HOME STUDIO DESIGNER: AN AUGMENTERED REALITY APPLICATION FOR DESIGNING YOUR MUSIC RECORDING STUDIO

Nshuti Jean-René
56955@etu.he2b.be

RÉSUMÉ

This report covers the creation configuration and implementation of an application using Unity 3D to help users design their own home music recording studio. The application uses Augmented Reality and machine learning technologies to allow users to reimagine their rooms at home by placing and manipulating life sized music studio equipment and furniture. All 3D models were downloaded from free online sources stated in the bibliography. This application leverages cloud technology to store and serve 3D models to users.

Home Studio Designer is an extension of the AR furniture app created by Priyanshu-CODERX. [1]

Keywords: AR; Furniture APP, recording studio designer; Unity; 3D

1. INTRODUCTION

Home studio Designer is an Augmented Reality (AR) application inspired by the ever more present furniture apps readily available on the market. As part of my university course Game development, we were tasked with implementing Unity's 3D tools as well as Machine learning (ML) algorithms to bring to life innovative AR/VR applications. Home Studio Designer was built on AR foundation with AR Core for Android and leverages popular pathfinding algorithms like BFS and A* as well as vocal recognition to search for furniture models influenced by user preferences.

2. OBJECTIVES

The aim of Home Studio Designer is to allow a user to design and imagine their home recording studios by visualizing how different soundproofing panels/absorbers look when placed on their walls, how certain musical instruments will look in their studio and trying out new music production equipment on their desks, racks, etc. Users will also be able to place furniture like sofas and chairs to see how these fit in their new recording studio.

Eventually, Home Studio Designer can leverage even more powerful Machine learning techniques to not only suggest models based on user preferences and but also to provide users with reliable acoustic treatment suggestions based on room dimensions, open space, and popular music studio design techniques.

3. PRESENT STATE OF AFFAIRS

3.1. Pre-requisites

- Unity Editor 2022.3.6f1
- Visual Studio Editor 2.0 and
- Blender 3.6
- Dependencies:
- Addressables 1.21.15
- AR foundation 5.0.7
- Google ARCore XR plugin 5.0.7
- XRInteractionToolkit 2.4.3

3.2. Functionalities

In the start-up scene, a user can:

- Select their preferences in terms of what kind of models they wish to place in the scene.

In the main scene, a user can:

- Select a furniture model from menu and place 3D the model in the room in AR,
- Select rotate and move p2D models that are currently placed in the room,
- Decide to visualize or hide detected planes in the AR scene,
- Reset the AR scene by deleting all placed AR objects,
- Download more 3D models that are currently available in my google bucket,
- Search for 3D models via textual input or vocal input.

4. PROJECT ARCHITECTURE

This app uses the library Addressables and scriptable objects to save 3D furniture models in the cloud. I configured the cloud using a free Google Cloud account allowing me to create the Home Studio Designer project and the storage space (called "Bucket" in the Google Cloud ecosystem) and named it "hsd-bucket". When the main AR scene is loaded, the app automatically connects to the bucket and downloads 9 furniture models for the user to use and search. The user can download more

models by clicking the download button until all models are available for placement in the AR scene.

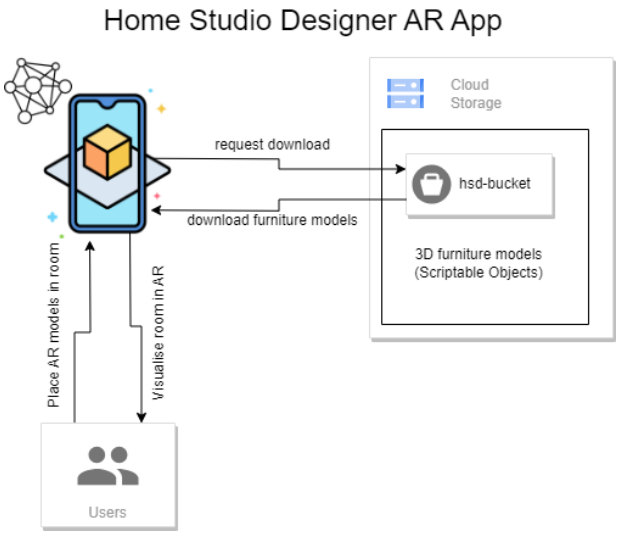

Figure 1: Home studio designer project architecture

4.1. Unity Scenes, hierarchies, and folder structure

In this section I present the two main scenes: start-up and main scenes that were implemented in Home Studio Designer.

