3D Interaction and Virtual Environments: Lego Builder

Report

Koen De Backer
Hasselt University
Expertise Centre for Digital Media Diepenbeek, Belgium
koen.debacker@student.uhasselt.be

Sven Coppers
Hasselt University
Expertise Centre for Digital Media Diepenbeek, Belgium
sven.coppers@student.uhasselt.be

Jacqueline Galeazzi
Eindhoven University of Technology
j.galeazzi@student.tue.nl

Juan Jose Martin Miralles
University of Balearic Islands
joanmartinmiralles@hotmail.com

Linsey Raymaekers
Hasselt University
Expertise Centre for Digital Media Diepenbeek, Belgium
linsey.raymaekers@student.uhasselt.be

1. INTRODUCTION

2D user interfaces combined with traditional input devices such as the keyboard and mouse have long become a commodity, but HCI researchers from around the globe are only beginning to discover the potential of 3D user interfaces, 3D interaction techniques and their required input devices [4].

Lego Builder is an experiment in which the user can use 3D gestures to play with the virtual equivalent of traditional Lego bricks. Building large structures can be time consuming. This brought the authors to focus on speed of performance. The user interface is based on the literal approach [4], because a 2D menu and a 3D virtual environment are combined.

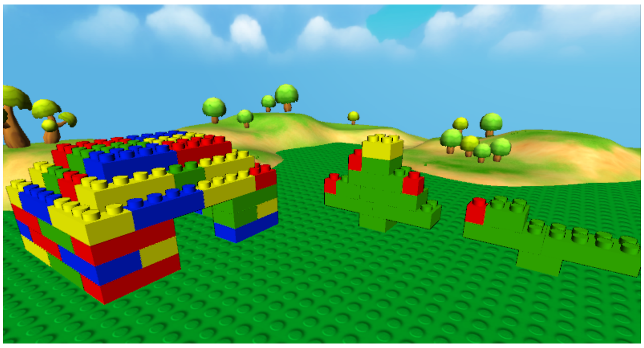

Figure 1: An example of what can be built within the Lego Builder

2. HARDWARE AND SOFTWARE

We have chosen to use the Leap Motion device [1] as the input device for the Lego Builder because it allows for accurate tracking of the user's hand gestures. This will be necessary since building with Lego asks for precise selection and manipulation. The Leap Motion has a small range and it can track gestures within a confined area above the device. For the current application we found this range to be sufficient.

The Lego Builder was created with the Unity game engine [3]. Unity contains a plug-in for the Leap Motion, thus data from the Leap Motion could easily be accessed within Unity.

3. CHALLENGES

Fine hand movements as input for the control of interfaces, games and 3d environments are not yet commonly used. Therefore, standards have yet to be developed of which hand gestures work well and how these gestures should be implemented. The creation of the Lego Builder is thereby faced with a number of selection, manipulation and navigation challenges.

The original plan was to use a combination of Leap Motion and Kinect [2]. The Leap Motion would allow for precise manipulation, whereas the Kinect could be used for big movements because of its range. While testing both devices we came to the conclusion that the Leap Motion would be sufficient for using the Lego Builder, because we were able to accurately control objects and environments using the Leap Motion alone. The challenge of combining input devices was therefore omitted.

Out of the many possible challenges to implement, we have decided to mainly focus on the ones described here.

3.1 Selection Challenges

In an environment filled with Lego bricks the user has to be able to select just one brick. To select a brick, the user can move the hand near a brick and make a grabbing gesture (form a fist). This causes the system to select the highlighted brick and attach it to the hand as long as it remains a fist. The use of a grabbing gesture for selection leaves pointing gestures available for controlling the menu. This approach is recommended by Bowman et al. [4].

2D menus are used for adding new bricks and changing brick color. The menus are displayed on the right and left sides of the screen. Originally, the users would be able to drag bricks from the menu to the environment in order to place them. However, Bowman et al. suggest that pointing works better for 2D menus [4]. The application was modified to remember the previously selected brick type and the grab gesture was allocated to create a new instance of the remembered brick type. Thus, a grabbing gesture is used for both selecting existing bricks and placing new bricks. Whenever a new brick is placed in the environment a short sound provides feedback that the brick was indeed successfully placed. The color menu works in the same way as the brick type menu: the selected color is remembered and used for future instances of the bricks. In order to avoid faulty recognition of gestures, we have allocated the grabbing gesture (selection and placement of bricks) solely to the dominant hand, leaving the non-dominant hand free to perform secondary gestures during placement of bricks.

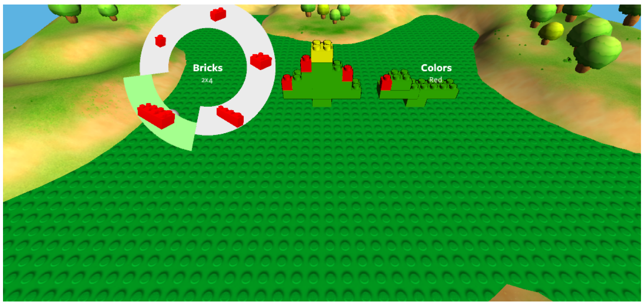

Figure 2: 2D menus for adding bricks and choosing colors

3.2 Manipulation Challenges

Selected bricks can be moved by navigating the virtual volumetric hand in the desired direction. The Lego Builder provides feed-forward in the form of a highlighted preview of the block placement, as can be seen in the Figure 3. As with real life Lego, there is a limitation in the ways Lego bricks can be placed on top of each other since bricks have to interlock. These constraints lower the required precision, which enhances the user's performance [4]. For example, the required precision is lowered by a horizontal grid on which bricks need to snap. Snapping makes sure that bricks are always placed on top of other bricks or on the ground-plane in the correct (interlocking) way.

Because of the way bricks interlock, we have decided to constrain the user in the possibilities of rotating a brick. For example, there is no use in rotating a brick by 45 degrees, since the brick would not fit onto other bricks or the ground-plane. In Lego Builder, bricks can only be rotated by 90 degrees along the y-axis. This constraint improves the performance as well. Rotation is done by making a pointing gestures with the non-dominant hand, followed by a circling gesture.

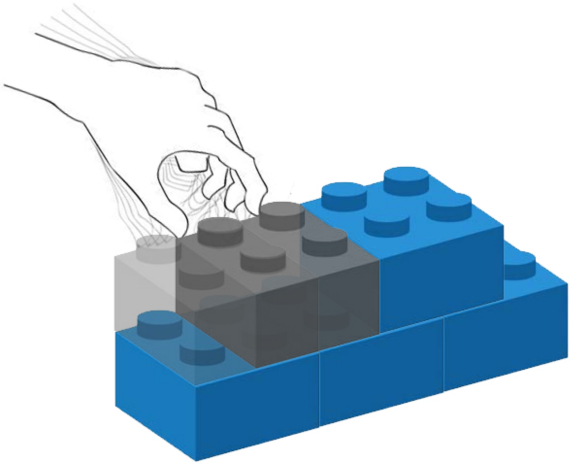

Figure 3: Snapping with feed-forward of brick placement

Finally, we implemented two gestures that aid in the building of walls. One gesture places bricks next to each other in a straight line, the other stacks bricks vertically. To place bricks in a line, the user performs a grabbing gesture with his dominant hand to start brick placement, and then holds his non-dominant hand open horizontally. Using bimanual asymmetric gestures is generally preferred by users [4]. The dominant hand is used for the brick placement gesture, as it is usually better trained than the non-dominant hand. The non-dominant hand is used to indicate the building state (horizontal or vertical), which requires only a small amount of muscle movement. The user can now place an *anchor* brick freely, after which more bricks are spawned automatically in a line after each consecutive grabbing gesture. The same procedure holds for placing bricks vertically, but the non-dominant hand is held open vertically. Using these gestures, the user only needs to put effort in placing the first brick precisely on the grid, and then place more bricks without effort. After each new brick, the camera is animated to move to the newly placed brick. This form of passive travel also prevents *teleportation*, which can cause the user to become disoriented [4].

3.3 Navigation Challenges

Precise navigation at small distances and fast navigation at large distances is one of the most important navigation challenges. To allow both precision and speed, the camera remains stationary while the user's hand is well within the Leap Motion's range. As soon as the user moves his hand towards one of the borders of the Leap Motion's range, the camera moves in the matching direction. The camera will accelerate over time when the hand is near a border, until a threshold is reached. This allows for fast navigation over large distances.

We realized that rotating the camera around multiple axes would not be the biggest priority for the Lego Builder application. Therefore the camera rotations were restricted, and made more hand gestures available for the more trivial functionalities. The camera can only be rotated by increments of 90Âř around the X-axis. This again lowers the required precision, as the camera is rotated by using a vertical swipe gesture with the dominant hand. The swipe gesture would not allow for precise rotation manipulation, as the hand movement cannot be directly translated to rotation degrees. The rotation is restricted to the dominant hand because we noticed a lot of false positives when for example performing vertical wall placement, where the non-dominant hand is also held vertically.

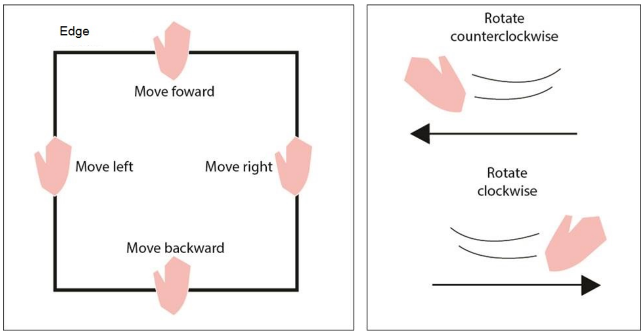

Figure 4: Navigating the environment and rotating the view

Speed of performance is the most important challenge. Therefore the combination of multiple user actions into a single gesture is considered. The user can move a brick by grabbing it. When using small hand movements, the camera will remain stationary and only the brick will move. Larger hand movements have an additional behaviour: the camera starts moving when the virtual hand reaches the borders of screen. While the camera is moving, the virtual hand and the brick will move together with it and the user is able to travel. This is how the Lego Builder enhances the speed of performance by combining navigation and manipulation tasks. This same technique is used for navigating while placing a new block.

4. PROTOTYPE

4.1 Evaluation

As an informal evaluation we had three of our fellow students perform several scenarios with the Lego Builder. First they received instructions on which actions could be performed and the available gestures. After that they tried creating straight lines of bricks, stacking bricks and selecting bricks in a dense environment with increasing difficulty. As a last assignment they were asked to recreate the object shown in figure 4, the first few bricks were already placed. The evaluation revealed some selection and manipulation issues, the biggest issues and limitations are listed below. Some of these issues have already been resolved, for others we give possible solutions.

Selection

- When selecting a brick from the environment, the predictive highlighting works correctly. However, as the selection action is performed, the wrong brick is selected or a new brick is placed. This makes selection in dense environments difficult. The issue is caused by a small shift in the center of hand position while performing the grabbing gesture. A solution would be to use the last position where the hand was registered as being open and discard the position of the hand while being closed.

- The test subjects had difficulty selecting the smallest brick type (1x1 brick), as both the brick and the selection preview were obscured by the userś hand. As a solution we have implemented a semi-transparent beam from the hand downwards onto the ground-plane and placed bricks. A future solution could be to make the virtual hand itself semi-transparent.

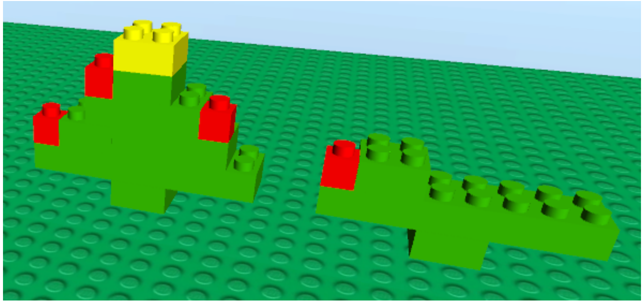

Figure 5: Evaluation scenario

Manipulation

- During the evaluation, a lot of unwanted accidental bricks were placed. The placing of extra bricks is due to the misinterpretation of gestures. This issue mainly occurred when selecting an existing brick, rotating the camera and going into menu mode. The misinterpretations are partly due to the Leap Motion itself, as it is not always able to accurately capture the hands. The Leap Motion website even states as one of the known issues that tracking quality is lower when making a fist or pointing (https://developer.leapmotion.com/features/faq). Secondly, we are using the same gesture for placing and selecting bricks. Having more distinguishable gestures, especially for placing new bricks and selecting old ones, would help to improve this problem.

Limitations

- After a short period of controlling the Lego Builder with hand gestures, the test subjects reported that it became exhausting. This is inherent to the use in-air gestures [4]. In our application, the user performs bimanual asymmetric and cooperative gestures. These types of gestures are usually faster and preferred by users [4] and therefore prohibit us to make alternative unimanual gestures available for both hands so the user could switch hand when the first one is tired.

- For the use of bimanual gestures we had to fixate the camera to either the stack or line of bricks the user is placing. Before this restriction to the camera movement, changes in center of hand caused the camera to move away from what the user was building. In order to increase speed of performance and ease of use for the bimanual manipulations, compromises had to be made to the 3D travel.

- There are some physical limitations to the Leap Motion. It has a very small range and it is easy to go out of range, especially when your hands are close to the device. Furthermore, it is limited in its accuracy. Although we were amazed by how well the device could track gestures, we also had to deal with a lot of false positives.
- It is currently only possible to stack bricks up to a maximum of 10, without using the special stacking gesture. This limitation is caused by the restriction in camera movement, the camera does not move upwards.
- Walls can only be generated towards the camera. We have chosen to keep this limitation, as it ensures that you can always see what you are building. In addition, the user can rotate the camera 180 degrees around the y-axis to build a wall in the opposite direction.
- The gesture for rotating a brick does not provide nulling correspondence, that is, rotating the same amount in the opposite direction does not undo the rotation [4]. This issue is solved by the undo gesture also taking into account the orientation of the brick.

Positive Remarks

- According to the test subjects, the gestures for navigation and placing bricks felt natural and were easy to learn.
- The combination of manipulation and navigation was successful, as the volunteers could easily move and place a brick with only one hand.
- The gestures for placing walls in horizontal and vertical lines prevented the user from making mistakes.

4.2 Discarded Techniques

Multiple techniques were created to support the user tasks, but some of them did not make it into the final prototype. This section covers those techniques and explains why they were discarded.

Multiple gestures

Instead of combining gestures, another possible solution to enhance the speed of performance is to support the recognition of multiple gestures at the same time. The user could for example navigate with the non-dominant hand and while manipulate bricks with the dominant hand. Our prototype initially supported this technique, but during our first tests we discovered an important limitation. When two hands are moving, they can collide with each other in the physical space, preventing the user from continuing with the virtual navigation and/or movement.

Short pinch

Selecting an existing brick or spawning a new one are two user tasks that are currently employing a continuous pinch gesture. An alternative approach could be to initiate the task with a short grasp or pinch, after which the Lego Builder assumes the user is still grabbing the brick. Next, the user can drop the brick by grasping or pinching again. This could lessen the hand fatigue test users experienced, as muscles would have to be contracted for a shorter amount of time.

