LegoLens: LEGO for the Microsoft Hololens

Thomas Nyegaard-Signori
sfq340@alumni.ku.dk

Tobias Carlos Tvarn
@alumni.ku.dk

Tor-Salve Dalsgaard
mhb558@alumni.ku.dk

Enes Golic
qzj710@alumni.ku.dk

ABSTRACT

This project set out to produce an augmented reality application using LEGO on the Microsoft Hololens. Using the open source toolkit produced by Microsoft called Holotoolkit a prototype was built. (SKRIV NOGET OM USER TEST).

Author Keywords

Microsoft Hololens; Holotoolkit; LEGO; Augmented reality.

INTRODUCTION

This report focuses on an application for the newly released Microsoft Hololens. The Hololens was released around march 2017, and the fact that the product is in its infant stage and is based on a very new technology opens up possibilities and removes any preconceived notions about which applications and uses the Hololens might have.

The technology is relevant with regards to mobile computing in one very apparent way, in that it is a wearable, computing unit. Other than that, it offers alternate reality (AR) possibilities because of its partly see-through screens and user tracking. Since the Hololens is still a new technology, the mobility and computing power of the product will most likely increase, making it resemble a ubiquitous computer more and more. This evolution of mobile computing was one of the reasons that the Hololens was chosen to develop on in the first place.

The application this report will cover revolves around LEGO. LEGO is a way for kids and adults to build constructions, vehicles and scenery, all in a very physical and three-dimensional way. This, then, seemed like a natural choice for an AR application, since the application layer between the user and the world could expand naturally on the possibilities and limitations of the physical, "real-world" LEGO.

The application in itself should be a sort of digital playground in which a user could interact with LEGO in ways they would find natural. Sticking pieces together the way they do in real life, stacking and constructing, all interactions that the user knows well from having played around with real LEGO. This was done using interaction through a virtual tablet, known as the "Generator board" and simple drag-and-drop with the bricks. We end up with a rough prototype which helped us discover the pitfalls and consideration concerning a LEGO implementation in an AR enviroment.

RELATED WORK

To ease the interaction between the user and the virtual layer presented by AR, one could think of different interaction techniques, so as to not rely on the gesture recognition of the HoloLens.

The implementation with the Hololens presented in [7]. This small, non obtrusive wristband that the prototype emulates could provide another form of interaction, which some users might prefer to the visual tracking of the HoloLens.

Another, non-obtrusive way of sending inputs to the HoloLens could be with the use of very small devices using the interaction technique presented in [2]. These devices could act as a secondary way of clicking instead of the "tap" gesture made available by the HoloLens.

Providing visual feedback to the user is essential for an AR application to provide a meaningful experience, but since the line is blurred between real world and virtual world, the question of natural feedback remains. This is presented in both [**?**] and [6], though in wildy different approaches. [**?**] approaches the natural feedback by means of kinesthetic feedback and [6] approaches it by means of haptic feedback. Using these modalities could provide the HoloLens with a more natural feeling when interacting with the objects in the virtual world.

A possible extra connection between the virtual and real world could be by including and extending real world objects into the virtual world. This is done in [3], and could help with creating a link between the two worlds.

DESIGN CHALLENGES

Accessibility

To manoeuvre around in any application, a menu is needed. The menu is something every user has experience with and it is the first thing a user is met with when accessing an application. This means that the menu has to consist of certain classic elements.

A user needs a way to close the application. On mobile phones nowadays this can be done with 'return' buttons on the phone, but applications generally have a built in exit function.

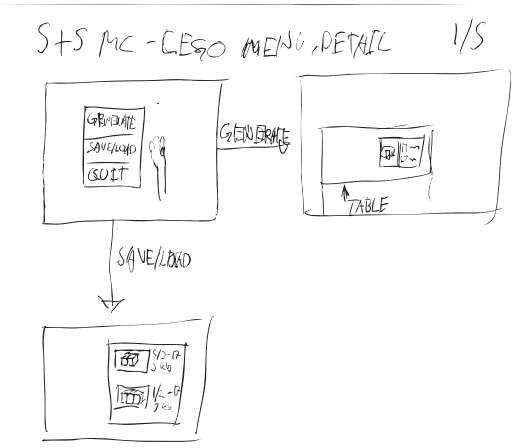

Figure 1. Initial sketch of a separate Main Menu

The user also needs to have some sort of options menu and guidelines. Stacking LEGO seems simple and intuitive, but all the operations and possibilities is something that can confuse a potential user.