4.1.1. The start-up scene

The start-up scene is a simple question and answers UI canvas with a Singleton Game Object named "User Preferences DataHandler" that stores the user's furniture room styles and preferences that feed into the search algorithms implemented in the main scene.

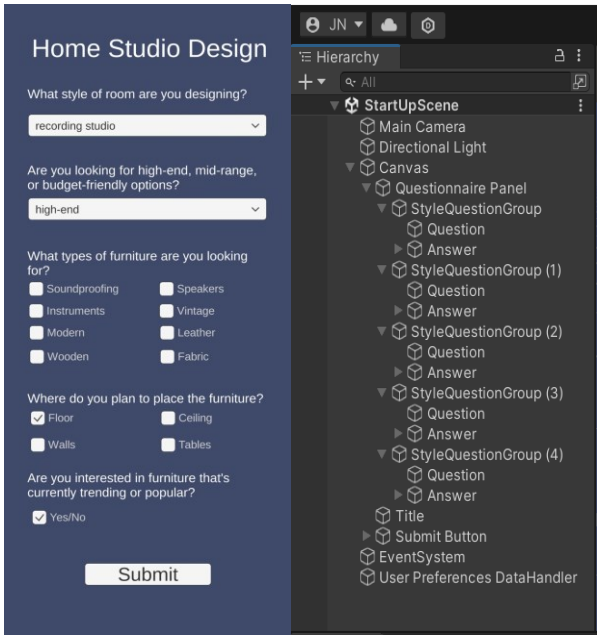

Figure 2: Start-up scene and Unity hierarchy

Figure 2 displays the Game view of the start-up scene of Home Studio Designer. The start-up scene collects user information that is used primarily when a user conducts a search for AR furniture models. The hierarchy (as seen on right hand side of Figure 2) contains Unity User Interface Game Objects that contain TextMeshPro text fields for the question Game Objects, dropdown menus and toggle buttons for the answers Game Objects.

4.1.2. The main scene

The main scene is the AR scene that leverages AR foundation and XR interaction toolkit to place, select, move, and rotate objects in the user's room using the phone's camera. A user can place 3D models on the floor and/or walls in this scene. At load, the user is presented with nine initial models but can also download more models on demand (By clicking the download button at the bottom right of the screen, c.f. Figure 3). When the user conducts a search, Home Studio Designer conducts an A* (or BFS) pathfinding search to suggest 3D models to the user. The shortest path is then printed as a list of possible items that match the search input text. It is possible to search through textual input or through speech recognition.

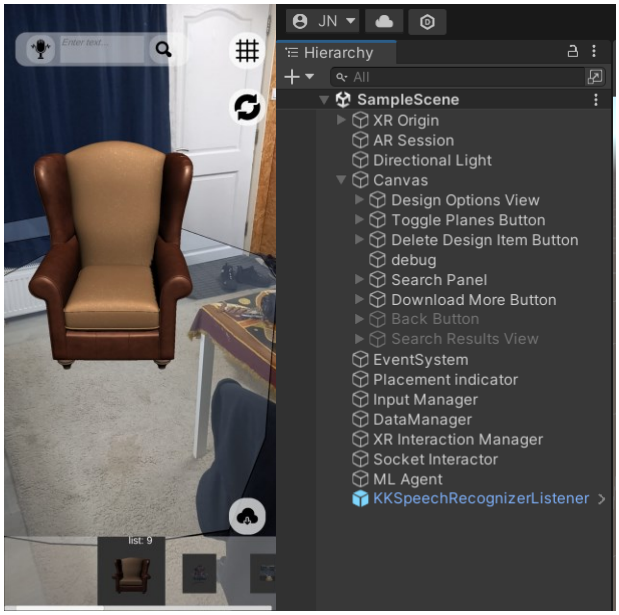

Figure 3: Main scene and Unity hierarchy

5. DEVELOPMENT PHASE

5.1. Unity 3D AR set up.

As stated earlier, this project uses AR foundation and XR interaction Toolkit. I based my set up on the reference project [1]. The right-hand side of Figure 3 showcases the AR Session and XR Origin Game Objects that are necessary to set up an AR scene with AR foundation. I also attached two components: AR Raycast Manager with default settings and an AR Plane

Manager to which I attached an AR plane prefab inspired by the reference project.

Home Studio Designers is currently an Android only application based on AR Core but is easily portable to iOS devices.

