EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

Summer Student Report

Development of Object Detection Features for Tangible Table Application

Author:
Jordi Alfonso
IR-ECO-EXH

Supervisors:
Iliana Tatsi, IR-ECO-EXH
Zornitsa Zaharieva,
IR-ECO-EXH

Abstract

This project exclusively focuses on developing multiple object detection using tangible markers and the integration of object content from local device files. Utilizing the Tangible Engine framework, these features were developed to be used in an upcoming educational application for exhibitions.

8 September 2023

Contents

1 Introduction

The Tangible Engine framework is a software system that facilitates the development of tangible user interfaces by enabling the recognition and interaction with physical objects in digital environments. It provides real-time tracking and interpretation of physical markers or objects, creating a bridge between the tangible and digital worlds. [1]

The MediaLab group at CERN, using the Tangible Engine framework and Unity, is currently developing an interactive educational game for exhibitions with the intention of explaining the components inside ATLAS and their functions.

The game consists on two phases: The explanation of the components and then a puzzle game to put those objects where they belong on ATLAS.

On the first phase, the user had to put the specified object on a marker on the table in order to get the information of that respective component, while other objects were rejected. When the user extracted all the desired information, they could go to the next step and repeat the process with the next object. This method, while functional, was unnecessarily tedious for the user. Instead of this approach, I implemented a feature where you would use a single marker to get the information of all components at the same marker.

Then, in order to be able to generate the content for those components without having to code it in Unity, I worked on a feature to generate their content (Text, images and videos) using only files inside local folders on the interactive table's device.

This report will focus on the implementation and further improvements to be made on these new features.

2 Detection of Multiple Tangible Objects using a Single Marker

All the process of object detection and content display it's done within a single Unity scene, where the user navigates through multiple slides inside the game. As seen in Figure 1, the slides in question are contained within an object. We are going to focus on Slide 2.

The objects and the content within them are contained on the "Objects" element, and each component's content are located on the "Content" element inside each "Objects" sub-element.

In order to access the correct content depending on the object introduced on the marker, a Unity script detects the ID of the "Objects" sub-element, using the number or its name (ex. Object01 has ID 01). Then, the script compares this ID with the one of the physical object detected on the marker. If both objects' IDs are the same, then the "Content" object gets activated, displaying all content options for the user to interact with. When the user removed the physical object, the Content" object gets deactivated.

"Option" objects inside the "Content" object are the different types of content that can be seen within a component, if the player rotates the physical object when it's placed on the marker, then the triangle-shaped cursor on the marker (As seen on Figure 2) will rotate, allowing the selection of the different displayed options.

Figure 1: Hierarchy of the Unity scene

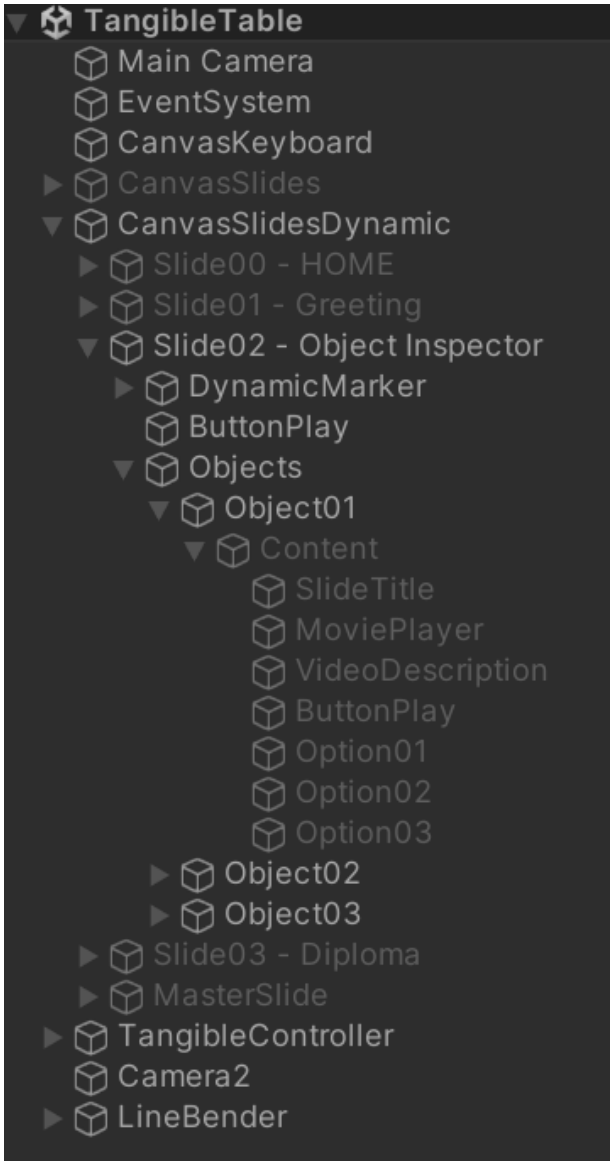
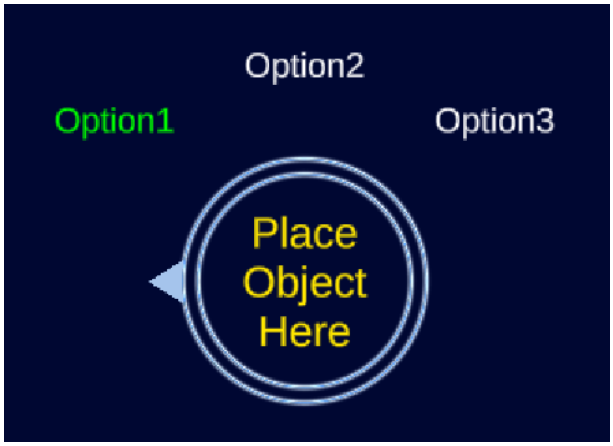


