Deliverable K:

User Guide

Team 1

Engineering Design - GNG 1103 - Section B03

Isaac Jeaurond - 300062033
Luke Alessio - 300190192
Seyed Ibrahim Hosseini - 300131150
Tesnime Zribi - 8402463
Mathuraa Balasundaram - 300114404

University of Ottawa - Faculty of Engineering

December 10th, 2020

Table of Contents

List of Figures

List Of Tables

n

List of Acronyms

Acronym	Definition
VR	Virtual Reality

Abstract

Due to unforeseen circumstances including the recent COVID-19 pandemic along with the delay of obtaining BIM files that function efficiently with our personal computers, the design team's project progress was significantly delayed. As there were a lack of testing resources and files, the team progressed the most when on campus and with aid from the TA. Despite these aspects hindering the development of the final product, the team managed to complete a majority of the subsystems and components planned within time.

This document works to provide the detailed process the team followed to arrive at the state it is at today. In case either the client or another team would want to complete the project, this document will be of aid to guide them to save them the time and resources. Instructions are provided to define the subsystems the team established and the functionality of each component. Images are placed accordingly to guide how exactly each subsystem functions, what it is capable of, and the procedures. Every subsystem constructed played a critical role in the development of the final solution, therefore, this was discussed in great detail.

Another section of this document articulates the procedure required to utilize this product to its fullest potential such as the installation and health and safety. This description appears tedious but it works to provide the user can refer to in case of queries about the process. Additionally, the maintenance section is included to provide extra information regarding the upkeep of the prototype.

In the last couple sections, recommendations for future work, lessons learnt, internet resources, and project deliverables along with project files were detailed. Deliverables outlined the steps the team took to achieve the final product. The design files include assets and scripts along with AR aspects the team utilized.

1.0 Introduction

This document has been separated into multiple sections in order for another individual to easily access critical details efficiently to implement the same model as we have established. The beginning section details the main issue the team targeted after obtaining information from the client. We further extracted the design needs and requirements related to the product.

Essentially, we constructed the blueprint and the fundamental basis for our final solution for the client. The next section outlines the steps the Unity subteam took to actually implement our solution with subsystems planned from prior deliverables.

There is an additional section that breaks down the steps the user must take to utilize the application to its fullest potential. If the user or the client runs into any issues regarding the operation of our application, this is the chief section they will be required to look up. Along with this component of the document lies the health and safety measures in combination with installation instructions will be provided.

In terms of how to maintain the application along with how to optimize the performance of it will be detailed in the following section. As this is a technological product, there are aspects such as troubleshooting and bug fixes that the client or reproducer should know.

Due to the COVID-19 pandemic and the delay on building files, we were not able to implement all the elements the team aspired to. However, we have outlined the steps required to completely finish the application to reach the best possible solution. This section details the avenues for future work in case that future teams or the client themself would aim to implement these components.

Finally, preliminary information and deliverables that aided us to reach the stage we are at today will be detailed in the final section of this document.

1.1 The Problem Statement

EllisDon is a construction company that builds, maintains, furnishes and equips different buildings around the nation, earning nearly $3.5 billion in annual revenues. As impressive those numbers are, their construction sites still lack digitization which limits their productivity, especially with the increase in labor shortages and building complexities. This could explain why we see an increase in productivity/revenues in other sectors such as product development but not in the construction sector. Another limitation in the productivity in construction for this company would be the use of 2 Dimensional plans which can create misunderstandings and thus a lack of communication when trying to interpret a 2D plan into the real world which is 3D. For example, the architect starts with a 3D mental image which he must convert onto a 2D plan, then the workers must recreate this 2D plan to a 3D building from the blueprints provided to them. As we can see, this process is inefficient thus, eliminating this back and forth movement between 3D and 2D plans, we could avoid confusion among workers and ensure that everyone is on the same page.

This is where an Augmented Reality (AR) app comes into play. An AR app would allow everyone on the construction site to access a virtual 3D plan of the infrastructure through their phone (the app would include building structure, electrical and the mechanical components of the building). By implementing such technology in the construction industry, we could increase productivity among

laborers, increase communication between the headquarters and the site supervisor, and decrease the risk of confusion when interpreting the construction plans.