Camera movement

Our initial prototype moved the camera when the hand was away from the center point, and would move faster the further away the hand was from that point. This resulted in very sensitive and twitchy navigation. We opted to only move the camera when holding the hand near the border of the Leap Motion's range and accelerate over time. This solution takes into account that 3D manipulation is more important than 3D navigation [4].

4.3 Unresolved Challenges

There are many more challenges that could be thought about. Those challenges are covered in this section.

Select multiple bricks

In order to increase speed, users should be able to select and manipulate multiple bricks simultaneously. This would make it easier to build, move or delete large amounts of bricks.

Placing new bricks underneath other bricks

By solving this challenge we would for example have been able to build underneath a roof. In our final prototype, users can only construct a building line by line. In other words, it is impossible to build a table if a ceiling has been constructed.

Navigating in occluded spaces

This challenge involves being able to look through a wall in confined places and place bricks. Navigating inside a house can be a good example where you would be able to place bricks below the ceiling.

Different orientations of the bricks

In the current prototype, the studs of the bricks are placed upwards. In the future it could be considered to positioning bricks with the studs facing other directions. This would allow users to build more complex models.

Freedom for the camera's orientation

The user has to be able to change the perspective with smaller increments than the currently used 90 degrees. In addition, a different gesture could be implemented to increase the user's control over the rotation around other axes.

4.4 Final mapping

Keeping the evaluation from the previous section in mind, the final mapping of hand gestures is listed in Table 1.

Gesture	Action
Move hand the borders of the Leap Motion range	Navigate through the environment
Swipe left or right	Rotate camera 90 degrees along y-axis
Grab in empty space	Spawn new brick
Grab near existing brick	Select brick
Open fist to flat hand	Place brick
Point at menu with index finger	Open menu
Point at menu item and move out of the menu	Select menu item
Shake flat dominant hand horizontally	Undo
Multiple grabbing gestures with dominant hand while non-dominant hand is open horizontally	Place bricks in a line
Multiple grabbing gestures with dominant hand while non-dominant hand is open vertically	Stack bricks vertically

Table 1: gestures and their corresponding actions

5. CONCLUSION

The Lego Builder is a project that tried to discover the properties, capabilities and limitations of 3D user interfaces that make use of the Leap Motion as a primary input device. There were many challenges with the selection and manipulation of the virtual bricks and with the navigation throughout the virtual environment. Not all challenges were solved but the ones that were covered provide a good indication of what is possible with the Leap Motion.

Gestures must be carefully chosen in order to prevent ambiguity that can cause the system to make mistakes. This ambiguity can partly be solved by the implementation. The effectiveness of an application depends on the details of the implementation [4]. Training the user to better gesticulate will also lower the amount of mistakes. On the other hand it is important to limit the set of gestures the user has to learn and remember. When more and more features are added to the interface, creating new distinct gestures becomes increasingly difficult and learning to use the interface becomes defiant. Therefore it is important to keep a balance between the amount of features and the usability of the interface.

In the case of the Lego Builder, there is still room to implement some more features and corresponding gestures. However, it is clear that implementing many more features will decrease the usability, because the system will start to make more mistakes or because the user has to remember to many distinct gestures. Therefore, any future work should carefully examine opportunities that will keep the gesture count as low as possible to provide a well performing and easy to use interface.

6. REFERENCES

[1] Leap motion. `https://www.leapmotion.com/`. [Accessed on the 18 October 2014].

[2] Microsoft kinect. `https://www.microsoft.com/en-us/kinectforwindows/`. [Accessed on the 18 October 2014].

[3] Unity game engine. `http://unity3d.com/`. [Accessed on the 21 November 2014].

[4] D. A. Bowman, E. Kruijff, J. J. LaViola Jr, and I. Poupyrev. *3D user interfaces: theory and practice.* Addison-Wesley, 2004.