



TECHNICAL DOCUMENT

**Centre for Digital Media
Term 2 Industry Project**

APRIL 8, 2019

TEAM CDM PINNACLE

Abe Nito
Christopher Pien
Daryn Mulholland
Fan Wu
Gabriel Reis

Table of Contents

About This Document	3
Overview	3
Prototype	3
Unity	4
Plugins and Assets	4
AR Foundation	4
Mapbox	5
Firebase	5
Instagram API	5
Native Gallery	5
Native Share	6
Text Mesh Pro	6
XR Remote	6
Asset Structure	6
Code Structure	7
Firebase Database	8
Features and Functions	9
Map View	9
Physical Location	9
Instagram API integration	9
Scene Creation	9
Photo Creation	10
Captioning	10
Object Selection	10
Object Placement	10
AR Orientation	10
Media Playback	11
Like	11
AR Capture	11
Downloading	11

Sharing	12
Limitations	12
Known Issues and Challenges	12
Further Development Considerations	13
Log In Screen and Authentication	13
Video Recording	13
Push Notifications	14
Common Surfaces	14
Limiting Pins	14
Time Limited Mementos	14
Movement/Rotation Gizmo	15
Rim Lighting	15
Map Rotation and Compass	15
Ghosting	15
Color Selection	15
Frames	16
User-uploaded models	16
Editing After Posting	16
Comments	17
Gamification	17
Target Platforms	17
Android and iOS	17
Development Tools	17

About This Document

This document serves to provide the reader an overview of the high level implementation and project structure, functionality, features, use cases, limitations, areas for future expansion, and other technical details of Pinnacle Production's *Memento* project.

Overview

Memento is an Augmented Reality (AR) experience that emphasizes geo-social interaction. Our client, *Truly Social Games*, gave us the high level task of creating a geo-social AR mobile app to enhance the way people interact and connect with each other. After some significant ideation, we decided on *Memento*, which strives to capitalize on the social and emotional connections bound to real life moments, and encapsulate them into a cohesive smartphone experience.

Prototype

The current prototype of *Memento* allows users to drop a *Memento* AR “pin” at their current geolocation that other users (including the creator of said pin) can then see and interact with.

Explore

When a user goes to that location in the real world, they will be able to open, view and interact with the content contained within the pin.

Create

The content, created by the pin “dropper” consists first and foremost of a photo, which can be taken at the scene, saved on a device or imported via Instagram. This photo manifests as a 3D AR polaroid object, which can be customized with a variety of polaroid skins, as well as a caption. After that, the user will create the remainder of, what we refer to as, the “scene”.

This “scene” includes other AR objects, like a party balloon, a pizza, 3D emoticons, a soccer ball, etc. Such objects can be scaled, rotated, and repositioned in 3D space using intuitive touch controls, so as to give the user full control of designing their “scene” to their liking.

Interact

Some of these AR objects can be interacted with as well, like a party horn that shoots confetti and a gramophone that plays music.

Share

In the end, users are able to “like” *mementos* as desired, take photos of their own personal experiences with the AR artifacts, and “share” them on various Social Media platforms as well.

Unity

Unity is a cross-platform game engine produced by Unity Technologies ApS. It allows developers to begin development for free, only requiring a licence to remove the “Made with Unity” splash screen or when profits from the application exceed \$100,000 USD. Unity allows developers to seamlessly develop for multiple platforms ranging from consoles, to PC, VR and mobile.

Plugins and Assets

AR Foundation

AR Foundation is a built-in multi-platform AR tool for Unity. It uses both ARCode and ARKit packages to create a interface that switches between the two AR tools, depending on the target platform.

Memento uses the *Plane Recognition* feature to search for a ground plane. Upon recognition, it places 3D objects over this plane. The tool also supports *Light Estimation*, which creates a light system based on the environment lighting.

For more information please visit [Unity AR Foundation Manual](https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@1.0/manual/index.html):
<https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@1.0/manual/index.html>

Mapbox

Mapbox SDK provides geolocation tools that allows displaying real maps based on the user’s location, stylizing maps, search for locations and much more. The SDK is fully integrated with Unity, and the free subscription grants up to 50,000 monthly active users. Memento uses a default map view provided by the SDK, as well as the pin feature, which support placing pins in a specified location.

