An exploratory object manipulation experiment using the core features of Hololens: A virtual library of augmented reality objects

Dragoş Circa[1], Alexandru Matei[1] **Alexandru Butean[2]**
[1] Connected Intelligence Research Center [2] Department of Computer Science
Lucian Blaga University of Sibiu
10 Victoriei Boulevard, Sibiu, Romania, 550024
{dragos.circa, alex.matei, alexandru.butean}@ulbsibiu.ro

ABSTRACT
In this paper we present a HoloLens application that allows the user to interact with virtual objects in a natural way, focusing on using only the built-in hand gestures and voice commands. The user can select what object to spawn and can change position, rotation and scale of the object. The objects are replicas to their real-world counterparts to increase immersion. They behave naturally when it comes to gravity and interaction with each other or the environment.

Author Keywords
Human-Computer Interaction, HoloLens; Augmented Reality; Head mounted device.

ACM Classification Keywords
H.5.1 Multimedia Information Systems: Artificial, augmented, and virtual realities

H.5.2 User Interfaces: Interaction styles, Prototyping

INTRODUCTION

Augmented reality (AR) is becoming a thing of the present with its support on various platforms, usage in various industries and captivating new ways of interacting with the virtual world. Such platforms include but not limited to: hand-held devices, desktop computers, IoT devices and head mounted devices using a head mounted display (HMD) with either see through or opaque display.

HoloLens [5] is a tether-less HMD AR device which uses a see-through display to “project” holograms into the real world. The device keeps track of the holograms position and visibility by continuously analyzing the user movement, head pose and environment occlusion. This works best at low speeds [7]. To better make use of what HoloLens is capable of we developed an application that allows the user to interact with virtual objects projected on top of real-world environment like in Figure 1. The interaction capabilities of the HoloLens are: gestures, voice commands and gaze. Also, collision of the user’s head with virtual objects can be implemented although not necessarily.

The build-in gestures are [6]:

- Air-tap – joining the index finger with the thumb while the hand is positioned in front of the user.
- Manipulation – moving the hand while index finger is joined with thumb - similar to dragging but in 3D.
- Navigation – same as manipulation gesture but with other purposes such as scrolling through a list.
- Bloom – joining all fingers of a palm together and then opening the palm. This gesture is used to put the current application to sleep and open/close start menu. This gesture cannot be remapped.

Using these gestures and speech recognition we were able to allow the user to control the following aspects of the virtual objects: position, rotation and scale.

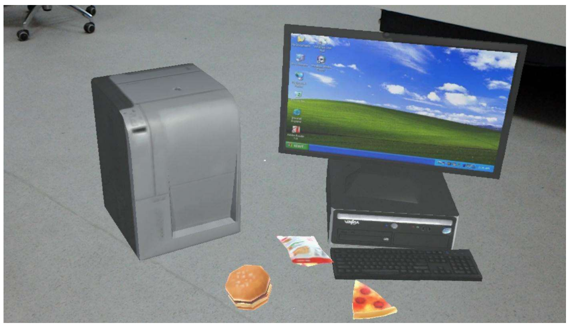

Figure 1. Real world scenario reproduced in AR

RELATED WORK

AR in order-picking – experimental evidence with Microsoft HoloLens [1]

The paper talks about the usability of HoloLens in an industrial setting by picking up certain objects under the guidance of the HoloLens device. The study aims to:

- Identify problems in the development of AR software
- General problems regarding the use of HoloLens glasses into a warehouse setting

- Examine user experience in different groups based on certain traits: age, gender, technical affinity

The first challenge was finding the right AR device to be used. Criteria that needed to be met were : a large field of view, long battery life, navigation precision, untethered movement. Only HoloLens met all these criteria. In order to improve the user experience, the task was "gamified" which consisted of adding badges and a leaderboard.

Another challenge was the fact that the user had to stand in a specific spot at the start of each course which cannot be always possible in a real scenario.

The data from 80 participants was collected during and after the experiment from which data from only 63 participants was used in the statistics. The result summarized is as follows:

- Men performed better than women.
- Age did not have a big impact on time to complete the course.
- Users with technological affinity had a better user experience but did not have a better time than those with no affinity.

The main issue for the subjects was the comfort, mainly the weight of the device. A HoloLens issue was the loss of tracking when coming too close to a shelf; also, voice recognition did not work properly in a noisy environment.