Lastly it shouldn't be complicated for the user to start a new LEGO session.

A Main Menu

To achieve accessibility a '5 plus 5' sketch generation session was held. Initially we thought about this application as a game, therefore the user would typically be met by a main menu screen, as figure 1 depicts. The user would open the Hololens application, and be met with a menu screen, where actions such as generating a play area, loading/saving a game and a way to quit the application were possible.

One common theme in the sketching of the main menu was separating the design of the menu and the interaction techniques with the menu. The fact that we were working with the Hololens, led to some sketches being focused on the interaction, such as figure 2 depicts, while other sketches solely focused on the design of the menu.

The possibilities with the HoloLens gestures and the augmented reality made it apparent that the menu had to be moveable, either by closing and opening it with the available gestures, or the ability to place it in favorable self-chosen spots.

The Generator

After discussing menus in the context of the Hololens, it became apparent that there was a need for a menu that could be placed and interacted with in the real world. This menu should only interact with the play area, and have functions tied to the bricks. Figure 3 is a sketch of the functionalities the generator could have, such as brick types and templates. The generator should be have functionalites that can alter the blocks, but also needs to be interactive and moveable in the play area.

As Figure 4 suggests, the generator could be placed on a table and used for generating bricks from that spot. We designed the generator as a tablet device. Being a solid familar object, makes it more intuitive and natural for the user to interact with.

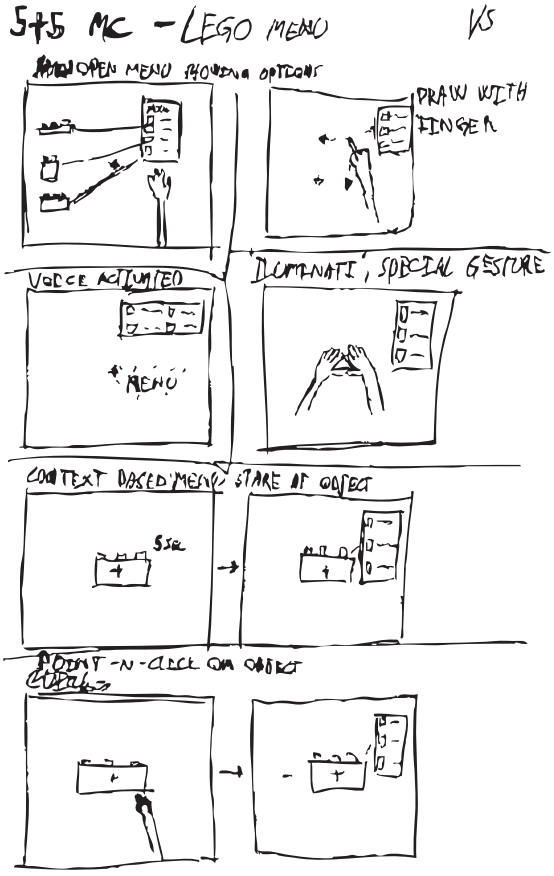

Figure 2. A focus on gestures rather than design, impacted the design. The figure shows different ways of creating an interactive menu with gestures.

To lift, move and place the tablet on a surface is a simple and familiar task for the user. To accompany the tablet device, a simple interface was designed. The interface only needed simple function all tied to spawning lego bricks. The user can spawn bricks of different types, change colors, save/load a play scene and spawn different prebuilt templates.

Final Design

During the sketching phase, several design choices were discussed. One of the very first concerns was the overall look of the main menu and the generator. The choice of big buttons, clear visual cues and short textual descriptions were present in almost all of the sketches in the early design phase. The need for a simple interface was a must, as stacking LEGO bricks shouldn't be a hassle.

We wanted an effective and efficient process of interacting with LEGO. We also wanted the users to be satisfied when using our application. Therefore the menus should not be overcomplicated in functions and design, but not sparse in functions either. To achieve these 3 key elements we designed the menu content with simple LEGO thumbnails, that explained what each button did along with simple textual descriptions.

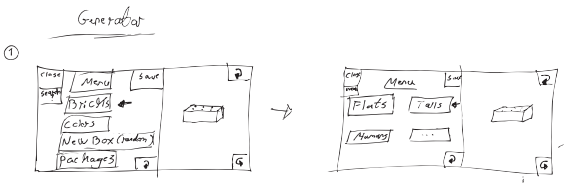

Figure 3. An initial sketch of how the generator could look like. The menu screen has alot of functionalities, and the actions chosen happen on the right side.