For more information please visit [Mapbox SDK](#):

<https://www.mapbox.com/maps/>

Firestore

Firestore is a mobile and web application development platform. It offers several web service tools, such as database, cloud storage, authentication, and others. Some of its features are also supported in Unity.

Memento uses the Real-time database feature to save and load mementos from the server. For being completely web-based, Firestore is supported for iOS and Android.

For more information please visit [Firestore](https://firebase.google.com/):

Instagram API

The *Instagram* API allows retrieving metadata from Instagram posts of a pre-registered account. The number of posts retrieved is limited to 20, and the metadata includes: time, photo, location, number of likes, caption, comments and more.

Memento uses the API to push pictures, captions and locations from the user's posts and create mementos at those locations.

Native Gallery

A free asset in Unity asset store that allows users to save images in the device's gallery. Available for both Android and iOS.

For more information please visit [Native Gallery](https://assetstore.unity.com/packages/tools/integration/native-gallery-for-android-ios-112630):

Native Share

A free asset in Unity asset store that allows users to share images in all the device's compatible apps, including *Instagram*, *WeChat*, and *Facebook*. Available for both Android and iOS.

For more information please visit [Native Share](https://assetstore.unity.com/packages/tools/integration/native-share-96546):

Text Mesh Pro

A built-in Unity plugin, which grants additional 3D text manipulation options for the standard Text Mesh in Unity.

XR Remote

A built-in Unity package, which allows a remote communication between an Android device and Unity, in order to more easily test the AR camera, and making it possible to debug AR features more precisely.