The following is the final problem statement we determined after deliberation as a team to effectively outline the client's needs:

To increase productivity and efficiency in the construction industry, companies like Ellisdon need to replace 2D blueprints and other outdated systems with more modern means of communication through digitalization by adopting cost-efficient and easy-to-use AR/VR technology.

1.2 Fundamental Needs of the Client and Users

After the introductory client meeting, a designated team member that took down notes during it shared the information they accumulated. Afterwards, utilizing the raw data obtained the team joined on a virtual meeting to extract the information into a problem statement then focused on the particular needs of the client. This process is recognized as the customer needs identification and the particular stage within the design cycle is known as the 'empathize and define' step. After defining this set of needs, the team organized them into a colour-coordinated table. This system enabled the team to focus on particular needs and requirements in order to conserve time efficiently.

The following table are the interpreted needs from the customer statements along with the importance ranked by both colour and number with 5 being of highest priority.

Table 1. Customer Needs Identification

Group	Customer Statement	Interpreted Need	Importance
Increasing Productivity	I want to get rid of the 2D step in the design process and develop everything in 3D	The software allows the user to view 3D models of a building in AR	5
	I want to use technology to tackle increasingly complex projects	The software can handle complex designs	3
Cost	I want to find a cheap (ideally, free) product	The software is extremely affordable	4
User friendly	I don't want to spend time and money on training	The software can be used by anyone regardless of technical expertise	5
Compatibility	I want the software to be compatible with most mobile devices	The software is compatible with most mobile devices	5
	Compatibility with tablets and iPads would be a nice touch	The software is compatible with tablets and iPads	3

	I want to avoid having to use a VR headset or remote control	The software can be used without a VR headset or remote control	4
Visualization	I want to use the app on-sight	The software works on-sight	5
	I just want to be able to view the design that's already been developed; I don't want to develop the design myself or make changes to the design	The software allows the user to view a design that has been developed without changing it	4
	I want to be able to use the mobile device with a 3D model in real life and overlay everything on-site	The software's AR allows the user to overlay the virtual on top of reality	5
	I want to be able to look at the building from outside and walk around it	The software captures the entire building from outside	4
	I want to be able to select a specific room in real life and then use the AR	The software allows the user to select any specific room within the building	4
	I want to see multidiscipline, just the walls and doors; I want to see what's behind the walls and what's behind the ceilings (i.e. mechanical, structural, electrical systems)	The software integrates with multiple disciplines ranging from architectural to mechanical, structural, and electrical	4
	It would be nice to access 2D plans in the software	The software contains 2D documentation	2
	I want a first-person view	The software represents the 3D models in first-person perspective	4
	I want the product to be developed for a single user	The software is developed for a single user	5
	It would be nice if the product had multiple-user functionality too	The software is also suited for multiple users	3

Add/Extract More Information on Building	It would be nice if I could click on an object and learn about its properties (dimensions, material, etc.)	The software allows the user to click on any specific object and learn about its characteristics	2
	It would be nice if I could add annotations	The software allows the user to create annotations within the model	2
Other Assets	I don't want to rely on Wi-Fi to access the software	The software works off-line	4
	It would be nice if I could also view the model when not on-sight	The software gives the user the option to view a model remotely	3

1.3 Design Criteria + Target Specifications

The following table shows each design criteria and their corresponding interpreted need. Each criteria can be described as either a functional or non-functional criteria. Functional criterias correspond to the basic behavior of our system while a non-functional criteria specifies *how* the system should work. (Reliability, usability are examples of non-functional characteristics) The second table details the competing platforms we compared and contrasted utilizing the benchmarking system to differentiate the winner.

