

AR project

Alessandro Acquilino (5198425) & Flavio Barrara Stefani (5175710)

Project Idea

Our museum is designed as an interactive story centered around the myth of Prometheus, where each of the 8 paintings progressively tells the story of the myth, displaying a statue which is a character that narrates the event:

1. Prometheus Creates Man (Prometheus young)
2. The Deceit of the Sacrifice (Zeus)
3. The Theft of Fire (Hermes)
4. The Gift of Fire to Humanity (Human)
5. The Creation of Pandora (Hephaestus)
6. Prometheus Chained and Tormented (Kratos)
7. The Liberation by Heracles (Heracles)
8. The Wisdom of Prometheus (Prometheus old)

Implementation choices

SDKs & Libraries

We chose to develop the project using the XR Simulation environment because it greatly simplifies testing since we have different mobile operating systems that would have required dealing with different build targets.

This, however, meant that we had to build a proper virtual environment to simulate a plausible museum (we went for a circular ancient style greek temple).

We only really used two packages besides the default ones:

- AR Foundation for the api.
- Text Mesh Pro for the text in the 3d environment.

Implementation details

We have 5 custom scripts on top of the default AR Tracked Image Manager used for image recognition and AR Plane Manager for floor recognition:

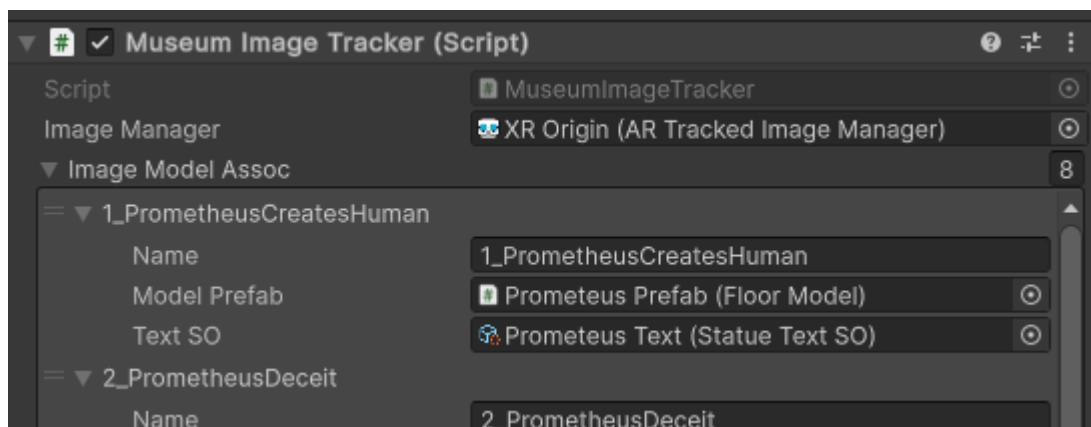
1. Floor Model

It is assigned to the model and it's responsible for keeping it fixed on the floor (once an appropriate spot is found) with the correct orientation.

2. Museum Image Tracker

This script has two roles:

- It contains a list of NameModelAssoc, which represent the associations between the FloorModel itself with its text (StatueTextSO) and a name. It allows to easily add or modify existing statues.



- It receives the `trackablesChanged` event from the `ARTrackedImageManager` and, if a new image was found, it spawns the appropriate model in the list based on the name of the image found.

3. Floor Detection Provider

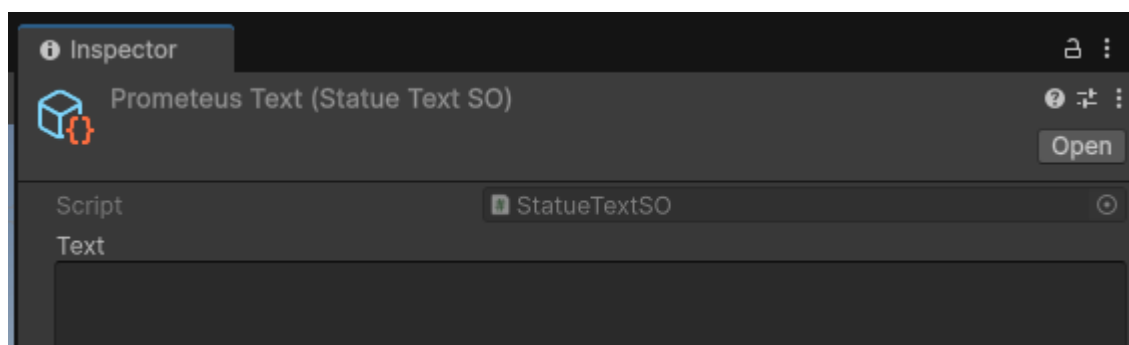
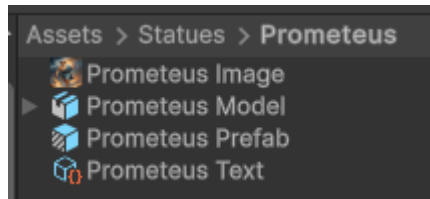
This script allows the statues to use the `TryFloorAtCoords` method to check if an appropriate point was found on the tracked floor to display the statue's model. The point is found by trying to cast a ray from above the desired point on each tracked plane. If an hit is found the point is returned as an out parameter.

4. Dialog

It is responsible of keeping the dialog face the direction of the user. The text is not displayed all at once but each character is added one at the time until the entire text is displayed; the text is updated only if the user is actually watching it.

5. Statue Text SO

It's a Scriptable Object (hence the SO in the name) which is a special class in Unity that can be used to store data in assets within the project's folder. It is very useful to define persistent data, in this case the statues' text, which can be conveniently assigned from the editor in the `MuseumImageTrackes`'s list of `NameModelAssoc`.



User Instructions

The simulation starts facing the direction of the first painting (with a "Start Here" text too), you will see the floor detection debug area in orange progressively expanding by looking in "new" parts of the temple. To "activate" a painting you have to walk near it, look at it, then move and look to its right so that the statue spawns once the floor is detected. In order to make the text progressively appear you'll have to look to the dialog, if you look away, it will stop spawning.