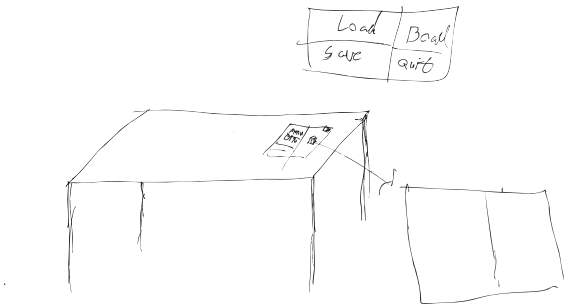

Figure 4. First sketch of the generator as a movable object. Here placed on the table, spawning blocks on the table.

As the development of the application progressed it became more apparent that an actual main menu was not necessary. Using way too many menus in our simple interactive application would hinder the effectiveness and efficiency for the user, because of the limited functions our prototype had to offer.

To circumvent a clonky and cluttered experience for the user we decided on creating a tablet looking generator. Using a single solid object with all functionalities in the same place, would ease the user interaction. Instead of going through a main menu and then having the generator menu, which contains the functionality for working with the LEGO bricks, we decided on spawning the generator at the application start up. Due to the way our prototype works it seems as a natural choice to "cut out the middleman", here the main menu.

Granted, with an eventual increase in functionality and complexity of the application, a root/main menu might prove useful as to not clutter the users experience when they are building as opposed to when they are in the main menu, where setting options, loading scenarios, downloading templates etc. would be a valid solution.

In the final design the user is met by a tablet device, consisting of large buttons with thumbnails and text explaining what each action does. As figure 5 shows, the user can choose different brick types and colors, as well as having a sandbox mode where a random amount of bricks are spawned to play with. The user can also spawn prebuilt templates, save/load previous sessions and go back to the main menu with a 'back' button.

IMPLEMENTATION

Language and tools

The LegoLens application is built using Unity, where custom scripts are written in C#. Unity is chosen, since it is used in

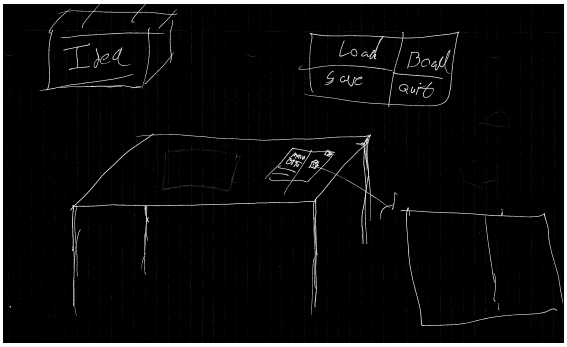

Figure 5. First sketch of the generator as a movable object. Here placed on the table, spawning blocks on the table.

Microsofts own Hololens academy.[1] Microsoft also provides a software toolkit called HoloToolkit[2], which binds Unity and the Hololens together.
For initial testing the Microsoft Hololens Emulator was used, while in later stages the application was deployed directly to the Hololens device.

HoloToolkit

The HoloToolkit provide implementations of common tasks in developing applications for the Hololens. This kit implements functionality, which is mostly Hololens specific, such as spatial mapping and understanding. Also functionality for input methods are provided. The Hololens comes with two build-in gestures: the bloom and the tap.

Spatial mapping and understanding

These two functionalities are fundamental for the use of the Hololens, since they let the Hololens understand the shape of a room. Thereby virtual objects can be placed in the real world. The toolkit provides prefabricated functionality for spartial mapping and understanding, which simply can be added in Unity.
The Hololens uses a depth camera similar to the camera build into the Kinect v2 and four cameras meant to understand environment.[3] The spatial mapping relies on the time-of-flight Kinect-like depth camera and the four RGB cameras to provide a robust tracking of the enviroment.
This connection between the real-world and digital playground created in the application was essential, both to the experience but also to make the application augmenting the reality. The mapping allows to create a relative coordinate system, such that virtual objects can be placed in accurately.