Legend: *#5 - Blue; #4 - Green; #3 - Yellow; #2 - Red*

Table 2. Design Criteria (extracted from Deliverable C)

Interpreted Need	Design Criteria	Functional / Non-functional
The software allows the user to view 3D models of a building in AR	Image Resolution (DPI)	Functional
The software can handle complex designs	Lots of Memory (GB) Reliability Robustness Maintainability	Functional

The software is extremely affordable	Cost ($)	Non-functional
The software can be used by anyone regardless of technical expertise	Usability	Non-functional
The software is compatible with most mobile devices	Operating system (IOS and Android) Screen sizes (cm) Portability	Functional
The software is compatible with tablets and iPads	Operating system (IOS and Android) Screen sizes (cm) Portability	Functional
The software can be used without a VR headset or remote control	Phone-Only Functionality	Non-functional
The software works on-sight	Blueprints & Mapping	Non-functional
The software allows the user to view a design that has been developed without changing it	View-Only (with annotations)	Functional
The software's AR allows the user to overlay the virtual on top of reality	Phone-Only Functionality	Functional
The software captures the entire building from outside	OS compatibility	Functional
The software allows the user to select any specific room within the building	Usability	Functional

The software integrates with multiple disciplines ranging from architectural to mechanical, structural, and electrical	Reliability Robustness	Functional
The software contains 2D documentation	Memory Space Usability	Functional
The software represents the 3D models in first-person perspective	OS compatibility	Functional
The software is developed for a single user	Login System Integrity	Functional
The software is also suited for multiple users	Login System Integrity	Functional
The software allows the user to click on any specific object and learn about its characteristics	Engagement	Functional
The software allows the user to create annotations within the model	Engagement	Functional
The software works off-line	Reliability	Functional
The software gives the user the option to view a model remotely	Remote Viewing Mode	Functional

Table 3. Benchmarking of Competing Platforms

Criteria	**Importance**	**Units**	**Unity**	**Tekla**	**Archicad**
OS compatibility	5	Yes / No	Yes	Yes	Yes
Usability	5	1-5	5	5	5
Cost	4	$ CAD	Free	$6 000 annually	Free

Engagement	2	Yes / No	Yes	Yes	Yes
Phone-only functionality (For AR)	5	Yes / No	Yes	No - Only compatible with HoloLens for AR.	No
Reliability	3	1-5	5	3 - Allows the user to view different aspects of the project but since it works with the cloud, you need wifi to operate the latest version of files.	4

1.4 Differentiating Our Product From Others

The main concept that separates our design from other products is that our product is accessible for everyone and can be used by anyone regardless of their technical experience. We created our application with the aim in mind to allow all users to have a smooth and reliable experience with our product. This means that our product is compatible with all devices as well as all operating systems so that our users can launch the application on-site or offsite.

Another important factor that makes our application distinct from other similar products is that our application can overlay virtual designs on top of reality on-site and allows the user to be immersed into an Augmented Reality and walk around their design in real time. Our application can generate the design of the entire structure from inside and outside of the building with toggle effects that the user can use and view different materials in the design. This function is also available for individual rooms in a building.

In addition, our software can be used remotely away from any construction. With this feature, users can create and open designs whenever and wherever they choose, with the additional feature of being able to create annotations on the files. Further, our annotation feature lets users select a part of the file they like and place a comment under it, which can be seen and opened later for reference. Lastly, our application has a functioning multiple users entry in which several users can join the same file and annotate within that file which has proven to be helpful with workers and designers connecting.

1.5 The Product's Main Function

The lack of technology in the construction industry limits its productivity while the buildings are getting more complex. This constraint is due to the misinterpretations that are involved with converting an architect's 3D model into a 2D plan and back to 3D building. Our application is the stepping stone to transition the construction sector into a technologically advanced domain by creating an Augmented reality of the user's plans in their device by uploading files and viewing them in 3D. Similar applications

exist which are extremely costly, but unlike other applications, our application functions cost free and with more features that may come handy to the users in the construction sector. To conclude, our products main function is to enhance efficiency in the construction industry by replacing some 2D blueprints and other outdated communication systems with a modern and digitized method of communication via easy to use and cost efficient AR application.