Figure 2: Visual representation of the Marker in the game



3 Dynamic Content Loader for the Tangible Objects

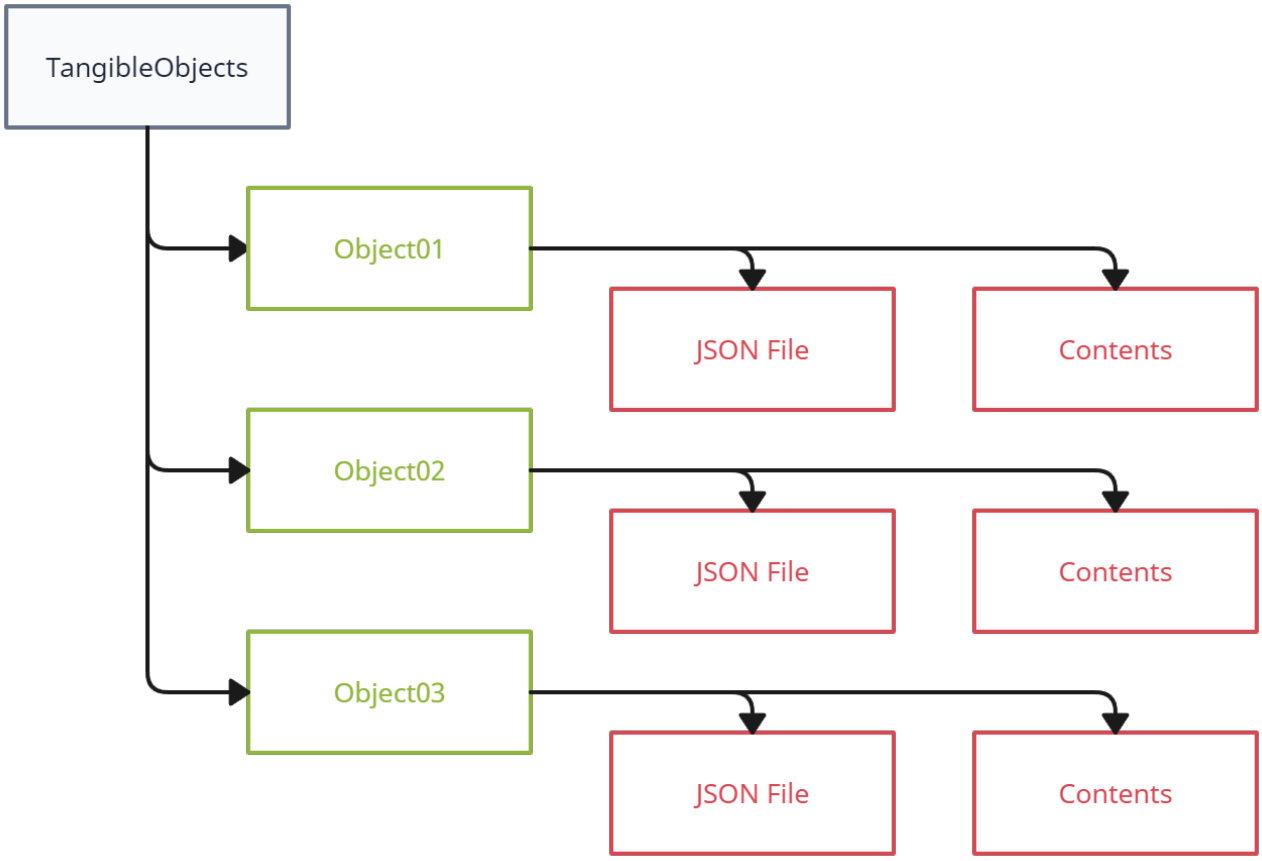
3.1 Architecture of the System

All the media of the components will be located on a folder on the C: Drive called TangibleObjects.

To load the contents of each object, the user will introduce the Text files, images and video files that they want to load within a sub-folder inside TangibleObjects. This subfolder will have the name of the component it's trying to import to the game.

Each subfolder will also feature a JSON file defining which option each file within the sub-folder is allocated. Multiple files can be used for the same option. The order they are introduced on the JSON determines the order they are loaded on the application.

Figure 3: Hierarchy of the folders on C: Drive



3.2 Implementation

In Unity, there's a script that will look into each sub-folder of TangibleObjects and read their JSON file. After this process is done the project will feature two Unity GameObjects:

TangibleOptions: Object where the content paths of a subfolder will be stored, along with the option value allocated to each.

TangibleObject: Object that features a list of TangibleOptions of a subfolder.

At the end, there will be an array of TangibleObjects, where this will be instantiated on the "Objects" list of the hierarchy explained on the previous section. This process will occur on the Start() function of the same script explained on the previous section, ensuring that the object detection will only happen after all content have been imported.

3.3 Missing Features and Potential Improvements

At this moment, while the structure of the feature is designed and a lot of it has already been implemented into the project, There's still a script missing to generate and import the "Option" objects within a "Object" sub-element to the game.

As per improvements that could be done to the implementation, It's worth noting that in the future, there could (and it should) be a method for both the media extraction and the object detection to use the component's name instead of an ID value, since that would enhance the process of creating a new Tangible element and defining the sub-folder and the files inside for that element.

4 References

1. *Tangible Engine - Object Recognition Software for Multitouch Tables.* (n.d.). Tangible Engine - Object Recognition Software for Multitouch Tables. From https://www.tangibleengine.com