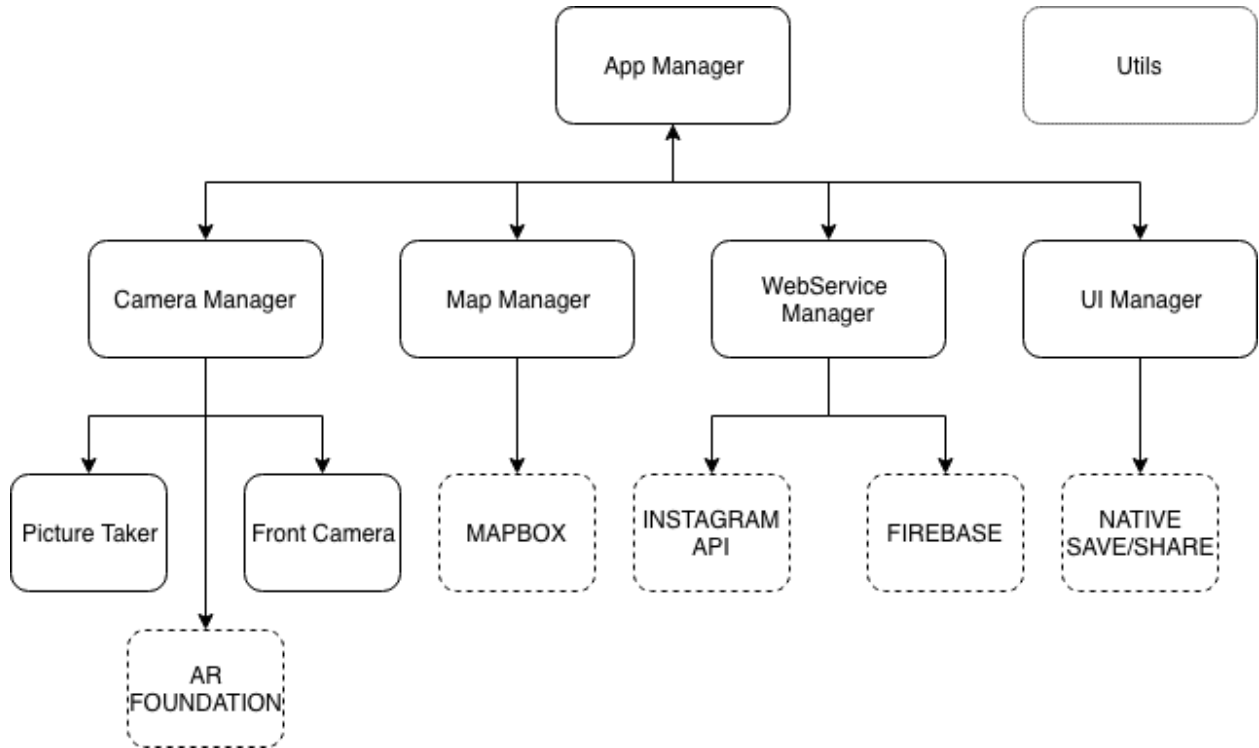
Asset Structure

The Unity project Assets folder is organized with the following subfolders:

- **Animations:** All the UI animations presented in the app
- **AR:** AR Foundation required files
- **Editor Default Resources:** Firebase resource file
- **Firestore:** Firestore required files
- **Fonts:** Font files and textures used in the app
- **Mapbox:** Mapbox required files
- **Materials:** Material used in memento prefab
- **Models:** All 3D models used in the app, along with their specific materials and animations
- **Parse:** Firestore required files
- **PlayServiceResolver:** Firestore required files
- **Plugins:** Additional plugins used in the app
- **Resources:** Assets used for debugging to replace real mementos
- **Scenes:** The single scene used in the app
- **Shaders:** Shaders created for the 3D models
- **StreamingAssets:** Assets used by AR foundation remote (debugging)
- **TextMeshPro:** TextMeshPro required files
- **Textures:** UI Textures used in the app

Code Structure

The following diagram shows the hierarchy of the various classes for the developed application.



App Manager holds the current state of the app and handles all communication between the other managers, ensuring a clear modular hierarchy between the classes. This approach allows multiple developers to work in different parts of the app without interfering with each other's work.

Utils class saves all constants for the app.

Firestore Database

All database calls are handled by WebServer Manager and the data is formatted as JSON. Currently, the Firestore server holds two databases: Mementos and Users.

The Memento database has all the information necessary to reproduce the saved Memento in any other device running the application. It is saved with a unique ID represented as an integer number, and with the following fields:

- Coordinates: latitude and longitude;
- Image: the jpg image encoded in base64;
- Message: caption;
- Pageentry_code: the frame's code (the codes can be identified in Utils.cs);
- Scene_objects: information regarding the AR objects that compose the scene around the polaroid. Each element is saved with the following fields:
 - Ar: object code (the codes can be identified in Utils.cs);
 - X and Z: represent the coordinates relative to the polaroid;
 - Hgt: object's height;
 - Scl: object's scale;
 - Rot: object's rotation.
- User_id: ID of Mementos owner.

```
538
  coordinates: "49.26697, -123.089271"
  image: "/9j/4AAQSkZJRgABAQAAQABAAD/2wBDAAgGBgcGBQgHBwc..."
  message: "Welcome to CDM!"
  pageentry_code: "2"
  scene_objects: "{\"ar\":5,\"x\":-0.3504401,\"z\":0.03397969,\"hgt\":0.5..."
  user_id: "gabe"
```

The Users database has the following fields, beside the unique ID:

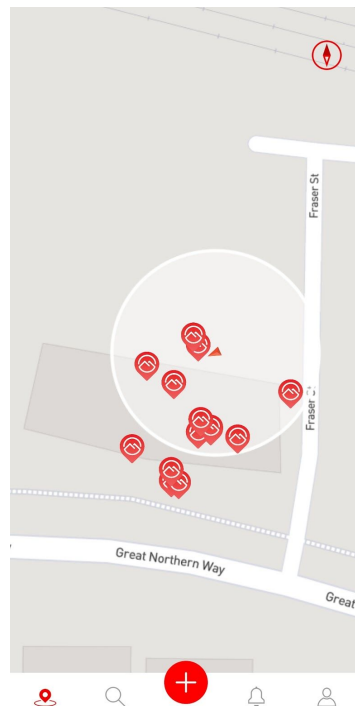
- User_name: user public name
- Pw: account password

```
gabe
  pw: 123
  user_name: "gabe"
```

Features and Functions

Map View

Memento allows users to view a map, and see their own location on the map. In the map they can pan and zoom so as to focus on the location of their choosing.