Input methods

The Hololens tracks the gaze of the wearer, by the assumption that the wearer always looks straight ahead. This is the limitations of the Hololens, which does not provide augmentation for the whole field of view. Thus if the wearers eyes would be tracked, the wearer could gaze outside of the Hololens field of

[1] https://developer.microsoft.com/en-us/windows/mixed-reality/academy
[2] https://github.com/Microsoft/HoloToolkit-Unity
[3] https://developer.microsoft.com/en-us/windows/mixed-reality/hololens_hardware_details

view. In this application a cursor appears where the gaze of the wearer collides with mapped objects in the real world or objects created by the application.
In this application the tap method is used. Here there are two different states the interaction can have: a single tap and dragging.
The single tap is used to navigate in the menu on the generator. The HoloToolkit provides functionality, such that objects can be dragged by holding tap. This is used to drag LEGO bricks and the generator around.

Unity scripts

While the HoloToolkit implements the augmentation of objects, the custom Unity scripts implement LEGO functionality. These scripts control the behaviour of the LEGO bricks, the generator and templates.

Generator

The generator script contains the main functionality of the implemented code. Menu interactions are implemented here and thus the necessary functions to instantiate and model bricks. The instantiation is done by loading the prefabricated LEGO brick object, and colouring it depending on user input. A brick is not affected by gravity until the user interacts with the brick. For simplicity the number of shapes a brick can have is limited.
Templates are implemented in a similar way as bricks, except that these can not be coloured. Also the sandbox feature is implemented in the generator script. It generates 20 bricks with random shape and colour.
The menu on the generator works by activating and deactivating the appropriate menu items.

Bricks and Templates

The brick and template scripts work in a similar ways. They both make sure that the generated objects do not rotate and do snap to a grid. An one-by-one brick is 0.8cm long and wide and 0.96cm high (with knobs). Thus bricks are forced to lay in positions which are a multiples of 0.8.

EVALUATION

participants test setting - hypothesis evaluation

The goal of the test was to test two hypothesis regarding HoloLens and the application:

H1: The user will be confused/overwhelmed by the virtual layer that the HoloLens provides

H2: LEGO is well suited for the HoloLens

We had two test participants, male and female from different institutions, both of age 23. Chocolate was offered as a reward but was not taken. These were then put through a three-part test, the first part acted as a training session for the actual interaction with the HoloLens application, the second one was different tests in the application wearing the HoloLens and the last part was a short questionnaire.

Since we wanted to test whether or not the HoloLens was too much of a overwhelming experience for the user, the tests were structured so as to minimize any complications with the application itself.
The first part, the training session, was created using Prototype on Paper[4] and physical representations of LEGO bricks. The setup was meant to emulate the view in the HoloLens app as much as possible whilst keeping the overall test setup as low-fidelity as possible. The users were asked to go through most of the menus functionality in short, manageable task which were meant to prime them for the same task in the HoloLens application.
The second part of the test was in the application itself after a brief introduction to gestures using the HoloLens app "Learn Gestures". This was done to give the test subjects a complete overview of the interaction possibilities with the HoloLens. The test subjects were asked to do simple tasks, such as spawning a brick of a chosen color and shape or instantiating a template.
The third and final part was a short questionnaire consisting of Lickert scale and free text questions. These questions were made with the hypothesis in mind without imposing a bias on the subjects.

The evaluation on the results were done in a non-formal way as opposed to the statistical methods described in the lectures (ie. ANOVA, Kruskal-Walls Test etc.). This was done because our tests were not of a comparative nature, and thus we did not find any way of imposing a strict statistical test on our (small!) test sample. Our test results are based solely on the subjects experiences and opinions and are therefore to be taken with a grain of salt. This test structure came to be because of our hypothesis.

Results

All the Lickert scale questions were done on a 1-5 scale.
Both participants found the menu very easy to navigate (both gave a 5 on the Lickert scale), and they both agreed on the application/HoloLens being able to track their movements and position satisfyingly. The subjects disagreed on the precision of the placement of the brick, and when asked about this precision one subject said that he/she did not feel comfortable with the interaction since he/she had experienced beforehand that bricks and generators could fall through solid, real world objects.

Observations

One user noted that the HoloLens' tracking was thrown off balance when looking into lights, and this confused him/her unnecessarily. Both users felt dizzy when they took of the HoloLens and complained about a queeziness when having the HoloLens on for 15 minutes or more.
Both user thought that a LEGO application would suit the HoloLens/AR in general, because of the possibilities, although the size of the bricks and projects as a whole would have to be considerably larger than real world LEGO bricks. Both participants agreed on the size of the bricks presented in the high fidelity prototype was appropriate, although they were not presented with alternatives.
It was observed that one of the subjects, immediately after

[4] https://marvelapp.com/pop/

dropping the generator on the ground, used the "bloom" gesture instead of looking down, since he/she thought that the application had crashed. This could indicate that H1 has relevance and should be investigated further.
One user noted that he/she was unsure whether the control of the brick was done using the gaze or finger dragging.

