JengAR - The Classic Game in Augmented Reality

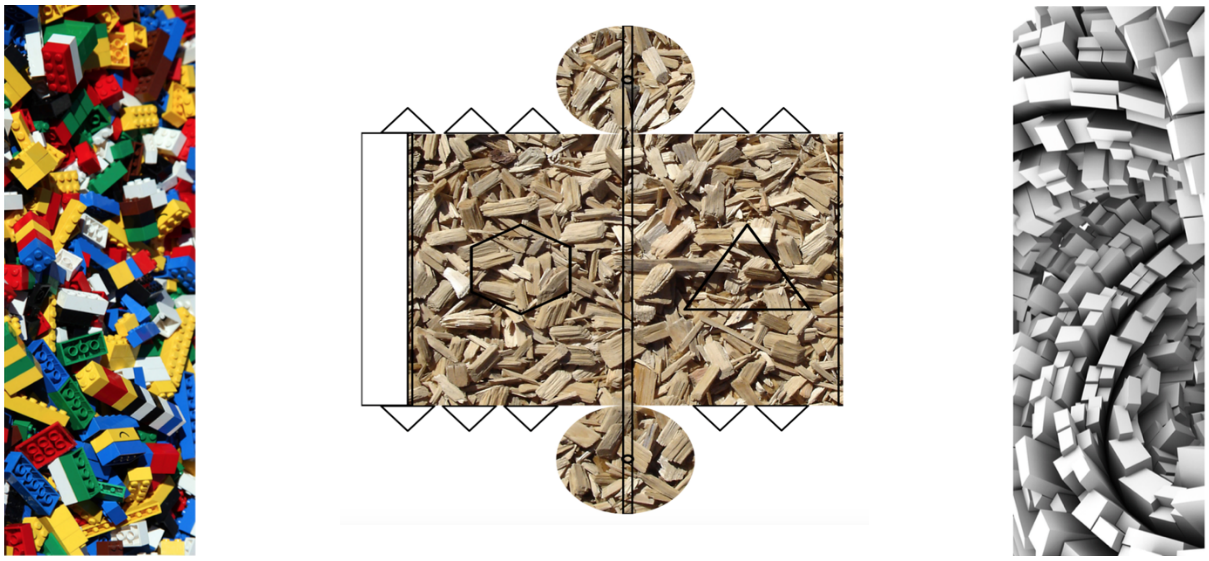
Akshay Nagpal, Navraj (Navie) Narula, Jake Petterson, Tiancheng (Harry) Zhang

Introduction

JengAR is a one-player, augmented reality game in which users can interact with a pre-existing Jenga tower containing 48 blocks. Like the real life Jenga game, users are able to select a block from the base of the tower and place it on the top of the tower. The user's goal is to avoid collision and collapse of the tower. If the block placement causes an imbalance of the Jenga structure, the program will detect this fault, resulting in the tower toppling over, and terminating the game. JengAR is designed to mimic real Jenga as closely as possible, just with different control mechanisms.

Image Targets

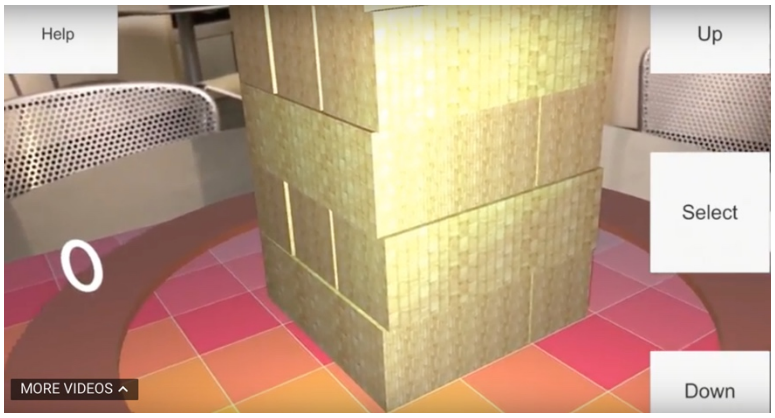
The image targets we have employed in our program include an "anchor," a wand, and a minimap. The anchor is a cylinder target to which the tower is attached, which recognizes the woodchip pattern. The wand target, which recognizes the lego pattern, acts as a selection tool for the user. The minimap we have employed recognizes the vortex pattern. Below are the visuals for each image target, borrowed from the Vuforia image target library.



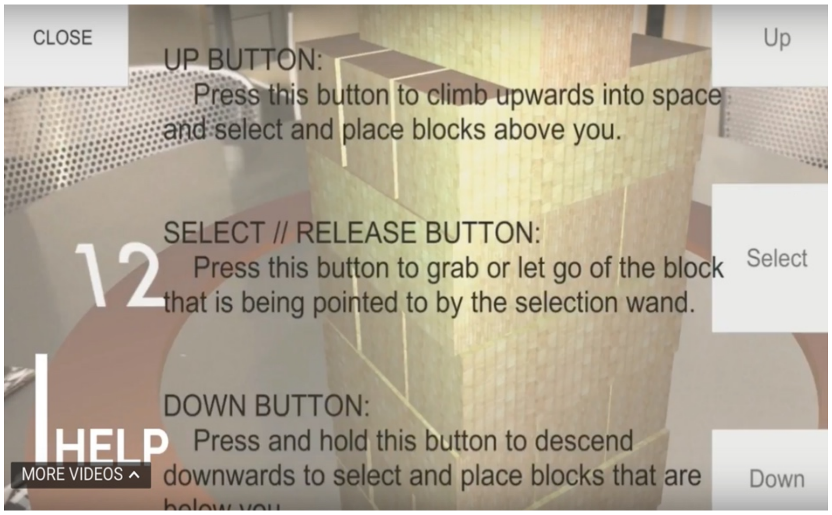
Left to right: Wand (lego pattern), Jenga tower (cylinder Target, woodchip pattern), and minimap (vortex pattern)

Layout and Initial Setup

When users first open up our Unity application, they will see the initial screen below:



Since the Jenga tower is the main object the user will interact with, it is placed in the middle of the screen. A hover ring surrounds the tower, which will allow the user to travel between floors. A number is visible on the hover ring, which indicates which level the user is at each time. The initial number the user will see is 0. To the right of the screen, there are three buttons displayed: up, select, and down. To the left of the screen, a help button is displayed. All buttons will be visible on the screen at all times. The select button will eventually switch to a release button when a block is selected. When the help button is clicked, the following screen is displayed below:



UP BUTTON:
Press this button to climb upwards into space and select and place blocks above you.

SELECT // RELEASE BUTTON:
Press this button to grab or let go of the block that is being pointed to by the selection wand.

DOWN BUTTON:
Press and hold this button to descend downwards to select and place blocks that are

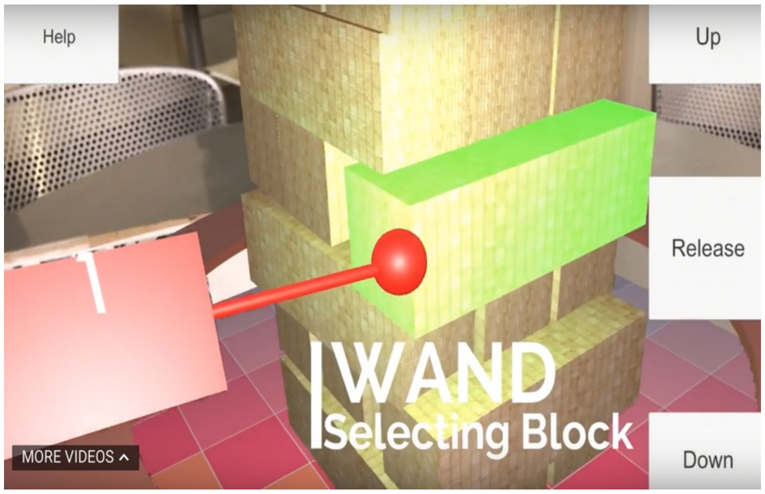
At the start of the game, the blocks are disabled. This is so that they don’t fall into space when the cylinder image target has not been seen, and its child, the floor has not yet appeared to catch the blocks. When users start each game, they should be sure that the Jenga tower is recognized

<u>first before introducing the wand to the scene. If this is done, the tower will not topple over at the start of the game.</u>

Design Choices

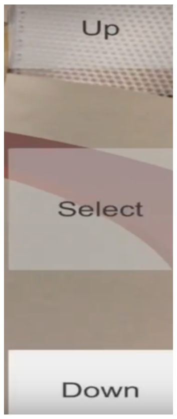
Selection

The user selects a block by means of forking. At this point, the block is attached to the wand and is released once the user places it on top of the tower. Once a block is selected, its material is changed to a bright, green color. This is done by transforming the RGB values associated with the material, and reverting them back to its original color once a block is released. Below, you can see what the block looks like once selected by the wand:



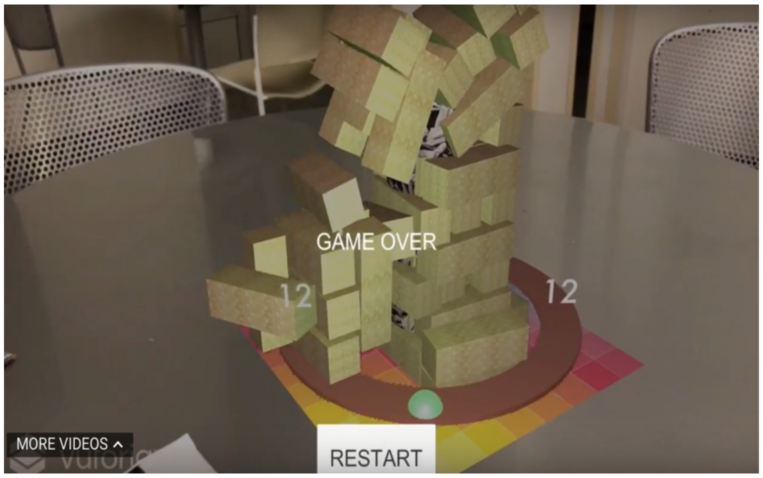
The color is a signal to the user, providing immediate feedback that a block has been selected.

Aside from selecting blocks, users can also select buttons on the screen. The help button on the top-right corner of the screen can be selected at all times. The up, select/release, and down button on the right side of the screen can only be selected one a time. Once one of these buttons is selected, the other buttons are "disabled," or rather grayed out. To the right is a screenshot of what the buttons look like when the down button is selected. The design is meant to help users recognize the mechanics of the game in the sense that multiple operations cannot be performed at once.



Collision

The wand itself also acts a trigger. A collision takes place when the wand interacts with the trigger on the box collider, causing it to fall. There are four planes that exist just slightly above the ground. When a block falls on any one of the planes, the game is over. The user is informed of this by a message displayed on the screen, stating: “Game Over.” The user may also restart the game by clicking on the “Restart” button, displayed in our screenshot below.

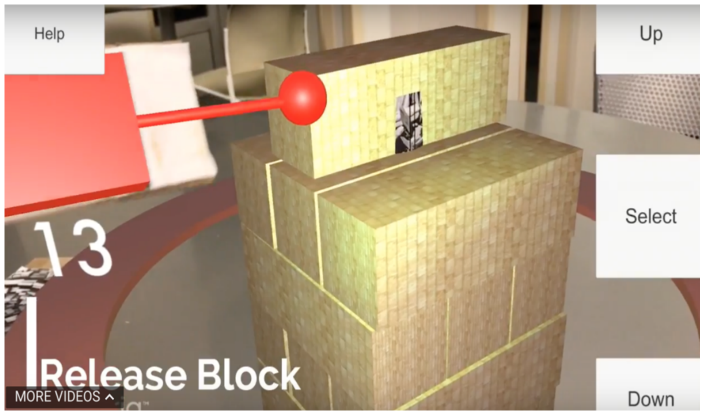


Friction

On each block, a physics material was added that has static and dynamic friction. This has the effect of making the tower more stable and making the game more playable than if there weren’t as much friction between the blocks.

Travel

The Jenga tower in its entirety is not visible to the user, who can only view three to five levels of a tower at a time. In order to travel between floors, the user presses and holds the up and down buttons that allow a hover ring to move up and down the tower; the user can also “walk around” the tower by walking around the cylinder target holding the camera. Limiting travel this way is fitting for our game since users will only be traversing a tower. Below is a screenshot following the travel process to the top of the tower, where a user can place a block.

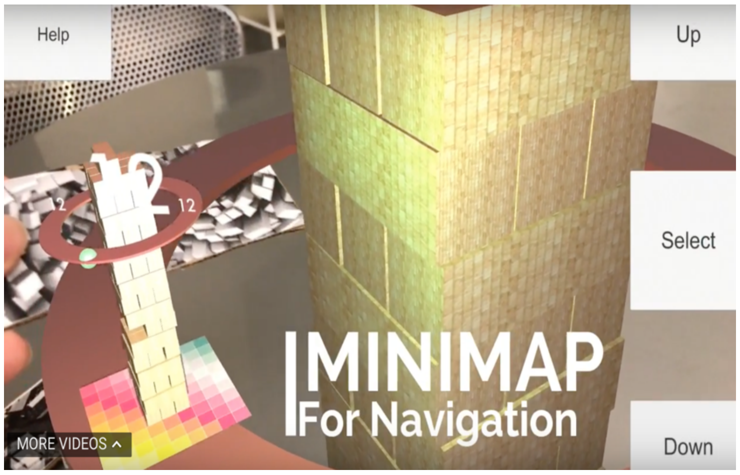


The upward and downward travelling accelerates from a relatively slow speed each time the user pressed down on the respective buttons. This design enables the user to move with precision when travelling short distances and move quickly when travelling long distances.

