft.