**BILKENT UNIVERSITY**

**ENGINEERING FACULTY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**CS 399**

# SUMMER TRAINING REPORT

## Umut Mücahit Köksaldı

**21402234**

**Performed at**

## HAVELSAN

**31.07.2017 – 25.08.2017**

**Table of Contents**

# 1 Introduction

I have completed my first summer practice between July 31 and August 25 at HAVELSAN Inc., located in Çankaya, Ankara. The company is mainly focused in the defense and IT sectors. HAVELSAN is mostly active in the fields of C4ISR, naval combat systems, E-government applications, reconnaissance surveillance and intelligence systems, management information systems, simulation and training systems, logistic support, homeland security systems and energy management systems. While working in HAVELSAN, I was exposed to a professional work environment and I was able to observe how the business processes were handled from a software perspective. I also had the opportunity to obtain knowledge on some new fields that were previously unfamiliar to me, as well as incorporating the skills I had learned throughout my studies into the work that I have done during my internship.

From the first day of my internship I was assigned to the sub-division of HAVELSAN located in METU Technopolis. In this department, HAVELSAN was working on realizing an incubator open to engineers who wished to gain access to expensive and sophisticated tools that they would otherwise would not be able to acquire on their own. HAVELSAN encourages these engineers to come up with their own innovative ideas and implement them in their incubator in order to boost innovation and development within HAVELSAN and Turkey. I was one of the initial engineers deployed to work until the public unveiling of the new incubator which would be named HAVELSAN Open Innovation Center. I was given the task of developing an augmented reality application which would serve as an interactive demo for the innovation center itself. As augmented reality was a new field for me to explore, I was able to gain lots of insight regarding this new and aspiring territory of software engineering.

Throughout the rest of the report, I will be providing information about the department and the workplace I have carried out my internship, as well as giving detailed information on the work I have done, and the circumstances of the workplace and the co-workers that were assigned to me during my internship. Lastly, I will present the outcomes of my work and the skills I have both utilized and gained during my employment.

# 2 Company Information

## 2.1 About the company

HAVELSAN was established by the Turkish Air Force Foundation in 1982 as a Turkish company named HAVELSAN-AYDIN, in order to provide maintenance for the Turkish Air Force's high technology radars. In 1985, HAVELSAN was separated from the foreign shareholders and incorporated as a national company with a share of 98% owned by the Turkish Armed Forces Foundation. With over 1300 employees working within the company, HAVELSAN has participated in innovative and sustainable developments in both national and international scale [1]. Focusing heavily on software, its main four departments are as follows:

- Command Control and Battle Systems Development
- Education and Simulation Technologies Department
- Management Information Systems Department
- Cyber Security Solutions Department

HAVELSAN dedicates itself do developing innovative and powerful solutions regarding each of these areas, and making the final product available both domestically, as well as importing it to its international customers [2].

## 2.2 About your department

The department I was assigned to at HAVELSAN was the Education and Simulation Technologies Department. In this department, I was positioned at their sub-office located in METU Technopolis, and I was given the task and the tools needed to develop an augmented reality application. The department was in the process of launching an incubator office named the Open Innovation Center that would serve as a means of providing resources to engineers who had innovative ideas but lacked the capability to develop them due to their inability to acquire tools and technologies [3].

These engineers could be both form within the company and outside. I was one of the first engineers positioned at this office and the application I was developing would serve as both the introductory demo application of the office and also as a means of showing the capabilities of the tools that HAVELSAN would provide to the engineers who applied to work there. As the name of the new office suggests, the main goal of the HAVELSAN Open Innovation Center was to provide an open space for engineers from all around the country to participate and develop their innovative ideas into sustainable products.

### 2.3 About the hardware and software systems

The department I was assigned to within HAVELSAN was the Open Innovation Center and this department was aiming to branch out as much as possible regarding its hardware and software systems, in order to accommodate a high number of outside and inside engineers who wished to become a part of the newly opened branch of the company. In the Open Innovation Center, there was a substantial collection of development kits to be used in the area of Internet of Things (IoT). In addition, there were multiple virtual reality (VR) headsets, namely the Samsung Gear VR headset and the Google Cardboard headset, available to the engineers for development. The office included a 3D Printer mainly for the use of the IoT development team in order to print out small components for the development stage. Lastly, I was given a Microsoft HoloLens augmented reality headset to work on and develop the demo application which would be used in the presentation of the Open Innovation Center. I would develop for this platform using the laptop I was given and the Unity 3D engine installed within. I was also provided a 3D modeling tool called Blender in order to quickly prototype some preliminary models to be used as holograms in the demo application. During my internship, I got to work with lots of new and interesting tools and

development environments, which have provided me with invaluable software engineering experience.

### 2.4 About your supervisor

My supervisor during my internship was the Technology & Innovation Manager at HAVELSAN. His name was Tolga EROL and he was a graduate of Mechanical Engineering from METU in 2001. In his career, he has gathered lots of experience working as a project manager in both HAVELSAN and other companies, managing and coordinating the teams undertaking different software engineering projects. He can be reached from the phone number *+90 312 219 57 87*, and also from the e-mail address: *terol@havelsan.com.tr*. His address is: *Mustafa Kemal Mahallesi 2120 Cad. No:39 06510 Çankaya Ankara TÜRKİYE.*

# 3 Work Done

During my internship, I was tasked with the development of an augmented reality application to be deployed on the Microsoft HoloLens I was provided. The application would be used as an introductory demo to the new Open Innovation Center HAVELSAN was preparing to launch. To this end, I was asked to conduct an analysis and research in order to find out how this application could be realized, and then implement the application. During my summer practice, I made numerous presentations to the VPs of HAVELSAN, my supervisors, as well as outside visitors such as the computer science professors from various universities in Ankara. My team consisted of one other computer engineer and two electrical engineers, four people in total. I will divide this section into sub-sections in order to accurately and thoroughly describe the work that I have performed, and the challenges I have faced. I will provide all of the code that I have written along with their explanations on the appendix section of this report.

## 3.1 Setting up the HoloLens

To start with, I was given the task of researching how to write an augmented reality application. During my research, I have found that the task could be accomplished using game engines, which supported AR development through different APIs provided by Microsoft. Two prominent game engines that I have tested for this task were the Unity 3D engine and the Unreal Engine 4. While both engines provided adequate amount of documentation and support for augmented reality development, Unity 3D was far easier to set-up and use compared to Unreal Engine 4. Subsequently, I have downloaded the toolkit Microsoft provides for Unity 3D called HoloToolkit, and began developing the augmented reality application on this engine and I used the Visual Studio 2017 Community edition as the IDE which was provided as the default IDE.

### 3.2 Spatial mapping of the office

The next task I started working on was spatial mapping the office in order to transfer it as a 3D model to the game engine. We needed to do this because otherwise we would not be able to place the holograms into the desired places accurately. The spatial processing was an easy task as there were already means of accomplishing this job embedded inside the HoloLens. We have connected to the HoloLens on our local network from a computer that was operating on the same network, and we have mapped our office as a 3D mesh that was ready to be imported into the game engine.

### 3.3 Placing the holograms

We aimed to develop our application in a way that would make our office look as an interactive environment to the person who would be wearing the HoloLens. To this end, we have downloaded aesthetically pleasing assets with animations from the Unity Asset Store, and then imported and incorporated these assets into our application. We have placed the holograms inside the scene by taking into account where the user would start up the application, and the relative positioning of the walls and obstacles inside the office. After this point, the team started to focus on different aspects of the application such as calibrating the sensors on the HoloLens, setting up the live streaming service and finding assets and animations to be imported into the game. I was working closely with the other team members as I was tasked with developing the functionality and providing stability to our application.

### 3.4 Hologram occlusion

So far, the application was working as intended with all holograms showing up at the desired locations. However, the office consisted of 3 different rooms: the main working area, the meeting room, and the server room. I was asked to put interactive holograms on each of these rooms, and at this stage of development, all holograms

would be visible all the time, even if they were behind a wall in another room. I was asked to come up with a way to prevent this, to make sure that the user would only see the holograms inside the current room they were in. I have assigned specially designed materials to the holograms that would ensure that they were hidden (occluded) if they were behind another hologram or some obstacle. The solution worked most of the time, however, as the spatial mapping of the office was not extremely accurate due to sensor inaccuracy, there were some discrepancies that would occur at certain locations in the office. In order to remedy this problem, using Unity 3D I have placed invisible cubes around the office which would act as triggers to determine the location of the user, and then enable or disable the different holograms spread around the office space. Using C#, I have written scripts that would define the actions of these trigger cubes, and as a result, I have resolved the occlusion problem, ensuring that the user would not see the holograms that they were not meant to see.

### 3.5 Gaze and gesture recognition

After I made sure the visual part of the application was working correctly, I then moved on to implement interactivity into the application. In order to let the user interact with the holograms placed around the office, we had to recognize two entities: where the user was looking, and the user's selection gesture. These two features were classified in HoloLens as the gaze and air tap, respectively. In order to integrate these two features, I wrote a script that would govern the interactivity in the game, called the GazeGestureManager. In the script, I cast a ray from the camera, which is the HoloLens mounted on top of the user's head, out into the world and wherever the ray hits, I placed a small circle which would act as a cursor to let the user know about where their gaze would exactly point to. If the user would do the air tap gesture with their hands while looking at an intractable object, that object would be notified with and its OnSelect

method would be triggered. I achieved this modularity by using the observer design pattern, which allowed me to separately define the behavior of the various objects we placed around the office space.

### 3.6 Hologram persistence and world anchors

I finished implementing interactivity into the application and subsequently there arose another problem: hologram persistence. At this stage, the application would have to be started from the exact center of the office each time it was launched; otherwise the holograms would shift and they would end up in undesirable places. This was because of a shift in the origin position of the world which depended on the holographic camera. However, I did find a solution to this problem through research. In order to solve this problem, I attached components called World Anchors to each of the objects that we wanted to be fixed in place, and then wrote a script that would serialize and store the positions of these world anchors into a database in order to preserve them even after the application is terminated. The script would also load these locations upon subsequent launches and place the holograms into the appropriate locations. As a last touch, I defined each hologram in the application space as moveable, which meant that users could pick them up and place them wherever they wanted to, and the world anchors would move with them as well.

### 3.7 Extras

Once I have finished working on the main part of the application, which was aimed to make a demonstration of the office space we worked in, I was assigned to develop a sub-environment that could be accessible within the main application scene. This scene would act as a simulation environment to illustrate explosions acting on an abstract cubic object. Developing this environment in the context of augmented reality rather than it being limited to the 3D space inside the game engine was extremely

desirable because it gave the user the freedom to easily walk around and view the destructible object from various angles. The environment features an arbitrarily and randomly shredded cube, and the user is presented with a control panel to set the radius and the force of the explosion to whatever values they desire. After the values are set, the user then needs to look at the cube and air tap, and the explosion will commence exactly at the position on the cube they were looking at. In order to recognize the gaze and the gesture, the aforementioned manager was reused. I developed the manager as a singleton so that it would not have to be re-instantiated as the application was switching the scene context. The simulation of the explosion was achieved by collecting all of the cube pieces that were in the radius, and moving them outward from the center of the explosion. Combined with gravity and a high explosion force, the effect reasonably simulated an explosion in our virtual environment.

To conclude, during my internship I got to work with new technologies and tools to build a full-fledged application from scratch. During this process, I got to learn about the challenges of working within the context of augmented reality, and I was able to apply both my previous engineering knowledge and the knowledge I obtained during the research I conducted in my internship to come up with effective solutions to these problems.

# 4 Performance and Outcomes

I was able to accomplish all of the tasks given to me in the required amount of time. At the end of my internship I had fully implemented the augmented reality application that would be used as a demo application of the newly established innovation center. My supervisor was extremely satisfied with my performance as I was able to deliver the features he asked from me and went the extra mile on several occasions to enhance the user experience of our application. Altogether, working in tandem with my team, we were able to complete the project that was assigned to us to the full satisfaction of both our department and the supervisor.

## 4.1 Applying Knowledge and Skills Learned at Bilkent

During my internship, I mainly programmed in the C# language, as this was the scripting language that was used by the Unity 3D engine. Initially I had to learn the language, which was relatively easy as it was quite similar to Java in many aspects except some small syntactic changes. Since Java was the first language I started programming in, I quickly learned how to work with C#, using the programming knowledge I obtained in CS101 and CS102 courses.

Moreover, one of the most difficult challenges I have faced was storing the locations of the holograms in order to load them up and preserve their positions upon successive application launches. In order to solve this problem, I serialized and saved the locations of the holographic game objects, and stored them inside a database created at the storage unit of the Microsoft HoloLens. The development of this solution, from design to implementation was aided by the knowledge I obtained during my studies in Bilkent. Both the research I have conducted in order to tackle the problem and the database management portion of the solution benefited from the fact that I had taken CS353 in the previous semester.

Furthermore, I have made sure to follow the coding conventions bestowed upon me during my first and second year in Bilkent. The code quality and readability was highly praised by both my teammates and my supervisor, and has enabled them to read, change and work with my code that much easier. My extensive knowledge of algorithms and data structures, which I have acquired during the courses CS201 and CS202 has helped me approach the coding problems in a productive and analytical manner and has made the code I produce a lot more efficient. Incorporation of simple and easy to understand algorithms into my code has also made my teammates learn from me and follow suit while writing their sections of the project as well.

Lastly, along these data structure and algorithm knowledge, I have integrated the design patterns that I got to learn in CS319. In order to achieve modularity in the interactivity part of the application, I have used the observer design pattern to its full extent. In addition, in order to achieve persistence of the top-level management classes that I wrote for the application, I used the singleton design patterns to prevent multiple copies of the object from being instantiated.

### 4.2 Solving Engineering Problems

To start with, one of the problems I faced was making the holograms persist across multiple application starts. Unity would pick the world origin as whichever point you launched the application inside the office, and as a result, all of the holograms would be placed according to this origin point. If the application was not started at exactly the designated place, the holograms would appear rotated and out of place, resulting from this undesired origin shift. In order to come up with a solution to this problem, I developed a system so that the world coordinates of the holographic game objects would be serialized and stored in a database the first time the application was launched. After each successive launch, these locations would be loaded up from the

database and even if the application was started in a different location inside the office, the holograms would appear in their appropriate locations. This solution made the arbitrary starting of the application possible, which was an extremely desired feature.

Furthermore, while simulating the explosion, I had to deal with some performance issues regarding rendering. To begin with, the simulation would be accomplished by using a cube mesh which was shredded using the Blender 3D modeling tool. However, if this shredded cube was to be loaded in the scene completely, it would have to render all of the pieces, even if the pieces were not being interacted with, which caused the relatively weak processor of the HoloLens to struggle. In order to increase performance, I went with just loading the unshredded cube at launch, which is simply a mesh consisting of 6 polygons, as opposed to the thousands of pieces that would have to be rendered with the shredded cube. If the user chose to interact with the cube by simulating an explosion, I would then load the shredded cube into the scene and remove the solid one, and only then begin the physics calculations for the explosion. This solution increased the performance of the application immensely, and it was lauded by both my teammates and my supervisor for being an intuitive solution to a substantial problem.

### 4.3 Team Work

I worked as part of a 4-person team which consisted of one other computer engineer and two electrical engineers. Initially, we worked together on setting up the HoloLens and mapping the office space via spatial processing. After we set up our development environment, we divided into 3 groups which worked on separate aspects of the application and the HoloLens. One group, which consisted of the electrical engineers worked on setting up the live streaming service, so that we could display on TV whatever the user was seeing. They used a wireless receiver which would be

connected to the television, and the HoloLens would communicate with it to send its visual and holographic data. The other computer engineering student worked on acquiring and setting up 3D assets that would be used as holograms in the application. I worked on programming the interactivity layer for the application as well as stabilizing the application to make sure the final product was a complete and sustainable piece of software. I worked closely with all groups; as we needed to seamlessly integrate the holographic assets into the world space in order to provide an intuitive and unique user experience. In addition, I had to communicate with the live-streaming team at all stages of development, as we needed to use the resources of the HoloLens conservatively. Since live-streaming would use up considerable amount of processing power from the HoloLens, I needed to make sure that the features I implemented were not too intense on the hardware of the HoloLens. With each group working together, we were able to accomplish our tasks and deliver the final product to the full satisfaction of our supervisor.

### 4.4 Multi-Disciplinary Work

During my internship, I got to work with two electrical engineers who were trying to set up a live streaming service from the HoloLens to the TV screen we had in the office. They were working with a wireless receiver that would be connected to the TV and would also communicate with the HoloLens in order to broadcast the camera data with the holograms incorporated. They were trying to set up the communication pipeline between the HoloLens and the wireless receiver, and I worked closely with them; providing assistance where I could, and learning about the constraints that the live-streaming service would put on the hardware, so that I could adjust my code accordingly. Working with the streaming service and the wireless receiver, albeit

briefly, provided me with lots of insight regarding a new field which was previously foreign to me.

### 4.5 Professional and Ethical Issues

As a team, we did not encounter any ethical issues among us or with our supervisor. As for professional issues, we had some problems with some members of the team failing to meet the deadlines. Some parts of the project were delayed as a result of this. Even though there were some problems that had arisen, we learned that it was important to communicate properly and work through the problem by identifying the causes and then moving on to the solution. My teammates and I learned to work together as time passed on and we became a more compatible team and we have managed to reduce the problems and work as a more efficient team. In addition, as a result of security concerns, we were not able to work on our own laptops, which hindered the process of development slightly, because the machines that were provided to us were not ideal in terms of hardware and software. With that being said, this problem was swiftly solved and we continued to efficiently work on developing the application.

### 4.6 Impact of Engineering Solutions

HAVELSAN is one of the leading engineering focused companies in Turkey. Their main work areas include C4ISR, naval combat systems, E-government applications, reconnaissance surveillance and intelligence systems, management information systems, simulation and training systems, logistic support, homeland security systems and energy management systems. Working in these fields, HAVELSAN produces sustainable, integrated and efficient solutions to many contemporary problems. Their biggest strength is the development of simulation technologies, especially in the context of military systems. They have developed

accurate simulations to various battle systems and devices such as fighter jets and armored vehicles, which are lauded both within Turkey and internationally. As a result of their efficiency in developing these solutions, they have exported many of their products to other countries such as Pakistan, Azerbaijan and Germany [4].

### 4.7 Locating Sources and Self-Learning

During my summer practice, I was tasked with the development of an augmented reality application, a field in which I had no experience. In order to accomplish the task that was given to me, I had to research a lot and learn about the different tools I would need to develop this application. I was given a head start because I was told to use either the Unreal Engine or the Unity 3D engine to develop the application. Since I knew that I needed to use a game engine, I needed to find out which one would be best suited for this task. I had to research the toolkit provided by Microsoft specifically for HoloLens and augmented reality development and found that Unity 3D would be the best tool to use because it was the most supported in terms of documentation and libraries, and then went to work on learning how to use these tools. I had to locate the documentation for both the core engine and the HoloLens toolkit, and studied these extensively throughout the development stage. Combining my newly obtained knowledge with my engineering skills, I was able to tackle all of the problems and develop the application.

### 4.8 Knowledge about Contemporary Issues

As part of my internship, I got to learn about areas of computer science such as computer graphics, rendering, and how these relate to augmented reality; the subject I was working on. I found out the challenges of rendering an object through the multiple lenses of an augmented reality headset, and how the engineers have overcome this problem. In order to make the hologram as realistic as possible, they would render two different angles of the same object, and then combine the results in the headset, then

display it appropriately to the user through the headset's lenses. The approach is similar to how we biologically and naturally see the objects around us, which makes the solution to this substantial problem extremely intuitive and easy to understand. While I was studying these concepts, I also got to learn about the different tradeoffs that engineers would face with regards to rendering objects in 3D space, and the efficient solutions developed to overcome some of the challenges such as normal maps, lighting techniques and algorithms and so forth. All in all, my internship was an extraordinary opportunity for me to get involved in some of the most up-and-coming fields of computer science.

### 4.9 Using New Tools and Technologies

Throughout my internship, I worked with prevalent and new technologies. The most paramount of which was the Microsoft HoloLens that I was given to develop the application on. This device was not only technologically new and advanced, but it was also still in its development stage with new features added to it over time. In order to develop on this device, I got to learn about some of the most advanced game engines such as Unity and Unreal Engine. These engines were still improving as there were new releases of Unity happening almost every month. In addition, the tools and libraries I was working with were also extremely new and fresh, and the concepts that I got to learn such as hologram anchoring, gaze and gesture recognition, were also fresh subjects of computer engineering. I often had to come up with my own solutions to the problems I faced because there simply was not enough documentation on the subject matter, as these fields were particularly novel. However, through the hardships I faced, I obtained invaluable knowledge about new fields of technology and engineering, for which I am grateful.

# 5 Conclusions

I have spent nearly a month, doing my internship at HAVELSAN. During the time I have spent inside the company, I was able to accomplish a lot and also learn a lot. I was tasked with the implementation of an augmented reality application which would be used as a demo application to introduce the Open Innovation Center that my department was planning to launch. I developed the interactivity layer of the application after setting up the development environment. I was also responsible for stabilizing the holograms and recognizing speech, gaze, and gesture. I conducted the necessary research in order to fulfill these tasks and made presentations to both my supervisor and other managers in our department documenting my progress. My team valued and appreciated my contribution to the system and they expressed their satisfaction with the efficiency and readability of my code, as well as the effectiveness and creativity of the solutions I brought to the problems we faced. In the end, my internship has both provided me with invaluable knowledge and experience about an up-and-coming field of computer science, as well as letting me hone my skills in a real work environment.

## References

[1] HAVELSAN, ‘Şirket Profili’, 2015. [Online].

Available: http://www.havelsan.com.tr/TR/Main/icerik/3/sirket-profili

[Accessed: 22-Sep-2017]

[2] HAVELSAN, ‘Organizasyon Şeması’, 2015. [Online].

Available: http://www.havelsan.com.tr/TR/Main/icerik/213/organizasyon-semasi

[Accessed: 25-Sep-2017]

[3] HAVELSAN, ‘Havelsan İnovasyon Merkezi’, 2016. [Online].

Available: https://inovasyon.havelsan.com.tr/hvlsn/ [Accessed: 27-Sep-2017]

[4] HAVELSAN, ‘Havelsan’ın Son Dönem Başarıları’, 2017. [Online].

Available: http://www.havelsan.com.tr/TR/Main/haber/1265/havelsanin-son-donem-basarilari-trt-haberde [Accessed: 28-Sep-2017]

## Appendices

```
using System.Collections.Generic;
using System.Linq;
using UnityEngine;
using UnityEngine.Windows.Speech;

/*
Top level class used to recognize voice commands from the user. Mostly used
for debugging purposes
 */

public class SpeechManager : MonoBehaviour
{
    KeywordRecognizer keywordRecognizer = null;
    Dictionary<string, System.Action> keywords = new Dictionary<string,
System.Action>();
    [SerializeField] private GameObject resetListener;

    // Use this for initialization
    void Start()
    {
        keywords.Add("Reset", () =>
        {
            // Call the OnReset method on every descendant object.
            // this.BroadcastMessage("OnReset");
            resetListener.SendMessageUpwards("OnReset");
        });

        keywordRecognizer = new KeywordRecognizer(keywords.Keys.ToArray());

        // Register a callback for the KeywordRecognizer and start
recognizing!
        keywordRecognizer.OnPhraseRecognized +=
KeywordRecognizer_OnPhraseRecognized;
        keywordRecognizer.Start();
    }

    private void
KeywordRecognizer_OnPhraseRecognized(PhraseRecognizedEventArgs args)
    {
        System.Action keywordAction;
        if (keywords.TryGetValue(args.text, out keywordAction))
        {
            keywordAction.Invoke();
        }
    }
}
```

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

/*
Behavior class that defines random movement over a predesignated area known
as navmesh
Used primarily for the drone in the scene
 */

public class NavMeshRandomMovement : MonoBehaviour
{
    [SerializeField] private NavMeshAgent agent;
    [SerializeField] private float walkRadius;
    [SerializeField] private float agentSpeed = 2f;
    private float timer = 3f;
    private float currentTime = 0f;
    [SerializeField] private Vector3 currentDestination;


    // Use this for initialization
    void Start ()
    {
        agent.speed = agentSpeed;
        Vector3 randomLocation = GetRandomLocation();
        currentDestination = randomLocation;
        agent.SetDestination(currentDestination);
    }

    // Update is called once per frame
    void Update ()
    {
        currentTime += Time.deltaTime;

        if (currentTime >= timer)
        {

            agent.speed = agentSpeed;
            Vector3 randomLocation = GetRandomLocation();
            currentDestination = randomLocation;
            currentTime = 0f;
            agent.SetDestination(currentDestination);

        }
    }

    private Vector3 GetRandomLocation()
    {
        NavMeshTriangulation navMeshData = NavMesh.CalculateTriangulation();
```

```
        // Pick the first indice of a random triangle in the nav mesh
        int t = Random.Range(0, navMeshData.indices.Length - 3);

        // Select a random point on it
        Vector3 point =
Vector3.Lerp(navMeshData.vertices[navMeshData.indices[t]],
navMeshData.vertices[navMeshData.indices[t + 1]], Random.value);
        Vector3.Lerp(point, navMeshData.vertices[navMeshData.indices[t +
2]], Random.value);

        return point;
    }
}
```

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

/*
Class to simulate explosions on the shredded cube
 */

public class ShatterOnImpact : MonoBehaviour {

    void OnTriggerEnter(Collider other)
    {
        if (other.tag == "Projectile")
        {
            Rigidbody[] rbs =
gameObject.GetComponentsInChildren<Rigidbody>();
            foreach (Rigidbody rb in rbs)
            {
                rb.isKinematic = false;
                rb.useGravity = true;
                rb.AddExplosionForce(30f, transform.position, 10f);
            }

            Destroy(other.gameObject);
        }
    }
}
```

```
using UnityEngine;
using UnityEngine.VR.WSA.Input;

/*
Top level manager class that recognizes the user's gaze and gestures, and
sends the signals appropriately to the interactable objects
 */

public class GazeGestureManager : MonoBehaviour
{
    public static GazeGestureManager Instance { get; private set; }

    // Represents the hologram that is currently being gazed at.
    public GameObject FocusedObject { get; private set; }


    private Vector3 message;

    GestureRecognizer recognizer;

    public static Vector3 gazedLocation;


    // Use this for initialization
    void Start()
    {
        Instance = this;

        // Set up a GestureRecognizer to detect Select gestures.
        recognizer = new GestureRecognizer();
        recognizer.TappedEvent += (source, tapCount, ray) =>
        {
            // Send an OnSelect message to the focused object and its
ancestors.
            if (FocusedObject != null)
            {
                FocusedObject.SendMessageUpwards("OnSelect");
            }
        };
        recognizer.StartCapturingGestures();
    }

    // Update is called once per frame
    void Update()
    {
        // Figure out which hologram is focused this frame.
        GameObject oldFocusObject = FocusedObject;

        // Do a raycast into the world based on the user's
        // head position and orientation.
        var headPosition = Camera.main.transform.position;
```

```
        var gazeDirection = Camera.main.transform.forward;

        RaycastHit hitInfo;
        if (Physics.Raycast(headPosition, gazeDirection, out hitInfo))
        {
            // If the raycast hit a hologram, use that as the focused
object.
            FocusedObject = hitInfo.collider.gameObject;
            gazedLocation = hitInfo.point;
        }
        else
        {
            // If the raycast did not hit a hologram, clear the focused
object.
            FocusedObject = null;
        }

        // If the focused object changed this frame,
        // start detecting fresh gestures again.
        if (FocusedObject != oldFocusObject)
        {
            recognizer.CancelGestures();
            recognizer.StartCapturingGestures();
        }
    }
}
```

```
using UnityEngine;

/*
World cursor to report and visualize the user's gaze (the direction they're
looking)
 */

public class WorldCursor : MonoBehaviour
{
    private MeshRenderer meshRenderer;

    // Update is called once per frame
    void Update()
    {
        // Do a raycast into the world based on the user's
        // head position and orientation.
        var headPosition = Camera.main.transform.position;
        var gazeDirection = Camera.main.transform.forward;

        RaycastHit hitInfo;

        if (Physics.Raycast(headPosition, gazeDirection, out hitInfo))
        {
            // If the raycast hit a hologram...
            // Display the cursor mesh

            // Move thecursor to the point where the raycast hit.
            this.transform.position = hitInfo.point;

            // Rotate the cursor to hug the surface of the hologram.
            this.transform.rotation = Quaternion.FromToRotation(Vector3.up,
hitInfo.normal);
        }
        else
        {
            // If the raycast did not hit a hologram, hide the cursor mesh.
            // meshRenderer.enabled = false;
        }
    }
}
```

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

/*
Class used to simulate holographic occlusion to the objects placed in the
hallway, with duplicate classes for each occluded area
 */

public class HallwayOccluder : MonoBehaviour
{

    [SerializeField] private GameObject hallway;

    void OnTriggerEnter(Collider other)
    {
        if (other.gameObject.tag == "MainCamera")
        {
            hallway.SetActive(false);
        }
    }

    void OnTriggerExit(Collider other)
    {
        if (other.gameObject.tag == "MainCamera")
        {
            hallway.SetActive(true);
        }
    }
}
```

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

/*
Object that manages the world anchor components attached to the gameobjects
 */

public class ManageAnchor : MonoBehaviour
{
    [SerializeField]
    private WorldAnchorManager anchorManager;

    [SerializeField]
    private bool active = false;

    [SerializeField] private GameObject debugCube;
    // Use this for initialization
    void Start () {


    }

    // Update is called once per frame
    void Update () {
        if (!active && anchorManager.AnchorStore != null)
        {
            anchorManager.AttachAnchor(gameObject, gameObject.name);
            active = true;
        }
    }
}
```

## Self-Checklist for Your Report

*Please check the items here before submitting your report. This signed checklist should be the final page of your report.*

- ☐ Did you provide detailed information about the work you did?
- ☐ Is supervisor information included?
- ☐ Did you use the Report Template to prepare your report, so that it has a cover page, the 8 major sections and 13 subsections specified in the Table of Contents, and uses the required section names?
- ☐ Did you follow the style guidelines?
- ☐ Does you report look professionally written?
- ☐ Does your report include all necessary References, and proper citations to them in the body?
- ☐ Did you remove all explanations from the Report Template, which are marked with yellow color? Did you modify all text marked with green according to your case?

Signature: _____