DISCUSSION

Confusion and overwhelming impressions (H1)

One aspect of the conducted tests was the apparent wow-factor of the HoloLens and virtual layer. The tests and observations showed that the subjects could not utilize the fact that the virtual layer extended beyond their apparent field-of-view. This resulted in many "out of sight, out of mind" situations, where the subjects tried to restart the application although nothing critical had happened. This can of course be explained and mediated using markers and other helping layers to indicate where the points of interest in the application may be located, but this would then in turn clutter and already sparse field-of-view. To ease the subjects into thinking in an AR setting we used the already existing "Learn Gestures" application, but this did not seem to have a strong beneficial impact on that issue.

Suitability of LEGO in AR (H2)

One of the hypothesis introduced in the previous section was the suitability of LEGO in an AR setting. This question was asked rather late in the development process. What became apparent was that the virtual LEGO in an AR setting would not be able to provide the same "finicky" feel that LEGO has, sitting at a table, obsessing over small details in an advanced setup. This is because of the computational limitations of the HoloLens and the technology architecture. The minimal rendering distance of the HoloLens is, right now, much larger than the distance one would be from a real-life LEGO project, ie., arms length. This limitation demands a much larger brick size than real-life LEGO and this in turn limits the overall complexity of a virtual LEGO project.
These considerations became our main focus in the user tests, whether or not the HoloLens is applicable in these small scale, high detail applications such as LEGO. High detail, in this sense, refers to the many ways that LEGO bricks can be joined together to create more and more complex structures. With these structures comes problems, such as occlusions, tracking precision and stabilization issues.

FUTURE WORK

LEGO size and compexity

To combat the issue with the size of actual LEGO and the complexity available in this scale a virtual magnifying glass could be envisioned. Using a special gesture, a certain area of the LEGO bricks could be brought into view using a secondary "screen", a menu that could show a zoomed in version of the actual view the user has. (Er det lort?)

Brick shortage

We have a severe shortage of bricks available in our prototype. Anyone who considers an AR LEGO application will soon have to think of which blockset or types they would include.

		Test Conditions	
Name	*First*	*Second*	*Final*
Marsden	223.0	44	432,321
Nass	22.2	16	234,333
Borriello	22.9	11	93,123
Karat	34.9	2200	103,322

Table 1. Table captions should be placed below the table. We recommend table lines be 1 point, 25% black. Minimize use of table grid lines.

The amount of distinctive LEGO bricks is staggeringly high, and this sort of work would probably benefit greatly from working together with LEGO as to get dimensions and oddities right. This work would also make such an application much more attractive, as one of the strongest selling points of LEGO is the variety but ensured compatibility.

REFERENCES

1. Ronald Azuma, Yohan Baillot, Reinhod Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. 2001. Recent Advances in Augmented Reality. (2001).
2. Patrick Baudisch and Gerry Chu. 2009. Back-of-Device Interaction Allows Creating Very Small Touch Devices. (2009).
3. Jens Grubert, Matthias Heinisch, and Dieter Schmalstieg Aaron Quigley. 2015. MultiFi: Multi-Fidelity Interaction with Displays On and Around the BodyReality. (2015).
4. Si Jung Jun Kim. 2012. A User Study Trends in Augmented Reality and Virtual Reality Research. (2012).
5. Pedro Lopes, Sijing You, Lung-Pan Cheng, Sebastian Marwecki, and Patrick Baudisch. 2017. Providing Haptics to Walls and Heavy Objects in Virtual Reality by Means of Electrical Muscle Stimulation. (2017).
6. Samuel B. Schorr and Allisson M. Okamura. 2017. Fingertip Tactile Devices for Virtual Object Manipulation and Exploration. (2017).
7. Srinath Sridhar, Anders Markussen, Antti Oulasvirta, Christian Theobalt, and Sebastian Boring. 2017. WatchSense: On- and Above-Skin Input Sensing through a Wearable Depth Sensor. (2017).
8. Yang Zhang, Junhan Zhou, Gierad Laput, and Chris Harrison. 2016. SkinTrack: Using the Body as an Electrical Waveguide for Continuous Finger Tracking on the Skin. (2016).