2.0 Prototype Development

2.1 Toggle Effects

Toggle effects were created by adding a different material to each section of the building. For example, you could create a new material having a blue color or with an added texture from the asset store and add this material to each electrical component. You would do this for every different component of the building. Once the material was added, we defined two different methods for the creation of toggle effects, either with the use of a script (see first and second picture), or by directly assigning a section to a toggle (see third picture). By using a script, you could make the toggle such that on press, the alpha value of the material is decreased to a certain value, which gives you a transparency effect. To do this, you create an empty object, and then add the script to that object. You would then link the appropriate toggle and material to the created empty object with the use of the public variables created in the script. Using the second method is simple but would only allow you to disappear the components, you can't specify any transparency effect. To use this method, you simply need to add an action to the toggle like the picture below shows. You then assign the section you wish to make disappear on the press of that toggle. This being said, when using the transparency code, the application didn't run as smoothly when using the AR compared to just making the objects disappear. Since we wanted a smooth-running game, we used the second method.

```
using System.Collections;
using UnityEngine;
using UnityEngine.UI;

public class Transparency : MonoBehaviour
{
    public Material mat;
    public Toggle tglebutton;
    public Renderer rend;

    public void ToggleOn()
    {
        if(tglebutton.isOn)
          {
            ChangeAlphavalue(mat, 1f);
          }
        else
        {
         ChangeAlphavalue(mat, 0.1f);
        }
    }


    void ChangeAlphavalue(Material mat, float alphaval)
    {
        Color oldColor = mat.color;
        Color newColor = new Color(oldColor.r, oldColor.g, oldColor.b, alphaval);
        mat.SetColor("_Color", newColor);
    }
}
```

*Figure 1: This is the code used for transparency effec*t

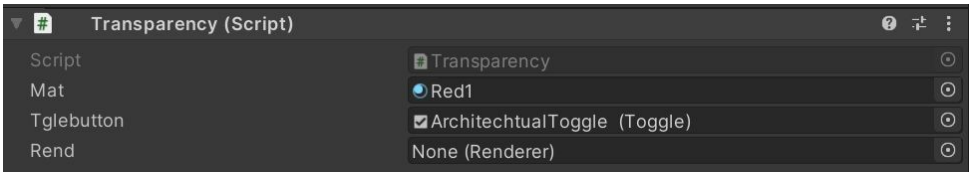


Figure 2: Demonstrates how to link the transparency code to the toggle

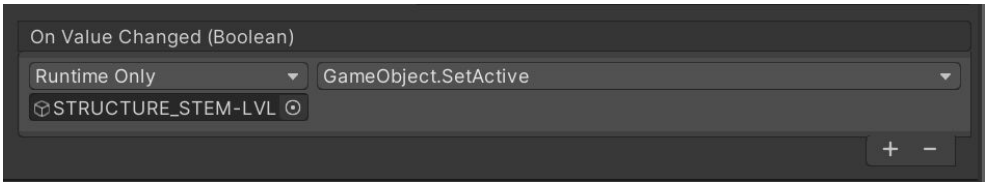


Figure 3: Method 2 example to make object disappear on toggle

2.2 Buttons for Main Menu

For the functionality of the buttons, we simply added a on-click event for each one (see figure below). So you would assign which object you want to "turn off" and another object that you want to "turn on". So when the little box is checked, it means that the attached object will be activated on click, but if the box is unchecked, the object will be deactivated.

Figure 4: example on how to assign objects to menu buttons

2.3 Loading A Scene

To create the functionality of loading a scene from the main menu, we created the script shown below and the loading screen panel shown in the second picture. After creating the script and loading screen panel, you need to create an empty object and attach the code to it. You then need to link the loading screen panel and the slider to their respective placeholders (see third picture). Then, you simply need to link a button from the menu by adding a on-click event to the button. You then slide the empty game object that you created (the ones that contain the script and the attached loading screen and slider) to the on click event, then choose the appropriate script and scene you wish to load (see fourth picture). Note that the scene index is specified when you build the project, so make sure that each scene has the appropriate build index when you are building the game to the phone (see 5th picture).

```
using System.Collections;
using UnityEngine;
using UnityEngine.SceneManagement;
using UnityEngine.UI;

public class LevelLoader : MonoBehaviour
{
   public GameObject loadingScreen;
   public Slider slider;
  public void LoadLevel(int sceneIndex)
   {
      StartCoroutine(LoadAsynchronously(sceneIndex));

   }

   IEnumerator LoadAsynchronously(int sceneIndex)
   {
      AsyncOperation operation = SceneManager.LoadSceneAsync(sceneIndex);
      loadingScreen.SetActive(true);

      while (!operation.isDone)
      {
         float progress = Mathf.Clamp01(operation.progress / .9f);

         slider.value = progress;

         yield return null;
      }
   }
}
```

Figure 5: Demonstrates the code used for loading a scene

Figure 6: Demonstrates the loading scene

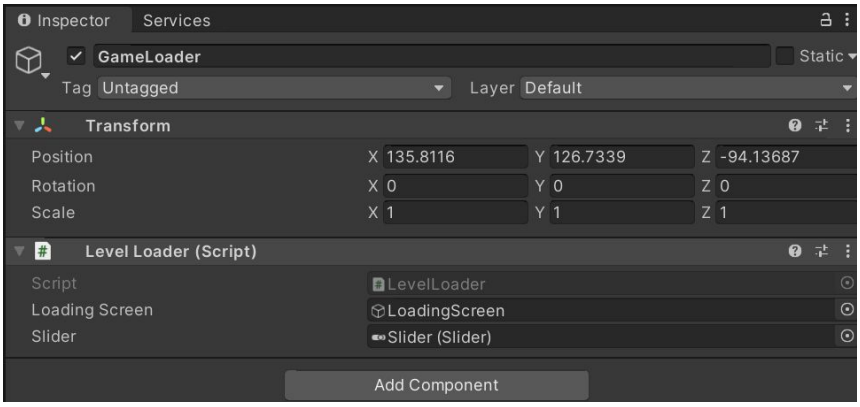


Figure 7: Demonstrates the empty object created, which was linked to the script and the appropriate loading screen/slider.

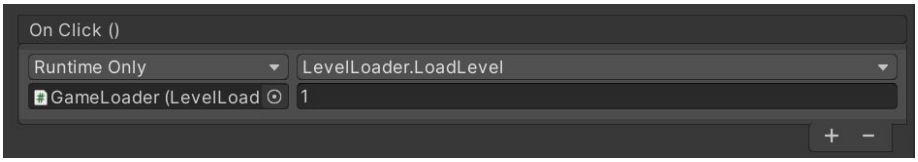


Figure 8: Demonstrates the on-click event for the chosen button.

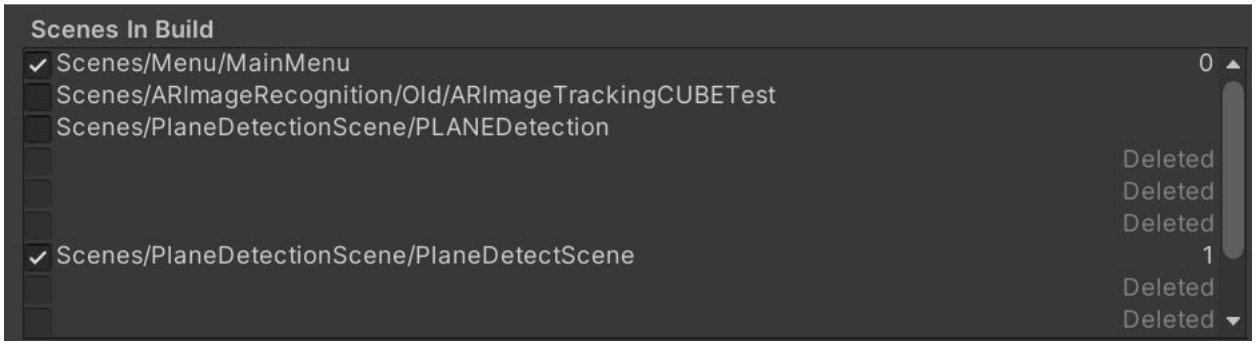


Figure 9: Demonstrates the build panel, note the small numbers to the right, those are the build index. So, make sure that those are the appropriate ones for loading the script.

