Department of Electrical & Computer Engineering
Part IV Research Project Report
2015

Jenga Simulator: Manipulation of Objects in 3D using Tangibles

Project 15 – 3D-2D-3D Interaction

Author: Simon du Preez
Project Partner: Ben Brown
Supervisor: Dr. Beryl Plimmer

DECLARATION OF ORIGINALITY

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Name: Simon du Preez

Jenga Simulator: Manipulation of Objects in 3D using Tangibles

Simon du Preez,
Department of Electrical and Computer Engineering
University of Auckland, Auckland, New Zealand

***Abstract*—Users find it hard to manipulate 3D objects with touch gestures in a virtual world. There are no standard touch gestures for the manipulation of these objects. Widgets and modes are often used to manipulate a 3D space which can easily lead to unintuitive mapping of controls. The Jenga game implemented provided a well-known context in order to test out a new gesture set developed. Both touch and tangible input was used. Tangible objects allowed users to physically interact with the game and as a result experience higher usability in the 3D environment.**

I. Introduction

THE global reach of multi-touch displays and devices continues to grow. Table sized multi-touch displays are appearing with greater frequency in places such as museums. They encourage collaboration, learning and design across a wide range of applications [1]. The use of tangibles, in addition to touch gestures, presents a novel way to interact with multi-touch displays. This enables an intuitive mapping between real objects and their digital counterparts [1].

Typically when interacting with a 3D world, especially in 3D modelling, users will be using a mouse and keyboard. Users find it hard to manipulate 3D virtual objects with touch gestures. Complex widgets and modes are often used to manipulate a 3D space which can easily lead to unintuitive mapping of controls. In addition this can require a lot of system knowledge and training. Previous work has looked into ways to making these widgets easier to use, however this is circumventing the overall problem [2].

Attempts to mitigate complexity of 3D manipulation have been explored through techniques such as palm menus, contextual gesture mappings and axis manipulation [3], [4]. There has been extensive research conducted into the touch based usage of devices, particularly in 2D digital space [3], [5], [6]. The focus of this research was to investigate intuitive ways to interact with objects in 3 dimensions. As well as this, this paper explores how tangibles can assist in this usability. In order to do this, a testing environment has been developed and implemented. The test environment was based on the game of Jenga. A well-known game which requires dexterity to manipulate blocks in all 3 dimensions.

II. Related Work

Ku and Chen [1] looked specifically into the rotation of a 3D model using a 2D touch surface. The focus was on intuitive interaction for large exhibit style displays such as kiosks. The study that was conducted involved both adults and children. There were three tasks involving the manipulation of a 3D turtle. They began with an easy rotation and ended with a more complex rotation. Two modes of rotation were used:

1) Single finger dragging
2) An extension of current 2D manipulation techniques (2 finger rotate)

Adults tended to rely more on previous experience and when that failed they followed the given instructions to the letter. Children tried the first thing that came to mind and then employed a guess and check method to explore the system. When the task became complicated performance of the rotations was unaffected by the technique used or the prior experience of the user. In addition the results found indicated standard 2D object rotation gestures, which use two touch points, were not appropriate for 3D environments.

Herrlich et al [5] looked into specific techniques for manipulation of 3D objects over a 2D surface as well. The techniques were designed for all 6 degrees of freedom (DOF). These are: up, down, left, right, in and out. Two techniques were compared against each other. The Turn&Roll technique and the PieRotate technique. Turn&Roll uses one finger rotation for horizontal and vertical rotation around the Y and X axis. For the Z axis it uses a two finger rotation. Translation is achieved via two finger panning and Z-Pinch for depth translation. PieRotate divides the screen into specific regions which correspond to a particular axis. A rotate gesture rotates the object depending on what portion of the screen the gesture is performed in. In addition a one or more finger pan is used for translation and a Z-Pinch for translation into and out of the screen. From user studies it was found that users preferred the Turn&Roll technique however if someone mastered the PieRotate technique it could be used more efficiently than Turn&Roll.

A Novel Multitouch Interface for 3D Object Manipulation [3] focused on 3D modelling using touch based interaction. One of the main goals was to achieve a widget-less design to reduce visual clutter on screen. In addition, scale and location independent gestures were sought after. The final design was achieved by using multi-touch gestures and a context based solution. If no object is selected then the application is in a global state where the user would interact with the world. Objects could then be selected with a tap and then manipulated using multi-touch and a selected axis. The axis is chosen based on the orientation of the gesture. In addition a palm menu was implemented which allowed for a five finger tap anywhere on the screen. This brought up a shortcut menu with 5 items. Lastly snapping aided with the imprecise nature of touch movements.

Other areas of research have focused on the tangible aspect of input. The THAW research [7] used mobile phones in order to act as a physical interface for interactions on a computer screen. This used the inbuilt camera on the smart phone to detect what was happening on the screen. As well as this, Lumino blocks have been developed which allow for tangibles to stack on top of each other [8]. The tangibles unique tag changes depending on how many blocks are stacked on top of each other. In this way, a physical Z dimension can be added to the table.

III. Context

The aim of this project was to produce a set of intuitive and generalizable gestures that allows users to interact with objects in a 3D environment. The use of both touch gestures and tangible objects was explored. In order to investigate intuitive gestures, a context was needed. Jenga is a well-known game which needs physical skill and dexterity in order to remove blocks from the tower. This involves the translation and rotation of blocks all in 3 dimensions. Therefore it is a suitable platform to test the gestures.

IV. User Study

An observational user study was conducted in order to get a preliminary idea on what gestures people find intuitive. The first gesture a user tries is generally the most intuitive to that person.

Design

The study was designed based on the Wizard of Oz experiment. This is where participants interact with a system where the response of the system is being simulated by a human. This allows the system to be tested and evaluated without actually being implemented.

Setup

The adaptation of the Wizard of Oz experiment presented in this paper was conducted with a physical game of Jenga that the users could not directly interact with. The physical Jenga tower was placed underneath a transparent Perspex screen. The Perspex screen, in this manner, would act as a touch screen. The users would be given a task and in order to complete the task would make gestures on the screen that correlate to a particular action. The action would then be performed by the "Wizard" who moves the block in the way the participant expects. Each session was recorded with multiple cameras in order to review gestures that may have been missed during the study. A set of objects was given to the participants in order to use as tangibles. They were able to use these tangibles as they saw fit. Two questionnaires were given to the participants. One before the study for background information and one after for the users to evaluate their gestures. There were four male and three female participants in the study. They were all experienced users of touch devices.

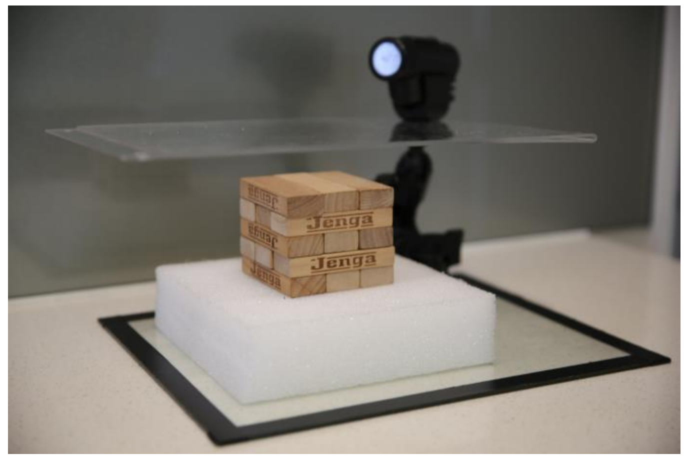

Fig. 1. Wizard of Oz apparatus showing Jenga Stack, Perspex screen, rotatable platform (foam base) and camera.

Procedure

The study was split into three different sections:

1) Physical manipulation of the blocks.
2) Manipulation of blocks via touch gestures performed on the Perspex screen.
3) Manipulation of blocks via touch and tangible gestures performed on the Perspex screen.

The first section was performed first by all participants. The second and third sections were presented in different orders to mitigate order-effect bias. Each section had tasks for translation and rotation of the blocks from different viewpoints. The full set of tasks ranged in complexity from simple translation to a full retrieval and replacement of a block. Each study took around 45-60 minutes to complete. Participants were told to explain their movements and state their assumptions as they were performing the tasks.

Results

For simple tasks users all performed very similar basic gestures both with and without the tangibles. Some had minor differences but they all adhered to the same theme. For basic translation of a single block, participants all used single finger swipe. Rotation of the stack and individual blocks was generally performed with well-established finger rotation gestures. These are the gestures you commonly use in 2D across many applications. Only specific details changed. This included the number of finger used (one to four) and whether they used equidistant, around the thumb or two-handed rotation.

For the more complex actions, users chose more specialized actions. The gestures tended to have a high degree of variance. The specific task of moving a block towards or away from the Perspex screen yielded very different solutions. Over half the participants on average either couldn't complete this class of task or used a gesture which conflicted with a previous gesture. This shows that there is a certain degree of intuition involved in manipulating a 3D environment for simple tasks. However intuition didn't appear to help with the complex tasks.

Many participants assumed movement of the blocks would snap to where they wanted it to be rather than a continuous motion. The two most commonly used tangibles, were the block tangibles and the stack tangibles.

V. Implementation

A. Microsoft PixelSense

This project targeted the Microsoft PixelSense table (specifically the Samsung SUR40 [9]) to develop the gesture set. The table is basically a large touch screen device which, in addition to fingers, can recognize objects placed on its surface. This is achieved by using infrared vision technology in contrast to the more common capacitive displays seen today. The Surface platform itself processes and recognizes three types of objects in contact with the screen: fingers, blobs and tags. In addition to this developers can process the raw image data themselves.

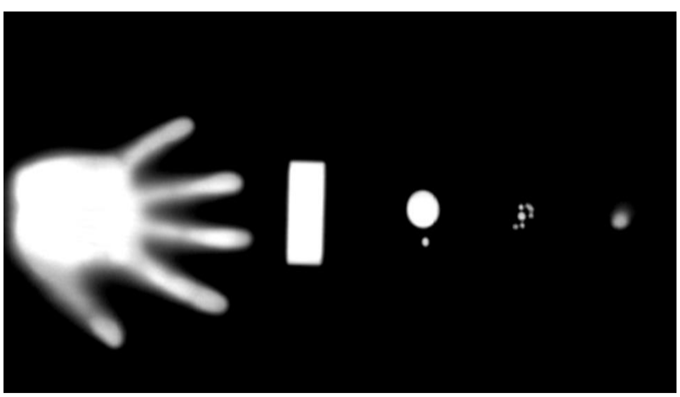

Fig. 2. Raw data shown from the RawImage Visualizer.
From left to right: Human hand, Jenga Block, Blob Pair, Byte tag and a finger.

B. Development Environment

To implement the game of Jenga a game development environment was needed. The first choice was to Unity. However Unity does not integrate well with the surface SDK. As a result it makes reading touch input from the surface very difficult. A more native approach is to use XNA game studio 4.0 which is what was used for the final implementation. The surface SDK includes many tools which were invaluable throughout the course of the project. These were the Input Simulator, Input Visualizer and the RawImage Visualizer.

C. Surface API

There are two different sets of API in the Microsoft Surface SDK. The core layer and the presentation layer. The presentation layer uses the Microsoft Windows Presentation Foundation (WPF) 4.0 which includes many standard UI elements, including buttons, labels and scroll bars. This is the standard API for most touch applications allowing rapid UI development. The core layer doesn't depend on a particular UI framework and so has fewer in-built controls. However because of this it is much more flexible in how you can develop and more suited towards applications which require 3D graphics or raw image data. [10] Because of the 3D nature of the project and the use of XNA, the core layer was the API used.

D. Physics

Initially the system used the JigLibX physics engine. This is one of the earliest 3D physics engine developed for XNA and as a result has been used in many applications. As development of the Jenga environment progressed a more suitable engine was found. The Henge3D engine allowed for much easier customizability for the Jenga application and so was used in the final implementation.

E. Tangible Recognition

As previously mentioned, the PixelSense can detect objects placed on the table with 3 kinds of input: blobs, fingers and tags in addition to the raw image data. Both blobs and tags can be used to track tangibles. Using the inbuilt blob or tag recognition is far more computationally efficient than using the raw image data. Specialized code is used by the vision system in order to track the tags and blobs. Blobs are recognized as the sharp edge of an ellipse. Tags show a unique pattern of dots. There are two kinds of tags: identity tags and byte tags. Tags use 8 bits of data and so there can be 256 uniquely tagged objects. Identity tags use 128 bits of data and so a much higher number of objects can be tagged.

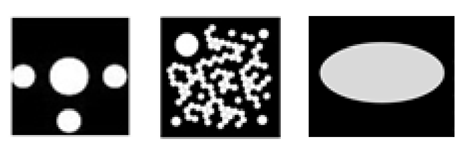

Fig. 3. From left to right: Byte tag, Identity tag and a blob

Initially tangible recognition was implemented using byte tags. The tags worked as expected in the case where physical position and orientation is not needed. However the inbuilt tag recognition was found to be extremely unstable for tracking of objects. Moving a tangible across the screen would result in inconsistent recognition. This caused multiple touchpoint ID's to be assigned to the same tag since from the tables perspective, the tag was continuously being lifted and put down on the table. This effect is illustrated in figure 5. Multiple calibrations of the table and tracking in an optimal environment (room without any lighting/no interference) had little effect to improve this problem.

Instead, tracking of objects was implemented using a different technique. This was based off prior research which used blobs in order to track objects. Blobs contain four pieces of relevant information. A bounding box, a major axis length, a minor axis length and a center position. To gather more information, processing of the raw image would have to be done. The new technique uses two blobs in a pair configuration to track position and orientation of an object. A blob pair consists of a small blob and a large blob. By drawing a line between the centers of these two blobs, the orientation of the tangible is able to be determined.

Implementation for the blob pairs consisted of two parts. Pairing of the blobs and recognition of a tangible. Initially, to decide which blobs are paired, threshold values were given for the distance between the blob centers. If the blobs were under the maximum distance threshold they were paired, otherwise not. However this was found to be to be limiting and specific to the blob pairs used. The final algorithm was based off the assumption that no small blob is of bigger size than any other big blob. Also the distance between the two blobs for a blob pair is the smallest distance compared with the other blobs.

Once the blobs were sorted into pairs, three distinguishing traits of the blob pairs were checked against previously registered tangibles in order to determine what object has actually been placed on the screen. The three distinguishing traits of the blob pairs are:

1) The width of the small blob
2) The width of the large blob
3) Distance between the two blobs

It was found that these three values were imprecise. Readings from the table had high variance which appeared to align with orientation of the table. Therefore a probabilistic algorithm was developed in order to determine the correct tangible object placed on the screen. A blob pair is assigned a closeness value to each registered tangible. The tangible with the highest value selected if this value is above the minimum threshold of 70%. This is shown in figure 4. The readings which varied the most were the width of the blobs and so a higher weighting was assigned to the distance trait for the blob pairs.

```
---------------BEGIN--------------------
Fine Camera : 0.9061756%
Jenga Stack : 0.6513986%
Jenga Block : 0.3070121%
Cork Screw : 0.195727%
---------------Final--------------------
Tangible is: Fine Camera (90.61756%)
---------------END----------------------
```

Fig. 4. Probabilities assigned to each tangible. The one with the highest is selected as the object placed on the table.

Using this technique yielded far better tracking which mitigated the hardware limitations. Figure 6 shows an identical swipe gesture performed with a blob pair compared with a tag in figure 5. However this implementation was still not perfect. Blob recognition relies on the sharp edges of the blobs. Finger points look nearly exactly like a blob except with blurred edges. This allows the table to distinguish between the two types of inputs. However, holding an object with a blob marker a few millimeters above the table causes the edges of the blob tag to become blurred. In this instant the table registers a finger. This event occurs every time a blob is placed on the table and had to be developed around. This is illustrated in figure 7. No optimal solution to this problem was found and was treated as a hardware limitation. A very short delay was incorporated into the system to wait 50 milliseconds. If the touchpoint was still a finger point after this delay, the system would proceed as normal. This delay had no effect on end usability.

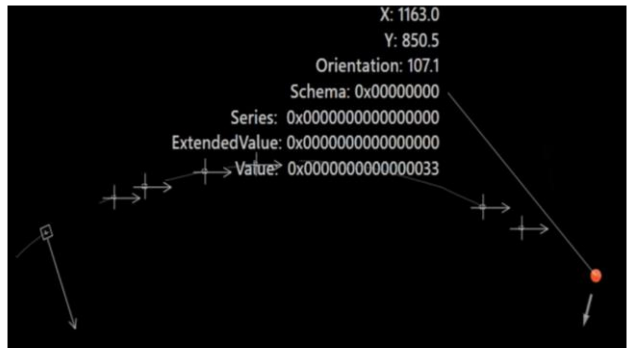

Fig. 5. A swipe gesture using in-built Byte tag recognition. Note: line is broken and inconsistent

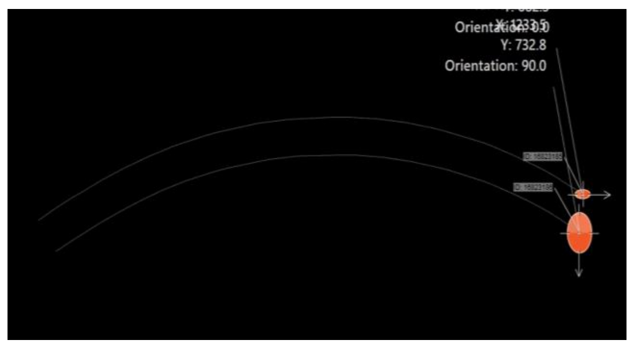

Fig. 6. A swipe gestures using implementation of blob pairs. Note: line is consistent providing better tracking than Byte tags

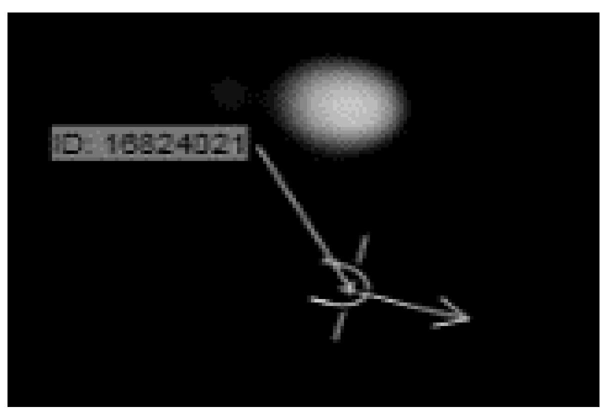

Fig. 7. A tangible with a blob pair is held millimeters off the surface. The surface recognizes a finger touch point.

F. Development of Gesture Sets

Originally the system was built in such a way that the gestures could not be swapped out easily.
As implementation progressed it became apparent a better system was needed. Instead support for pluggable gesture sets developed. A gesture set in this way is a self-contained unit which can be changed easily. This allowed multiple gesture sets to be tested simultaneously. This was particularly useful for showing to users to get their preference for particular gestures.
There are two different types of gestures that were focused on. The first was movement of the camera in 3D space around the Jenga stack. The second, was the manipulation of a single block in order to play the game.

G. Camera Manipulation

In an actual game of Jenga you typically walk around the Jenga tower in order to see what's on the other side. However this isn't possible on a touch device. On a table this is technically possible but would require tracking of the person around the table. Therefore gestures for the manipulation of the camera were developed.

Touch

Initially a free flying camera was used. This allowed users to move forwards and backwards as well as move the camera up, down, left and right. However that was inconsistent with the camera that was simulated in the user study. In addition the amount of control it gave the users was deemed unnecessary given the context of Jenga. Users would get caught up in the movement of the camera and as a result had a lower level of immersion in the game.
The next iteration produced a different system which used slider bars. Many participants in the Wizard of Oz user study used this or a similar technique. The camera would focus on the Jenga tower and rotate around it when using the slider bars. In this way the target of the camera was fixed. The slider bars were found to be a step in the right direction but it still caused a disconnect between the users and the game. Users would have to look away from the stack, then use the slider bars and then look back to the stack again. This tended to take the user out of the game mentally. In addition the slider bars were in contrast to a goal of the project which was to use as few widgets as possible.
The final iteration allowed users to rotate the camera using a swipe gesture. Camera manipulation can be done anywhere on the screen aside from on a selected block and with any number of fingers. This was allowed by the context based selection in the final gesture set.

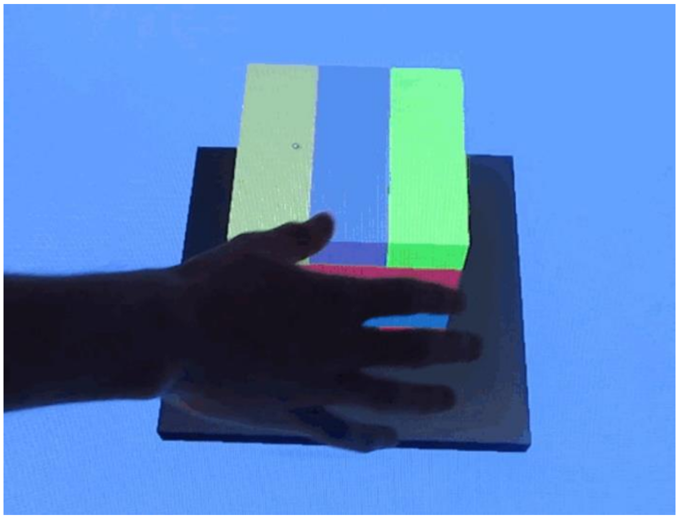

Fig. 8. Five finger rotation of the camera.

Tangibles

Two main tangibles for movement around the world were implemented. The first is the stack tangible. This tangible was used in the user study by many of the participants. Placing the stack tangible on the table in a particular orientation snaps the view to one of the five faces. This is the only tangible which used the inbuilt byte tags for recognition. As mentioned earlier byte tags are useful in the case where position and orientation is not needed. For the stack tangible all the information needed is that the tangible is on the table. There were four byte tags placed on the side faces of the tangible. One blob-pair tag was placed on the bottom of the stack tangible which allowed finer grain control from the birds-eye view of the Jenga stack.
For finer gain control of the whole camera, a second tangible was implemented. Spinning the fine grain tangible rotates the view around the stack. In addition, moving the tangible up, down, left or right moves the camera view in the same way a finger swipe would with the same gesture. E.g. A swipe to the left rotates the camera in an anti-clockwise direction.

H. Block Manipulation

Touch

Initially translation was done with a touch and drag style movement in the first iteration of development. However it was found that a context based selection was needed to confirm user intent. Therefore the selection of a block via double tap was implemented. Double tap was selected since a number of participants used this particular action in the user study. A single tap was explored, however because of the number of false positive tap events that were produced by the table, double tap was used instead. This reduced the number of errors made. Once a block has been selected, it is able to be moved around using single finger drags. The block that is selected always moves in the plane that is perpendicular to the current view.

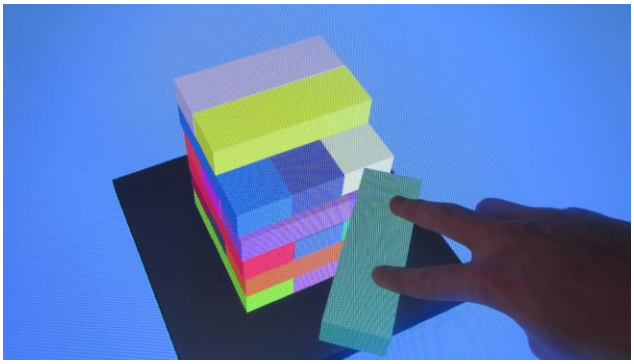

Fig. 9. Two finger rotation of a block.

Rotation of the blocks is based off well-established rotation gestures that are typically used on devices today. This was popular during the user studies and found to be most intuitive given how common it is. Initially rotation was done using only two fingers equidistant apart. But implementing block selection opened up many different possibilities. For the final iteration users were able to rotate a block using any number of fingers. In addition users do not have to be touching the block at all. A midpoint calculation for the points in contact with the screen is made. If the midpoint hits a selected block, the gesture is related to manipulation of a single block. In the other case where the midpoint is not over a selected block, the gesture performs camera manipulation. This is shown in figure 10 where rotation is being performed using midpoint calculation.

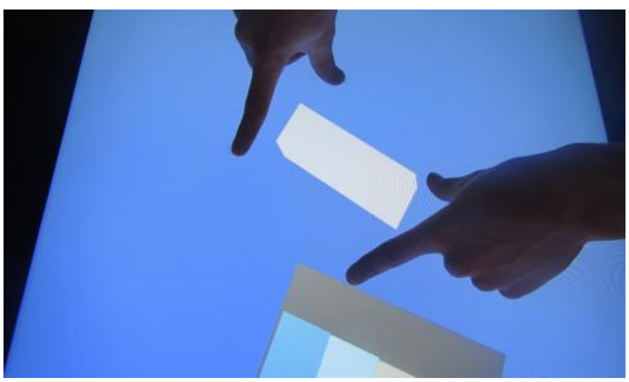

Fig. 10. Four finger rotation of a block using midpoint calculation.

The last action was the one which caused the most problems for participants in the user study. This was where a block was to be moved towards or away from the participant. This was implemented using a pinch to zoom style gesture to move the block. As with rotation this can be done with any number of fingers and without touching the block. The gesture is designed to mimic the grabbing of a block between two fingers and pulling it towards you.

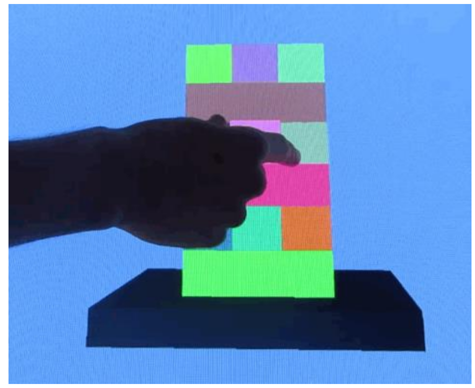

Fig. 11. Zoom gesture to move the block towards the screen.

Tangibles

In addition to these touch gestures, two main tangibles were developed which enabled all the discussed functionality. The first tangible is the Jenga block. Placing the block on a digital block will select it and then movements of the physical block are then mirrored on the digital block. One of the main obstacles with the Jenga block tangible was deciding how to snap the virtual Jenga block to the physical one. If the Jenga block snaps directly to the middle of the physical Jenga block, it causes problems when attempting to slowly remove a block from the stack. For example, if a user places the physical block with the midpoint of the physical block directly on the edge of a virtual block, this will cause the virtual block to snap half the length of the block to the center. This result of this is that the block leaps to that particular position often causing the tower to topple. Another solution to this problem is to implement the block tangible so that no snapping occurs. The problem with this is that the virtual Jenga block no longer aligns with the physical Jenga block. A final solution to this was that the virtual block would slowly track to the correct position over a few seconds. In this way it is almost undetectable by the user but by the time they remove a block from the stack the virtual block aligns perfectly with the physical block.

The second tangible for block manipulation was the corkscrew tangible. This enables the movement of the block towards or away from the screen. Spinning the corkscrew tangible to the right moves the block moves into the screen and to the left moves the block out of the screen. The Z movement of the block is in the direction of the camera from the selected block.

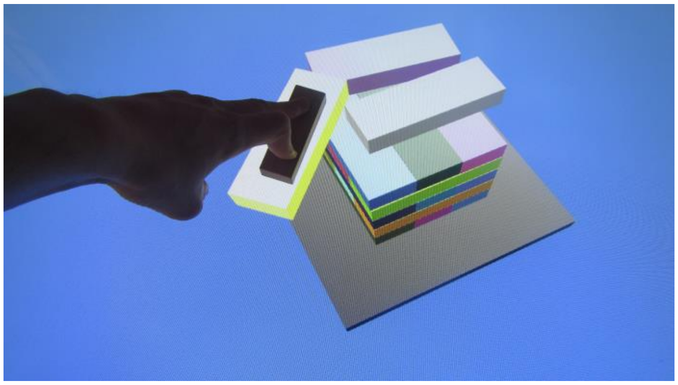

Fig. 12. Using the Block tangible in order to manipulate the blocks.

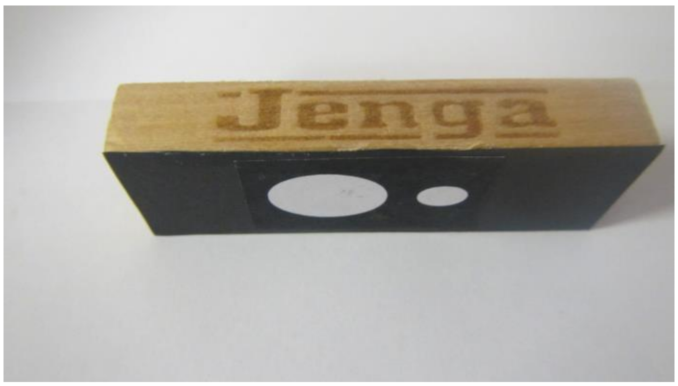

Fig. 13. Jenga block tangible with blob pair on the bottom.

VI. Evaluation

A final usability test with several participants was performed in order to test the viability of the final gesture set. Participants were shown the major iterations of development and were asked to use the system in the way they would normally play the game. Users were given a short two minute brief on the gestures and tangibles the system supports.

Overall the final gesture set performed to a satisfactory standard for all participants. Users quickly picked up on the all of actions and were all able to complete multiple turns and play the game. Certain gestures were used noticeably more than others and participants commented on the usability of most gestures whether it be good or bad.

Firstly for camera manipulation, participants used the touch gestures as well as the fine grain tangible to move the camera the most. These were used in approximately equal amounts. Some participants commented they had a preference for the use of the tangible over the touch gestures. All users gave the stack tangible a try, but never used it more than the first time opting for one of the other two methods.

For block manipulation users tended to use finger gestures more than the tangibles for all but one of the actions. For Z-translation participants favored the zoom tangible. The block tangible was used by some participants but overall touch was used more. When manipulating the blocks the users generally used an isometric view. In this scenario the block tangible doesn't align well with the blocks. This is a possible reason for the lesser use of the block tangible over touch gestures.

Overall tangibles had a large impact on the usability of the system. Participants noted particular actions were made easier with the addition of the tangibles while some were easier with touch gestures.

VII. Future Work

Throughout the project, Jenga has been a context to allow the development of a gesture set to manipulate objects in 3D. The final gesture set discussed in this paper should be tested for its generalizability. Currently it is uncertain how specific the gestures are to Jenga and how useful they will be in different applications. Further research will be able to extend or modify this current gesture set in order to test its application in other areas.

In addition the blob pair technique can be built upon, made more robust and allow more unique tags. The current implementation discussed in this paper added a probabilistic approach in order to determine which tangible was placed on the screen. For this only the major axis, minor axis and distance was taken into consideration.

VIII. Conclusion

The Jenga simulator implemented had a purpose to explore gestures that allow users to manipulate objects in 3 dimensions. In addition, the use of tangibles was studied as to how they can aid in the usability of such a task. The first step of the process was to conduct a user study to gain a preliminary idea on which gestures users find intuitive. The project faced many challenges with the largest being the limitations of the table and its inconsistent input. The results of the investigation have indicated that tangibles can provide a higher degree of usability depending on the particular tangible and action. Most touch screens like phones and tablets can't recognize tangibles and wouldn't be practical even if they could. The application of tangibles is best suited for environments such as educational contexts where large tables can be on display. For example, kiosks and museums. This research extends our current knowledge in the domain of 3D manipulation, however continued research is required in the area in order to come up with a truly standard and generalizable set of gestures.

Acknowledgment

First and foremost the author would like to thank Dr. Beryl Plimmer, who was the main supervisor for this project and provided much needed guidance throughout the whole project. Without this guidance, the project would not have been a success. In addition the author would like to thank the HCI research group at the University of Auckland for their continued feedback. Finally, a thanks to Ben Brown who was the project partner for this project.

References

[1] Bertrand Schneider, Patrick Jermann, Guillaume Zufferey, Pierre Dillenbourg. Benefits of a Tangible Interface for Collaborative Learning

and Interaction. Retrieved from: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5654494

[2] Oscar Kin-Chung Au, Chiew-Lan Tai, Hongbo Fu. Multitouch Gestures for Constrained Transformation of 3D Objects. Retrieved from: http://dl.acm.org/citation.cfm?id=2322125

[3] Oscar Kin-Chung Au, Chiew-Lan Tai. A Novel Multitouch Interface for 3D Object Manipulatio. Retrieved from: http://visgraph.cse.ust.hk/projects/multitouch/Multitouch-3D-Manipulation.pdf

[4] Jingbo Liu, Oscar Kin-Chung Au, Hongbo Fu, Chiew-Lan Tai. Two-Finger Gestures for 6DOF Manipulation of 3D Objects. Retrieved from: http://visgraph.cse.ust.hk/projects/two-fingers/TwoFingerGestures/Two-Finger%20Gestures%20for%206DOF%20Manipulation%20of%203D%20Objects.pdf

[5] Marc Herrlich, Walther-Franks, Rainer Malaka. Integrated Rotation and Translation for 3D Manipulation on Multi-Touch Interactive Surfaces. Retrieved from: http://link.springer.com/chapter/10.1007%2F978-3-642-22571-0_16#page-1

[6] Caho-Jen KU, Li-Chieh CHEN. A Study on the Natural Manipulation of Multi-touch Gestures for 3D Object Rotation Using a Large Touch Screen. Retrieved from: http://ebooks.iospress.nl/volumearticle/36476

[7] Sang-won Leigh, Philipp Schoessler, Felix Heibeck, Pattie Maes, THAW: Tangible Interaction with See-Through Augmentation for Smartphones on Computer Screens Hiroshi Ishii. Retreived from: http://tmg-trackr.media.mit.edu/publishedmedia/Papers/545-THAW%20Tangible%20Interaction%20with/Published/PDF

[8] Patrick Baudisch, Torsten Becker, Frederik Rudeck. Lumino: Tangible Blocks for Tabletop Computers Based on Glass Fiber Bundles. Retrieved from: http://www.patrickbaudisch.com/publications/2010-Baudisch-CHI10-Lumino.pdf

[9] Samsung. Samsung sur40. http://www.samsung.com/uk/business/business-products/smart-signage/specialised-display/LH40SFWTGC/EN EN, September 2015. [Accessed: 12-9-15].

[10] Microsoft. Presentation Layer vs. Core Layer Applications. https://msdn.microsoft.com/en-us/library/ff727793.aspx, September 2015. [Accessed: 12-9-2015].