Wayfinding

When the user is on a certain level, the number of the level is displayed on-screen to the user to assist in wayfinding. For instance, if the user is on the fifth floor, they will see the number 5 on the hover ring.

We have also incorporated a mini-map into the program to assist with wayfinding. If a user is on a certain level, they may still be able to view the whole tower in its entirety via this mechanism. A green, "you are here," dot on the minimap allows the user to see where they are in respect to the whole tower. This minimap is visible in the screenshot below:



Heuristics

In terms of design, our system also does well to recognize nine out of ten of Jakob Neilsen's usability heuristics. The only one that we left off entirely is "help users recognize, diagnose, and recover from errors." See below for a short discussion of why we excluded this heuristic.

Heuristics	How We Satisfied Them
Visibility of the System Status	● Accomplished via a mini-map ○ Users can see where they are in regards to the whole tower despite the fact that they be on a certain level ● Button switching between select and release
Match between system and the real world	● Jenga tower collisions, mimicking the manner in which the actual Jenga game is played ● Viewing multiple levels at a time, which corresponds to real-life eye movement
User control and freedom	● Free ability to select any block from any part of the tower (i.e. besides the top) and place the block on the top of the tower
Consistency and standards	● Navigation scheme seems simple enough to avoid inconsistency ○ Labels on the button allow the user to not have to wonder if different buttons mean the same thing.
Error prevention	● In the case of the user's camera having difficulty recognizing the image target, we wouldn't want the blocks to be enabled without the floor (whose parent is the image target) already being there. We have a simple script that detects the existence of the floor and then enables the blocks. ● Handled the case where the user presses any two buttons at the same time.
Recognition rather than recall	● Buttons visible at all times, including a help button which gives the user the constant option to refresh their memory on how things work.

Flexibility and efficiency of use	● Button interaction and forking ● Use of minimal tools, assets, and image targets ● The up/down travel accelerates from a relatively slow speed in each continuous instance of travelling, giving user precision when travelling short distances and speed when travelling long distances.
Aesthetic and minimalist design	● The Jenga tower, which is the object the user interacts with at all times, is always front-and-center in the interface as long as the cylinder target is in camera view ○ No surrounding (i.e. distracting) objects exist, keeping the design minimal.
Help users recognize, diagnose, and recover from errors	● We made the choice that including an undo button would defeat the purpose of the game. It would make playing too easy and the game too trivial. Losing is possible! ● We also considered the idea of giving the user a couple "lives," but determined that this strayed from our goal of making JengAR as close in form and function to real life Jenga as possible.
Help and documentation	● A "Help" button exists on the top-left of the game screen at all times ○ When clicked, a short summary of the functionality of the Up, Down, and Select/Release buttons gets overlayed on the screen. (The game doesn't pause; the instructions appear in front.)

Requirements

Development platform: Unity 5.6 and Vuforia 6.2; Windows 8.1 and up or OS X (macOS) 10.8 and up. Deploying to iOS requires a compatible version of XCode.

Deployment target: Android (tested on a Nexus 6P and a Huawei Mate 9) or iOS (tested on an iPhone 7 Plus running iOS 10.3.1)

Video Demonstration

Link to video - https://www.youtube.com/watch?v=w4esGA0QbNI

Permission

Akshay Nagpal, Navie Narula, Jake Petterson, Tiancheng (Harry) Zhang are willing to have their names appear next to their presentation of their work on the project web page and video for COMS W4172.

Paragraph for Course Page

JengAR is a one-player, augmented reality game in which users can interact with a pre-existing Jenga tower containing 48 blocks. Like the real life Jenga game, users will be able to select a block from the base of the tower and place it on the top. The user's goal is to avoid collision when removing and placing blocks. The game can be restarted once the tower topples over.

Note: The representative screen shot, paragraph, and permission are included as separate files in the submitted folder.