Physical Location

Mementos are placed at the user's current physical location, highlighting the geospatial aspects of the application.

Instagram API integration

Memento can pull photos from a user's Instagram account and then place them into the world based on the location metadata retrieved from Instagram.

Scene Creation

Users are able to create entire scenes consisting of a polaroid with a frame and caption of their choosing, surrounded by 3D objects. This is limited to a one meter circle around the center of the polaroid.

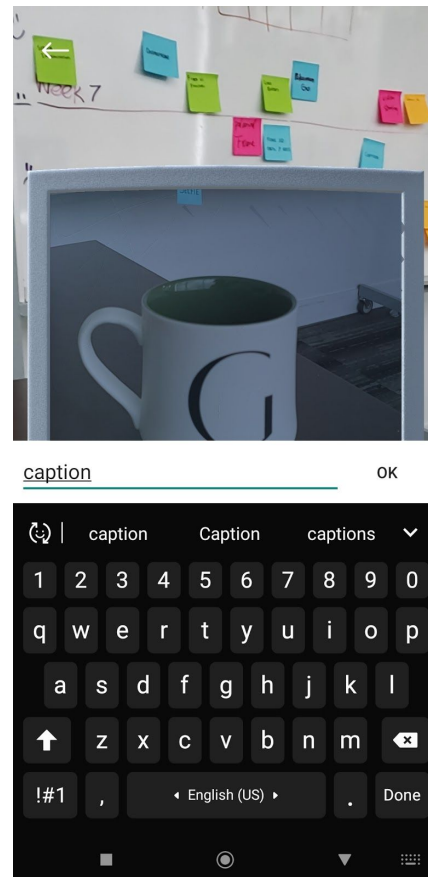
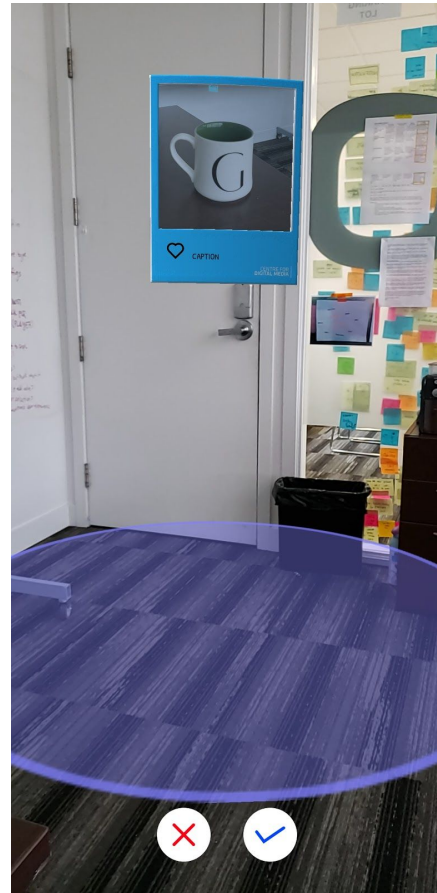
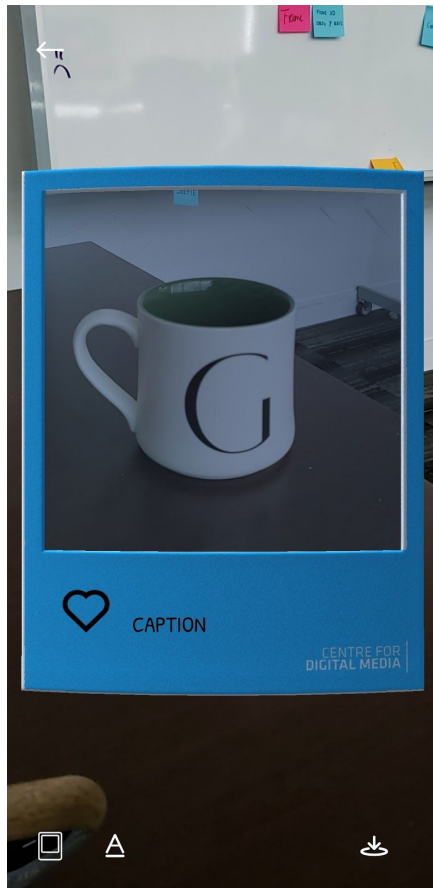


Photo Creation

Users are able to take photographs through both the front-facing and rear-facing cameras. Additionally they can choose a photo that is already on their device via the native gallery integration in its operating system.



Captioning

Users are able to create a caption that is attached to their photo of up to 50 characters.

Object Selection

While creating a scene, users can select objects in AR by long-pressing on an object. A circle icon will rotate around their finger, indicating that the long-press is occurring.

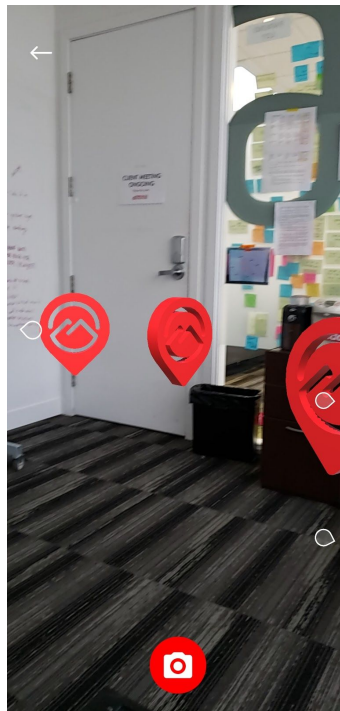
Object Placement

When placing an object into the scene, users are able to use multiple gestures to orient the object however they desire. Two finger swipes up and down allow them to move the object along the Y axis (up and down). Pinching in will scale the object down, while

expanding the pinch will increase the scale of the object. Rotating two fingers around the object will rotate the object around the Y-axis. There is currently no way to rotate around the X or Z axes, as there was no good way to select what axis you wanted to rotate an object around, and since the Y-axis would be the most common, it was selected as the default.

AR Orientation

When in AR view mode, semi-transparent “drops” appear around the edges of the screen. These point towards nearby memento pins, and scale in size, increasing as the phone is rotated closer to a memento pin, eventually fading out of existence as the 3D model of the pin enters the field of view. Currently, this is limited to pins within 90 degrees of the current position of the phone, as there were irresolvable issues when the range was broadened beyond 90 degrees, where the markers would rapidly flip from side to side, or seemingly point at non-existent pins.



Media Playback

Memento makes very limited use of audio. Currently, it is capable of playing any audio file that can be played through unity, In the current prototype, these are all Creative Commons licenced or Public Domain .mp3 or .wav files.

Like

Users are able to tap on the heart that is part of the memento to like a memento. Currently this is only stored locally rather than on the server, but the server could easily be expanded to include it.

AR Capture

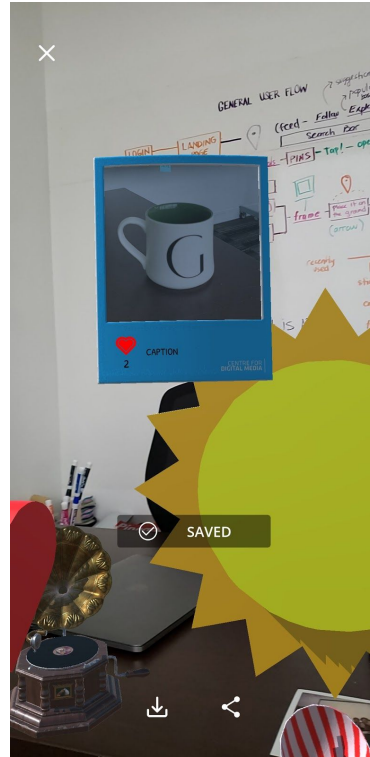
Memento is able to capture a single frame out of the AR camera and then either save it to the device or interact with it from the device's memory. Since this is captured through the AR camera, it is at a lower quality than if taken through the device's native camera, (at screen resolution rather than maximum camera output resolution and without additional software enhancements to image quality available through the stock camera) but this is offset via the inclusion of the added AR elements.