2.4 Plane Detection

To set up plane detection, we began by installing two important plug-ins: AR Foundation (which came with AR Subsystems) and the ARCore XR Plugin. Then, after creating a new scene with an AR Session and an AR Session Origin, we added a visual placement indicator to help the user understand where an object will be placed in the world, as well as a point cloud dataset. We also created a script to allow the user to tap and spawn the object or double tap and destroy it. We linked this code to an initially empty game object and added our placement indicator to it. Then, instead of using the default rotation that comes with the pose, we created another script to calculate the object's rotation based on the camera direction. This allows the user to control the rotation of the placement indicator simply by turning their phone. Finally, we imported a free plugin called LeanTouch to help with additional functionalities such as allowing the user to pinch the screen to zoom into the object.

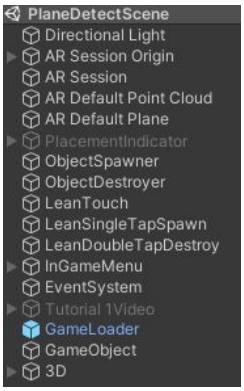


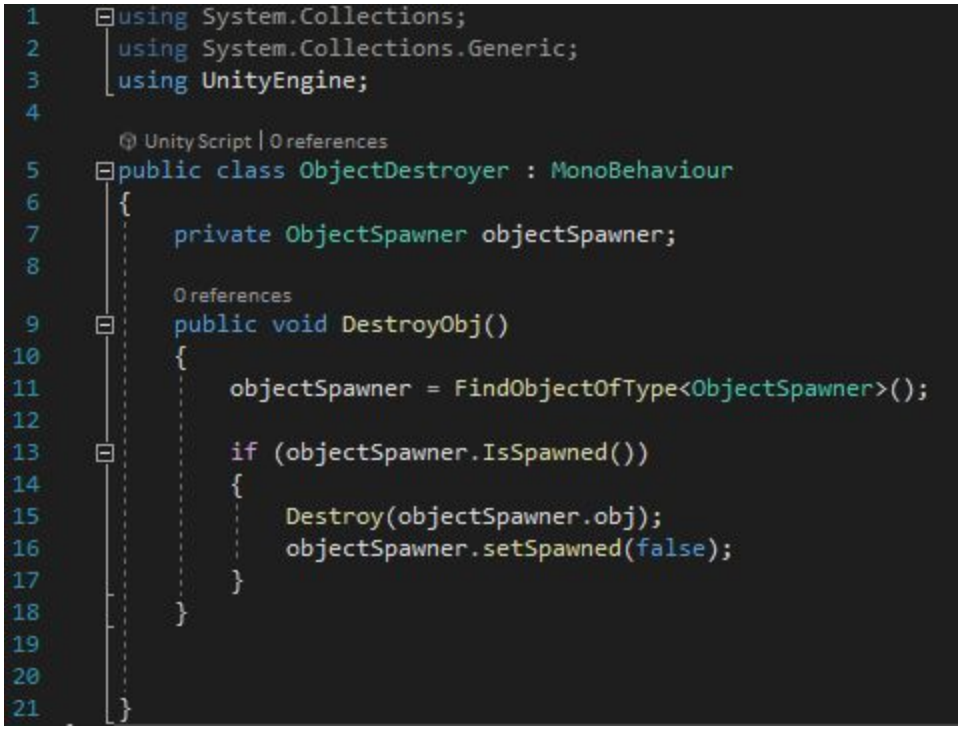
Figure 10: Hierarchy in plane detection scene

```
11      //private float rotateSpeedModifier = 0.1f;
12      private float rotSpeed = 0.5f;
13
14      // Update is called once per frame
        Unity Message | 0 references
15      void Update()
16      {
17          if (Input.touchCount > 0)
18          {
19              touch = Input.GetTouch(0);
20              if (touch.phase == TouchPhase.Moved)
21              {
22                  //rotationY = Quaternion.Euler(
23                  // 0f,
24                  //   -touch.deltaPosition.x * rotateSpeedModifier,
25                  //   0f);
26
27                  // rotationX = Quaternion.Euler(
28                  //   0f,
29                  //  -touch.deltaPosition.y * rotateSpeedModifier,
30                  //  0f);
31
32                  // transform.rotation = rotationY * transform.rotation;
33                  //  transform.rotation = rotationX * transform.rotation;
34                  float rotX = Input.GetAxis("Mouse X") * rotSpeed;
35                  float rotY = Input.GetAxis("Mouse Y") * rotSpeed;
36
37                  Camera camera = Camera.main;
38                  Vector3 right = Vector3.Cross(camera.transform.up, transform.position - camera.transform.position);
39                  Vector3 up = Vector3.Cross(transform.position - camera.transform.position, right);
40
41                  transform.rotation = Quaternion.AngleAxis(-rotX, up) * transform.rotation;
42                  transform.rotation = Quaternion.AngleAxis(rotY, right) * transform.rotation;
```