5.2. 3D tooling

The Assets folder of the Unity project contains two subfolders named "Models" and "Prefabs". The "Models" folder contains all 3D models that I downloaded and edited using Blender before importing the FBX files into Unity. The "Prefabs" folder contains the prefabs I had to create to convert the read only models into interactable AR models. The figures below aim to illustrate this process. Figure 4 below displays an example of an FBX model exported from blender and imported into Unity with their respective textures and materials.

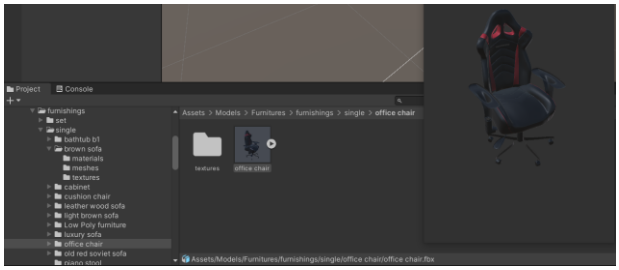

Figure 4: Example of an FBX asset imported into Unity in the "Models" folder.

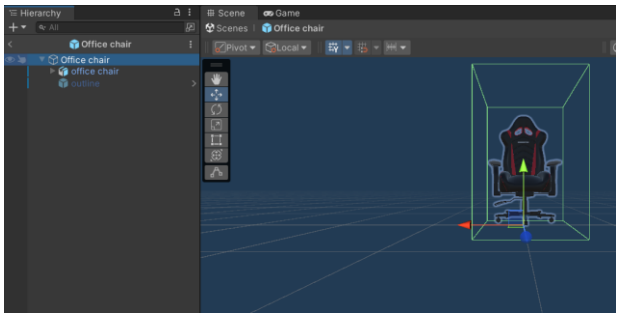

Figure 5: Example of a Unity Prefab created from a given FBX model in the "Prefab" folder.

Figure 5 is an example of one of the unity Prefab Game Objects that I created using the models downloaded from various free online web sites like Sketchfab, 3dmdb, and others [2-4].

All prefab Game Object have a FBX model as a child and an invisible 3D object named "outline". The Game Object "outline" is a semi-transparent 3D cube object used to visualise furniture selection and also referenced by XR interaction Toolkit scripts for selection, translation and rotation. The XR interaction toolkit scripts that are attached to each prefab are:

- AR Selection Interactable
- AR Translation Interactable
- AR Rotation Interactable and a
- Box collider necessary for interaction with the user's touch.

I managed to collect a total of 152 3D models organised in two global folders: "recording studio" and "furnishings". The "furnishings" folder is further subdivided by packaging, whether in the form of a set or as a single 3D model. The "recording studio" folder is subdivided into 6 folders: headphones, instruments, pictures, screens, soundproofing, and speakers.

5.2.1. Storing and accessing the models

The prefabs and models described in the previous section are referenced and called from scriptable objects when they are instantiated in the AR scene. The scriptable objects are also responsible for holding the characteristic data of each 3D model. This data was necessary to respond to the machine learning requirements of this project which will be covered in the following section. Scriptable objects are instances of the class "DesignItem" found in the Assets/Scripts/UI folder that inherit from the "ScriptableObject" class. The following figures aim to illustrate how scriptable objects are saved in the Google bucket (cloud storage).

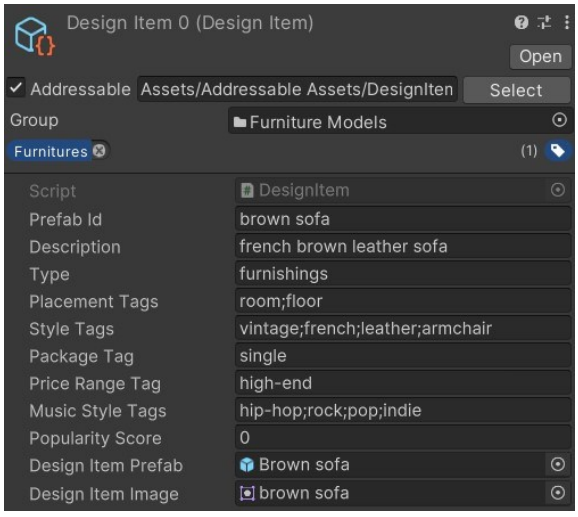

Figure 6: Example of an addressable Scriptable object.