Downloading

After the user has taken a “photo” in the AR view mode, they have the ability to either download or share that image. When the user hits the download button, the phone will save the image to their internal storage/mementos/memento.png. Currently it will

override the last saved image when a new image is saved, but this behavior can easily be altered to append a number to the end of the file name, or any number of other possible solutions.



Sharing

Social sharing is done through the native Android and iOS pathways, with the feature working flawlessly on both systems. However, Android's sharing implementation provides a very large list of recommended applications to share to, many of which are not ideal, as well as offering no way to opt out of these pathways or to whitelist others.

Limitations

The version of AR Foundation used in this project (1.0.0) does not support AR in the Front Face camera, making it impossible to display AR elements using this camera. However, taking pictures from the camera is possible.

In order to use the AR features in ARCore (Android) and ARKit (iOS), it is necessary to use a version of Android 7.0 or higher, and iOS 11.0 or higher.

Firestore presents authentication issues when tested on a Macbook, making it impossible to retrieve Mementos from the server while using Unity in a Mac.

Known Issues and Challenges

Some issues were found throughout the app development, due to resource limitations the team decided to focus on those which happen more frequently or greatly hinder the user flow. The following list presents the issues not addressed by the team in the final build:

1. Map not showing on Map View
 - When: Sometimes, when using the app for the first time.
 - Possible cause: The app tries to read GPS location before user gives the permission and fails.
 - OS: Android.
2. Interaction radius does not update accordingly to zoom
 - When: Zooming in/out on Map view.
 - Possible cause: The value used to set the interaction radius scale is rounded to a Int value.
 - OS: Android / iOS
3. Polaroid AR object disappears in creation mode
 - When: Rarely, after placing polaroid in ground, during creation mode
 - Possible cause: The AR camera loses the horizontal plane reference
 - OS Android / iOS
4. AR object interaction not working when touching the object
 - When: Creation mode, touching party horn or confetti popper
 - Possible cause: The sphere collider that read the touch disappears while placing the object and not enabled afterwards.
5. Camera is unable to focus, making the acquisition of ground planes difficult
 - When: Creation and View modes on Samsung S10 variants
 - Possible cause: unity is unable to control the camera, as this issue is present on some other apps which make use of the camera on S10s.
6. UI “pointers” disappear too soon (i.e., too far before the Memento pin is in view) on iPhone SE
 - When: View mode on iPhone SE
 - Possible cause: current build not optimized for smaller screen size

Further Development Considerations

Log In Screen and Authentication

The current code for Memento has an interface for account implementation. However, it was not further developed due resources limitation. App Manager does call for authentication at the start of the app, but without any Log In screen defined, the same user is authenticated everytime, and all Mementos created are linked to it.

With the creation of an Authentication system, further social features can be developed, such as Friend list, Follow users, Liking mechanics, Comments, and others.

Video Recording

The current implementation does not have the ability to capture videos while using the app, although it should be fairly simple to do so. On the Unity asset store, there are numerous assets which provide this functionality, however they would have to be vetted and ensured to be compatible with the used AR camera, not requiring their own camera prefab to function.

Push Notifications

You could implement some sort of push notification or simply use vibration to notify the user when there are nearby Mementos to view. The one issue with this would be that should the app grow large enough, this would have to be cut down, either to posts by friends or posts with enough traction to warrant the additional attention.

Common Surfaces

While the current Memento prototype attempts to place Mementos somewhat close to their original physical location, it would be interesting if we could create common surfaces, where multiple users on separate devices could view objects in the exact same locations with the same orientations as if they physically existed in the real world. This would go a long way towards selling the physicality of AR and allow for new and interesting interactions between multiple people in AR.

Limiting Pins

Currently, there is no mechanic to limit the number of pins per area or user. However, as the number of users increases the number of pins in an area, that area can be overwhelming to the user. There are several ways to address this issue. One way is grouping close mementos to a single one, other mechanisms could include dynamic filters, such as hashtags or user preference-based.