Figure 11: Script used to rotate in 3D

```
8       private PlacementIndicator placementIndicator;
9       private bool objSpawned = false;
10      public GameObject obj;
11
12
        0 references
13      public void Spawn()
14      {
15          //Get the placement indicator component
16          placementIndicator = FindObjectOfType<PlacementIndicator>();
17
18          if (!objSpawned)
19          {
20              //instantiate our object to spawn on top of placement
21              obj = Instantiate(objectToSpawn, placementIndicator.transform.position, placementIndicator.transform.rotation);
22              objSpawned = true;
23
24          }
25
26      }
27
        1 reference
28      public bool IsSpawned()
29      {
30          return objSpawned;
31      }
32
        1 reference
33      public void setSpawned(bool val)
34      {
35          objSpawned = val;
36      }
37  } //class
```

Figure 12: Script used to spawn object



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

Unity Script | 0 references
public class ObjectDestroyer : MonoBehaviour
{
    private ObjectSpawner objectSpawner;

    0 references
    public void DestroyObj()
    {
        objectSpawner = FindObjectOfType<ObjectSpawner>();

        if (objectSpawner.IsSpawned())
        {
            Destroy(objectSpawner.obj);
            objectSpawner.setSpawned(false);
        }
    }


}
```

Figure 13: Script used to destroy object

3.0 Instructions to Use the Prototype

This product is created to allow any user to interact with 3D designs of their plans using the AR system we have created. With the help of hand controlled electronic devices such as phones, iPads, Tablets, and laptops our users can effectively visualize and annotate their design. In order to simplify the users experience, the following series of bullet points will describe how any user can use our product with all the available features.

3.1 Installing the Application

In or to install this application, the user must open the project file from Unity. Once the application has been opened in unity, users can upload files that they would like to view and compile the project.

3.2 Homepage & Main Menu

One the application has been successfully installed and opened, users will be welcomed in the homepage where they will select their preferred language. Our product functions in both French and English for users all across Canada. If this is the first time a user is opening the application, they will be directed to the Video tutorials we have created.

These tutorials have an option where the user can skip them and move to the Main Menu. In the main menu, the following are 5 options available for users to choose from:

Internal View - This option will allow the user to view the internal design of their project. Firstly, each user will have to place themselves in the marked spots that are used to coordinate themselves inside the AR design.

External View - This feature is used to allow users to see outside of their design. Mainly this feature will be used to see how parts assemble together in a project that is not yet set up. This feature also required the users to start at specific markers to assure that the design viewed correctly.

Tutorials - These are a series of small video tutorials displaying how the application works. The videos cover all important aspects of the application such as how to set up the AR design; how to place comments and annotations in the design and more. Each video is created with the option to skip incase users are already familiar with those concepts.

Change language - This feature allows users to select between English and French. The default language of this product is English but that can be changed to French based on the user preference.

Exit - This option closes all open files for the user and closes the application.

4.0 Prototype Maintenance

Despite the issues that our team faced such as delay in delivery of our design, lack of access to the STEM building due to COVID-19, limited time, our application is functioning well and in order to maintain the product's life span the design needs to be debugged, tested, and updated by a team. This design was created using the Unity software, therefore updates are necessary to assure our product is up to date with the latest model available for the ease and comfort of our users. Since this project was created by a team of students, further development can also be done at a professional level.

One main aspect of this project is that our users can connect via our application and annotate or view their designs. It can be predicted that once a certain number of users are connected via application at a time, our software might develop bugs that are not yet visible. Since our team will not be able to conduct further testing on this project at this time, further debugging process and development is crucial to assure that our product will be reliable for users.