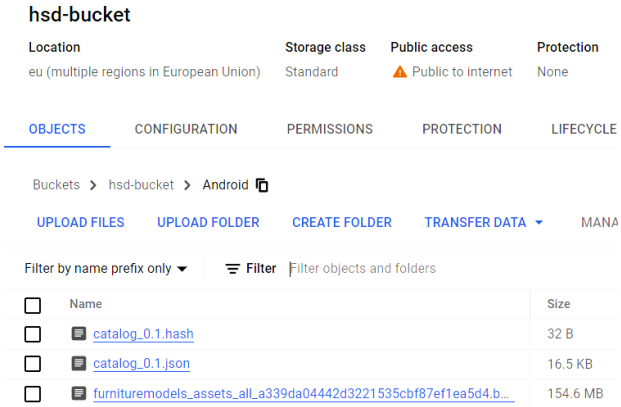

Figure 7: Addressable data storage of scriptable objects in the Google Bucket

Figure 6 displays an example of an addressable object holding the data of each model in the form of attributes and holds a reference to the relevant Unity prefab and User Interface image. Figure 7 shows the files generated by the Addressables package wherein all "DesignItem" objects are stored and where they are downloaded from when using the Home Studio Designer main scene.

5.3. Machine learning

The initial aim of this project was to use a genetic algorithm to suggest 3D models to users based on their personal preferences, room dimensions and popular music recording studio designs. However, generating the data necessary to implement an efficient genetic algorithm proved too complex with the time constraints for this project. I thus focussed on implementing pathfinding algorithms like the BFS and A-star algorithms as well as implementing a speech recogniser to facilitate searching for models for users. To create the search graph, I used the data set I created for the 152 models I collected. Figure 6 is an example of the data saved for each model.

The two key components that drive Home Studio Designer's machine learning functionality in its current state are the "FurnitureGraph" class and the integration of a Speech Recognition Asset.

5.3.1. FurnitureGraph: A Graph Representation of Design Items

The "FurnitureGraph" class forms the core of the pathfinding system. It serves as a structured representation of design items, connecting them in a graph that captures spatial relationships and attributes. Here's an overview of the key functionalities within the "FurnitureGraph":

1. **DesignItemNode Creation**: Each 3D model in the dataset is represented as a DesignItemNode, encapsulating attributes such as position, dimensions, and popularity score. This allows us to consider each design item as a node within the graph.
2. **Connections and Attributes**: I establish connections between DesignItemNodes based on proximity and other relevant attributes. These connections are crucial for defining possible movement paths in the AR environment.
3. **Heuristic Evaluation**: To enable efficient pathfinding, I integrate a heuristic evaluation mechanism. By incorporating factors such as popularity and spatial relationships, I can estimate the potential desirability of certain paths.
4. **A* Algorithm**: I implement the A* algorithm, a state-of-the-art pathfinding technique, to navigate the graph efficiently. The algorithm prioritizes paths based on a combination of movement cost and heuristic evaluation, enabling us to find optimal routes. The heuristic functions that were tested were the edit distance and cosine similarity algorithms to check the similarity between two strings. In my implementation I used the Prefab Id which is effectively the name of the model to test for similarity.

5.3.2. Integration of speech recognition asset

Incorporating machine learning capabilities into the AR system further enhances user experience and interaction. I integrated the Mobile Speech Recognizer asset available on the Unity Asset Store. This asset empowers users to issue voice commands to manipulate and navigate the AR environment. Here's how the integration works:

1. **Graph-Based Actions:** Upon receiving a voice command, the system leverages the Furniture Graph's optimized pathfinding to determine the best way to fulfill the user's request. This involves evaluating paths, considering attributes, and displaying the path as a list of possible furniture models to be placed in the AR scene.
2. **Seamless Interaction:** The combination of graph-based pathfinding and machine learning-driven speech recognition ensures that users can interact intuitively with the AR environment. They can effortlessly rearrange furniture models to create the desired spatial layout of their home recording studio.

6. REFERENCES

[1] Priyanshu-CODERX. "Free Augmented Reality Course". July 2018. Youtube cours. https://www.youtube.com/watch?v=FJAO6jDYljs&ab_channel=freeCodeCamp.org.

[2] Sketchfab. https://sketchfab.com/feed. Accessed 28 July 2023.

[3] 3dmdb. https://3dmdb.com/. Accessed August 2023.

[4] Cgtrader. https://www.cgtrader.com/. Accessed August 2023.

[5] Zmudzinski, Piotr. "Mobile Speech Recognizer", Unity Asset Store. https://assetstore.unity.com/packages/tools/audio/mobile-speech-recognizer-73036. Accessed 8 August 2023.