Staatliche Museen zu Berlin
Preußischer Kulturbesitz

museum4punkt0

Stiftung Preußischer Kulturbesitz
& Staatliche Museen zu Berlin

StorytellAR

This is a prototype developed by NEEEU Spaces GmbH and Museum4punkt0 to test different concepts on how Augmented Reality can help to develop innovative forms of storytelling in the context of guided tours in Museums.

This is a Unity Project, you need Unity 2018.2.14f1 to run the code.

ARKit is used in this project, and it works only for iOS devices which are compatible.

The repository depends on the [OSC Simpl] (https://assetstore.unity.com/packages/tools/input-management/osc-simpl-53710) library from the Unity Store. This is a paid package.

The list of compatible devices is here:

https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/DeviceCompatibilityMatrix/DeviceCompatibilityMatrix.html

Description

In this project, the devices “controller” or “controlled”. If a device is set as Controller (in settings of the device) it will control what the other devices see.

The project works using ARKit for placing the 3d models in the space and OSC for the synchronisation.

The app is build up out of multiple parts working together, the UnityARKit plugin is used to connect Unity to ARkit for tracking the phone's movement, adding the camera feed and tracking the marker images used for localisation.
Open Sound Control (OSC) is used for the synchronisation between the different instances of the app. Everything else is based on Unity's Event system to enable and disable the different states of the app. It is recommended to look through the Main scene to see the build up. Comments have been added describing which parts do what.
The different states of the app are controlled by turning on and off gameobjects that contain their respective parts. The buttons in the app send an OSC event across the network when pressed, these are only available to instances with the role of Admin.

The paid package simpleOSC is used to manage the communication among devices. It is a paid package that needs to be downloaded from the assets store.

The control of the events is done using a feature of Unity called Timeline. For more information on the Timeline, check the Unity documentation (https://docs.unity3d.com/Manual/TimelineSection.html).
Baaditrack is the name of a control track in the Timeline. You can set cue points on the timeline and switch to them via public methods. In order to understand it, please see: https://docs.unity3d.com/ScriptReference/Playables.PlayableDirector-time.html

Installing / Getting started

Install Unity 2018.2.14f1.

Download the repository from

https://bitbucket.org/neu-io/storytellar_museum40

Open the project using Unity Hub.

The project will open, and you will find more information about the parts inside.

In order to make the project work, you need to go to the asset store and get the OSC package [OSC Simpl] (https://assetstore.unity.com/packages/tools/input-management/osc-simpl-53710) library from the Unity Store. This is a paid package.

Deploying / Publishing

For building in Unity3D, for iPad.

1. Make sure the build target is set to iOS in *File -> Build Settings*.
2. In the same menu hit the *Build and Run* button.
3. In XCode select your iPad and hit run.

Might anything go awry consult the documentation here:

https://unity3d.com/learn/tutorials/topics/mobile-touch/building-your-unity-game-ios-device-testing

Using the app

In order to use the app, you need at least two devices.

1. In the controller device go in to the *Settings* app search for the app name and select the role controller for the device.
2. In the slave devices, go in to the *Settings* app search for the app name and select the role for the device.

Modes of Work

Mode 1 (Magic Mirror):

In this mode, the controlled phone places the 3d Model and animations in a point in space while still using the camera.

The controller is then able to turn it around, trigger animations and display information.

In order to control the position of the 3d models and animations, there are two ways defined in the code:

1. Through a marker. By using the image marker Assets/Marker/Marker_new.jpg
2. By hand, adjusting the position depending on the coordinates of the camera. (This is the active one) You will have to measure the angle and position of the phone to make it look realistic.

Inside of the Magic Mirror mode, you can display the model of the Hut in 3d by placing the image inside Assets/Marker/Side2.jpg in front of the camera.

Mode 2 (Guided mode):

In this mode, the 3d models are rendered in real size, and the controller phone is able to switch information and animations on or off, change lighting, highlight parts of the 3d model, or other functions specified in the controls.

The “digital” world of every device needs to share a system of coordinates. An image Assets/Marker/Marker_new.jpg currently works as marker. Both phones need to “see” the image in order to coordinate.

This image needs to be printed in the size specified in the code, when defining the marker within the ARKit features. It currently is 32cm wide.

If the tracking is lost, the devices need to “see” the marker again to recalibrate its spatial position.

Contributing

"If you'd like to contribute, please fork the repository and use a feature branch. Pull requests are warmly welcome."

Links

- Museum4punkt0 website: https://www.museum4punkt0.de/en/
- NEEEU website: www.neu.io
- Project homepage in NEEEU’s website: https://www.neeeu.io/portfolio/projects/ethnologische_museum/
- Repository: https://bitbucket.org/neu-io/storytellar_museum40

Licensing