Augmenting Coding: Augmented Reality for Learning Programming [2]

The paper is a study that aims at comparing the usability, ease of use and interactivity differences between HoloLens AR, Mobile AR on iPhone and classic app in learning programming. The study was conducted on 12 participants, two females and ten males. The subjects had no programming background.

There were three tasks:

- Path finding: participant would code a program to direct an avatar
- More complex path finding
- Find a bug in a path finding algorithm

If a participant spent more than 8 minutes on a task, then that task would be marked as failed and completion time recorded as 8 minutes.

The result of the study was that Mobile AR was the fastest environment and head-mounted AR the slowest. The main reason was that users were already familiarized with mobile interaction while HoloLens brought an entire new set of interaction.

As far as ease of use is concerned most participants considered head-mounted AR as most immersive although the sloppy gesture recognition made interaction difficult and the device did induce dizziness on some participants.

The Virtual Twin: Controlling Smart Factories using a spatially-correct Augmented Reality Representation [3]

The paper describes a proof-of-concept application which runs on HoloLens and displays a 3D model of a factory. All the machines are represented by either a 3D model of them or by placeholders. When a user gazes at one specific machine additional state information is displayed such as physical properties, performance indicator, run-time information, error status etc. The user can interact with each machine.

To test the application a scenario was created using a smart factory and a service technician wearing a HoloLens device located 842.98km away. In this test case the technician remotely interacted with the machines by turning the stations on and off. The requests to the smart factory – factory model, state information, actions – were sent using a cloud interface.

This application of the HoloLens or any AR or VR device for that matter is meant to allow faster access to information. For example, "the machine at the far right-back" is much easier than finding the identifier of each machine. This means the interaction between supervisors and the factory is more intuitive which will result in increased productivity and less downtime.

Simulator Sickness in Augmented Reality Training Using the Microsoft HoloLens [9]

The paper talks about a study performed on 142 subjects to find the effects of using AR. The subjects were picked from three industries: aviation, medical and space. The initial hypothesis is that due to the visual clues of the real world the simulation sickness is less prominent due to the visual clues. In virtual reality studies show that simulation sickness occurs due to viewing moving images irrespective of the degree of realism (such as moving in VR virtually but standing still in the real world).

The test was performed using two software applications: a recorder and a player. The rest of the paper talks about how the study has been performed, what conditions were present while performing the study and observations along the study.

The conclusion is that there is almost no simulator sickness when using Microsoft HoloLens, although eye strain was present. Other symptoms are stress, boredom or general fatigue, although these might have been due to external factors.

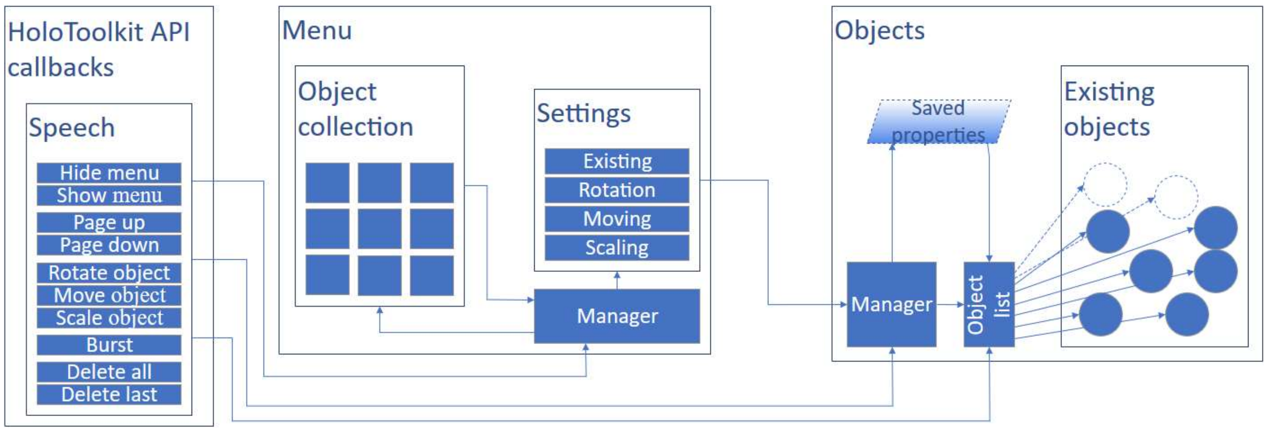

Figure 2. Application logic diagram

Bringing the Augmented Reality Benefits to Biomechanics Study [8]

The paper talks about a prototype of an application with which an observer can see the observed user's bones and soft tissue: muscles, ligaments and tendons. This is achieved by using a Microsoft Kinect for bones and skeleton recognition, apply the motions to rigged 3 models, simulate soft-tissue deformation and then render in 3D space on either HoloLens or a mobile device. It is explained in detail the flow of the application and how each of the before-mentioned steps are executed.

APPLICATION LOGIC

The application is made using Unity3D (v. 2017.2.4f) using C# for scripting. Besides Unity's built-in libraries, .Net 3.5 and Mixed Reality Toolkit [4] are used. The models used are from various free sources. The animations and animated objects were developed in 3DS Max.

Figure 2 shows the application logic diagram. It consists of 3 modules: a speech module that implements different HoloToolkit API callbacks, a menu to apply different settings and select the objects that will appear in the world scene and a module for the objects that are spawned and manages them over their lifetime.

When the application launches the user is prompted with a fixed UI to aid in positioning the HoloLens device correctly on the user's head. When the user says the phrase "Next" or air-taps anywhere the main scene is loaded, and the current's scene resources are cleared by Unity engine.

Menu

After the main scene is loaded the user can observe a 3D-floating menu that will position itself in the user's field of view as seen in Figure 3. The left-hand side of the menu presents nine blue squares with thumbnails representing various objects. These squares are arranged in a 3x3 formation. To be able to support a bigger number of spawnable objects a navigation panel consisting of 2 arrows and a page indicator have been added. On the right-hand side it is featured a settings menu from where the user can specify how the objects interact with the environment and how the user can interact with said objects. Also, buttons to remove last spawned object and all objects are presented.

The user can interact with any of the settings by gazing at it and then performing the air tap gesture. The position of the user's gaze is indicated by a small illuminated circle that follows the user's head pose. An alternative is to use speech commands although they are limited to a few simple commands. When the user presses the down navigation arrow, the next nine items from the objects array are assigned to squares. If there are fewer than nine objects, the extra squares are cleared of their previous objects and left blank.

The settings section is comprised of four groups. The first one controls global settings which are applied to all objects present. The next three groups determine what kind of interaction is currently enabled and various aspects of it.

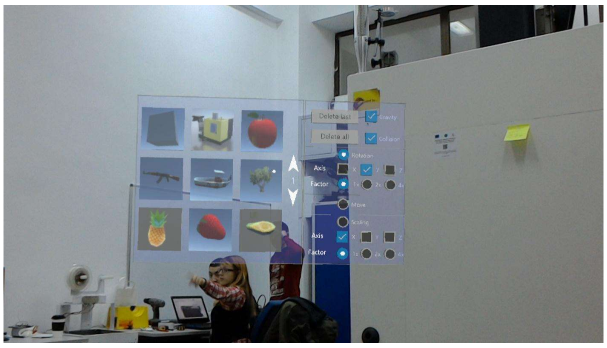

Figure 3. 3D floating menu

Objects

Unity uses prefabs to manage objects and their properties. In our application, each prefab is composed of several components: a 3D model, texture, collider, sound emitter and a script.

An object inherits all the proprieties of the prefab he is assigned to. This provides a means of customizing the objects with a certain ease. An example in this case is a bouncy ball, included in the set of spawnable objects.

When the user selects an object from the nine displayed in the menu a new copy is instantiated and placed in the world in the user's current field of view. One of the following four outcomes from Table 1 can occur depending on the current settings.

Gravity	Collision	Result
On	On	Object falls on the floor or on other objects (real or virtual)
On	Off	Object falls through all objects and the floor in to the void. The object is then removed when it's out of user's view
Off	On	The object will spawn and stay there as there is no momentum to it. It can be moved by throwing another object at it or by spawning another in the same place
Off	Off	The object will spawn and stay in place as there is no momentum to it. It cannot be moved by anything

Table 1. Possible spawn outcomes

Next, the instantiated object is added internally to a list for later management: to change its interaction type and properties (gravity, collision) or to remove the object from the world.

To aid the user in finding the object an arrow will be created that points to the latest spawned object.

Object lifetime

The object can be destroyed by four means:

- Pressing "delete all" button – removes all objects in scene
- Pressing "delete last" button
    - If the object is the last in the spawned objects list contained within *SpawnedObjectManager* script, then it is removed. If not pressing "delete last" removes the last object in the spawned objects. To remove a specific object, one must remove all objects spawned after his instantiation.
- Falling below -500 on the y coordinate. This can happen due to mapping errors or to collision setting
- Closing the application

Interaction

An object interacts with the environment as one could expect from any real-world object. The main difference is that while the objects interact with the environment, the real environment does not interact with the virtual environment. A simple example would be throwing a ball at a chair: the ball would bounce and the chair will move. In this application, the virtual ball will bounce of the real chair, while the chair remains unaffected. The real ball to real chair interaction can be replicated when virtual objects interact with each other (in ex: virtual chair with virtual ball). If the objects properties are set correctly - weight, drag, etc. – the simulation should behave naturally.

Because only the hand position is used in this HoloLens iteration there is no logical way to perform all three manipulations at the same time, so a three-way toggle is implemented. The active gesture is activated with the UI or by speech.

The user can interact with the objects by gazing at them and by performing air-tap and moving the finger while air-tapping. This is known as dragging and it results in different actions depending on what the current interaction mode is set to: translation, rotation or scaling

Movement

To enter this mode the user must either say the command "move object" or select the move option from the settings panel.

When the dragging motion is performed with the hand (object C) the object (object D) moves corresponding to the moving of the finger. The movement is similar to perspective projection where the projection plane is the plane defined by the finger (object C) and head pose (object A). This ensures a good user experience and prevents fatigue of moving distant objects. As can be seen in Figure 4, the arrow 1 is shorter relative to the user than arrow 2.

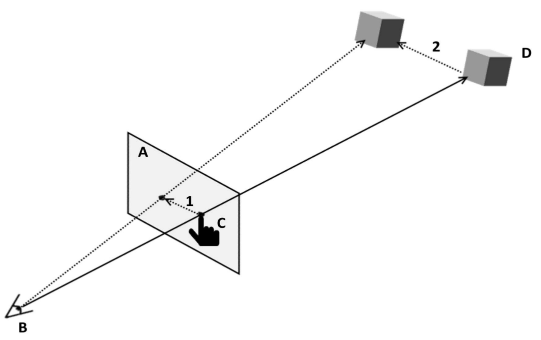

Figure 4. Relative movement of objects

Rotation
The user begins by tapping on the object and holding. Then by performing the dragging action different rotations will be applied on the object, as in Table 2.

Hand movement	Quaternion mapping
Left and right (x axis)	Y rotation – rotate to left or right
Up and down (y axis)	X rotation – tilt forward or backwards
Forward and backwards (z axis)	Z rotation – tilt to right or left

Table 2. Rotation actions

The resulting quaternion is then multiplied by the rotation sensitivity and then added to the object's current rotation. Rotation sensitivity can be changed from the settings panel from the rotation group to one of the three factors: 1x, 2x or 4x. Unity handles rotation past 360 degrees and below -360 by resetting it to 0.

A challenge we encountered was the shaking of the user's hand, which added rotation on undesired axes. This problem was solved by reducing the rotation sensitivity in those cases and by locking the rotation of the object to one or more axes.

Scaling
The hand movement is multiplied by the scaling sensitivity factor and then directly applied to the current object scale vector. This vector is continuously applied while the user is in scaling mode and didn't end the dragging motion. This allows the user to reference the initial scale of (1,1,1). If the object has been previously scaled, then the scale vector is applied to that scale.

Spatial perception
To add immersivity to the application we used the mapping provided by the HoloLens API. To occlude the objects - such as objects behind real objects – a black material has been used for the mapping model.

All the sounds are 3D as well. This means that the sound intensity diminishes relative to the user's distance to the source. This applies to each ear independently which helps the user in locating the objects.

Speech recognition
The speech recognition component of the application is made using HoloToolkit API which in turn relies on Windows speech recognition library. In Table 3 we presented the implemented speech commands and their actions.