Time Limited Mementos

We have toyed around with the idea of adding some sort of time limitation to Mementos for multiple reasons. First of all, it would serve as a mechanism to cull Mementos over time meaning that we would not continuously be adding to the total number of Mementos, resulting in both a lower server load (for saving and loading Mementos) but also a lower visual load on views, meaning that users would be more likely to fully interact with Mementos in front of them rather than glossing over and not interacting whatsoever. Secondly, this could tap into the “fear of missing out” phenomenon, where people are incentivised to interact with a platform not only for its content, but because they fear that they will miss content on it if they do not interact, boosting user interaction and retention.

Movement/Rotation Gizmo

Currently all movement and rotation is done through user gestures. As an alternative to this, we could implement a gizmo, which would allow users to intuitively move and rotate objects in AR without the need for potentially arcane gestures. The one issue which implementing gizmos though, is that they heavily add to the visual clutter on the screen, and the actual handles on the gizmos can be difficult and frustrating to interact with on a small screen.

Rim Lighting

Adding rim lighting to interactable objects could be a subtle hint to users that they can get additional interactions out of an object in AR as well as helping AR objects stand out a bit more from the background. This however comes at the cost of highlighting that an

object in AR is digital, reducing the physicality and potentially hurting the user's sense of immersion.

Map Rotation and Compass

The current Map view is locked to North. In order to create a more free and dynamic view, the controls for the map should be modified to allow free orientation based on user's preference. Furthermore, the compass located on the top-right of the map view should rotate accordingly to the new orientation.

Ghosting

For the purpose of the user's camera never clipping with 3D objects, some form of ghosting could be implemented. This is where objects begin to fade to totally transparent as they would collide with the camera totally avoiding the situations which can result in discomfort in AR.

Color Selection

Currently there is a single texture option available for all objects in AR. In the future, it would be possible to implement a color or texture selection, wherein users can either choose colors to be multiplied over the existing texture map, or wholly new textures. This would heavily increase the amount of customization available to the end user, without requiring new assets to be modeled.

Frames

In addition to the current polaroid frame, the users could be provided with additional options. This would not only add to the amount of expression that the individual users have, but would also introduce an avenue for microtransactions where additional frames could be sold to the users.

User-uploaded models

Implementing some capacity for user-uploaded models would vastly increase the ability for users to customize Mementos to match their artistic vision. This brings with it a host of issues. First, there would need to be a system implemented to either curate which models are available or to report models in Mementos, lest users begin to create and use explicit or copyrighted materials which they have no legal right to. Additionally, this would create a large burden on the servers, which would be forced to host both model and texture files for all user created content, increasing bandwidth and storage requirements, as well as opening the operator up to having to deal with DMCA requests on content. In terms of existing solutions, none of the current platforms seemed to fulfil our needs. Google poly is mostly used to host Tiltbrush sketches, which are far to geometrically complex, as well as somewhat unique in that they use the vertex color to hold all texture data meaning that we would have to recreate Google's functions to convert the vertex color to a texture resembling the original. Likewise, Sketchfab's API is equally flawed, returning models in a format that is natively unusable in Unity, as well as only returning the albedo map, without any metallic/smoothness or normal maps, resulting in models which look bad and do not match the original uploaded to the site.

Editing After Posting

The currently version the app doesn't allow the user to edit the Memento after its creation. For further iterations it is necessary to add new mechanics to edit your own Mementos, as well as deleting. It will also be necessary to define which elements of the Mementos can be altered afterwards.

Comments

Ideally for a social platform users would be able to not only like, but also comment on content that they see, as that builds a sense of community and allows users an additional avenue to interact with content.

Gamification

It is worth exploring the possibility of building mini-games on top of the current available AR interactions. Such gamification would need to be carefully considered,

however, since we would not want them to detract from the overall experience that Memento was built to achieve. Some possibilities could include finding a particular Memento within a given time limit, or perhaps solving a puzzle to “unlock” a Memento.

Target Platforms

Android and iOS

The mobile application was developed for the following target platforms:

- iOS 11.0 or more;
- Android 7.0 (API 24) or more.

Obs: The target platforms are a constraint of the AR plugin used for the camera features.

Development Tools

The app was developed in Unity 3D 2018.3.9f in C#, using Visual Studio 2017 7.8 IDE. The iOS build was created using xCode version 10.2.