Lastly, similar to any other application, frequent testing of different aspects of the application will reassure our users and the development team that our product is the best it can be. As of now, our application works very well and it has passed the testings that we have done, but it is without doubt that if our team had more time and better access to resources, we would be able to enhance our product to be at peak performance.

5.0 Conclusions & Recommendations for Future Work

Due to the various anomalies and circumstances that the design team faced throughout the length of the project, not all of the features that were originally intended to be included in the final product were able to be implemented by the end of the semester. Because of this, the design team has included a list of useful features and improvements that were originally intended to be included, but were not able to be in time.

While the design team was able to include an annotation system in the final prototype of the application, the annotation system was not yet linked to the cloud, and as such, different users viewing the same object would not be able to see the annotations made by their coworkers. Adding a cloud-based system for annotations is greatly recommended.

Another useful feature that was not able to be included in the final prototype is an employee log-in tool, where employees working alongside the client are able to enter a username or employee number alongside a password, in order to access a specific building view. This is a very useful feature for effective communication and managing of employees in a workspace, and as such is greatly recommended for future work.

Furthermore, a few of the glitches regarding the gaze interaction features and layer visuals systems were not able to be entirely fixed before the ending of the semester. It is recommended that these features be improved upon or fixed, so that they are more effective and informative for users of the application.

A final recommendation is for at least one member of a future design team to become proficient with the c-sharp programming language, as this is often very complicated, and necessary to use Unity efficiently.

6.0 Bibliography

EllisDon, (July 2016). *EllisDon Ranked #2 by On-Site Magazine's list of Top 40 Canadian Contractors*. URL:

https://www.ellisdon.com/news/ellisdon-ranked-2-by-on-site-magazines-list-of-top-40-canadian-subcontractors/

7.0 Project Deliverables

In the design team's folder on MakerRepo entitled "GNG1103B6 AR Project by Futurlistic", all project deliverable files are stored and available for download. The intention for these submission files is for the clients, professors, and/or teaching assistants to review the progression of the design team's progress throughout the months that the project was being worked on. A link below has been provided for easy access to the design team's MakerRepo page, which also includes screenshots of the application development.

https://makerepo.com/IsaacJeaurond/gng1103b6-ar-project-by-futurlistic

A table has also been included below, which provides a brief summary and description of each project deliverable that has been included in the MakerRepo page. If further details are desired, use the link provided above to download and view any of the desired project deliverable files.

Table 4: List and Description of Project Deliverables

Deliverable Name	Description of Deliverable
Needs Identification & Problem Statement (B)	- Data involving client's needs / wants for the application - Ranking of interpreted needs by importance - Problem statement
Design Criteria & Target Specifications (C)	- Identified needs from deliverable B, and compiled them into a design criteria - Comparing potential programs - Benchmarking potential programs
Conceptual Design (D)	- Brainstorming from each design team member - Subsystem concepts - Global concepts - Benchmarking global concepts & finalizing concepts
Project Schedule and Cost (E)	- Estimate for the cost of the AR project - Estimated schedule for the remainder of the project
Prototype I & Customer Feedback (F)	- Summary of current project progress - Analytical and physical report of the current features - Summary of client feedback
Prototype II & Customer Feedback (G)	- Summary of current project progress - Analytical and physical report of the current features - Summary of client feedback
Prototype III & Customer Feedback (H)	- Summary of current project progress - Analytical and physical report of the current features - Summary of client feedback

8.0 Final Product

Although all of the relevant deliverables and design content have been uploaded to the design team's MarkerRepo page, due to the large size of the design files and application in terms of storage space, the final product was unable to be uploaded to MakerRepo at this time. Because of this, a client wishing to access the final product of the application can contact any member of the design team via email, and have the necessary files sent to them that way. The design team's email addresses associated with the University of Ottawa can be found below.

Isaac Jeaurond - ijeau038@uottawa.ca

Tesnime Zribi - tzrib068@uottawa.ca

Luke Alessio - lales097@uottawa.ca

Seyed Ibrahim Hosseini - shoss063@uottawa.ca

Mathuraa Balasundaram - mbala042@uottawa.ca