Command	Action
Rotate object	Sets the interaction mode to rotation
Move object	Sets the interaction mode to move
Scale object	Sets the interaction mode to scale
Hide menu	Hides the menu
Show menu	Shows the menu
Delete all	Removes all objects in the scene
Delete last	Removes the last spawned object
Page up	Navigates to the previous page of items in the menu objects collection
Page down	Navigates to the next page of items in the menu objects collection
Burst	Spawns 20 objects and applies momentum to all

Table 3. Possible voice commands and their actions

Other features

Animations
Certain animations are implemented with the goal of improving the user experience. The most noticeable animation is the one applied to the menu. It runs as follows:

1. Spawn: The back panel of each section – object collection and settings panel – are spawned as little cubes.
2. Expand: Each cube expands on its X coordinate:
    a. Horizontal expansion
        i. Left cube will become object collection back panel and expand to the left
        ii. Right cube will become settings panel and expand to the right
    b. Vertical expansion: both expanded cubes now expand vertically, transforming themselves into the back panels of the menu

    c. All the borders are spawned and expand towards the user.

3. Population: the controls are populated on the menu
    a. All the squares in the object collection appear instantly
    b. The groups from the settings panel are displayed sequentially from top to bottom

When the close command is issued, the animation is played again but in reverse. While the animation is playing the panel will keep following the user's field of view.

Object indicator

The arrow created when an object is spawned has an animation flow on its own:

1. Spawning: a small cube is spawned indicating the direction of the newly spawned object.
2. Expansion vertically: the small cube expands horizontally on its X axis in both directions.
3. Offsetting edges: this will turn the deformed cube into an arrow.
4. Scaling down: not to clog the user's view.

The arrow will stay on the edge of the user's view indicating the location of the newly spawned object. This is useful when the spawn point is either outside of the user's field of view, the object is occluded or it falls down too quickly due to gravity.

CONCLUSION

In this paper we described the interaction capabilities of a HoloLens device using a proof-of-concept application and its inner workings.

While more interaction types can be implemented, we focused on exploiting the built-in ones as much as possible through familiar commands, gestures and objects. User experience was also an important factor, as this application can be used to introduce new users to the HoloLens and by extent, to the augmented and mixed reality worlds.

ACKNOWLEDGEMENT

This work is supported through the DiFiCIL project (contract no. 69/08.09.2016, ID P_37_771, web: http://dificil.grants.ulbsibiu.ro), co-funded by ERDF through the Competitiveness Operational Programme 2014-2020.

REFERENCES

1. Bräuer, P., and Mazarakis, A. AR in order-picking – experimental evidence with Microsoft HoloLens, *Conference: Mensch und Computer 2018, Pages 361-368*, 2018.
2. Dass, N., Kim, J., Ford, S., Agarwal, S., and Chau, D. H. Augmenting Coding: Augmented Reality for Learning Programming, *Chinese CHI '18 Proceedings of the Sixth International Symposium of Chinese CHI, Pages* 1-4, 2018.
3. Kritzler, M., Funk, M., Michahelles, F., and Rhode, W. The virtual twin: controlling smart factories using a spatially-correct augmented reality representation, *IoT '17 Proceedings of the Seventh International Conference on the Internet of Things, Article Nr 38*, 2017.
4. Microsoft, Github, [Online]. Available: https://github.com/Microsoft/MixedRealityToolkit-Unity
5. Microsoft, HoloLens, [Online]. Available: https://www.microsoft.com/en-us/hololens
6. Microsoft, Mixed Reality - Gestures, [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/gestures
7. Saddik, A.E., Zhang, L., Dong, H., and Liu, Y. Technical Evaluation of HoloLens for Multimedia: A First Look, *IEEE Multimedia, Volume 25, Issue 4, Pages* 8-18, 2018.
8. Voinea, A., Moldoveanu, A., and Moldoveanu, F. Bringing the Augmented Reality Benefits to Biomechanics Study, *MVAR '16 Proceedings of the 2016 workshop on Multimodal Virtual and Augmented Reality,* Article Nr 9, 2016.
9. Vovk, A., Wild, F., Guest, W., and Kuula, T. Simulator Sickness in Augmented Reality Training Using the Microsoft HoloLens, *CHI '18 Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems,* Paper nr 209 , 2018.