6.804: Intuitive Physics in Jenga

Nisha Devasia (ndevasia@mit.edu)
Undergraduate
MIT EECS

Nada Hussein (nhussein@mit.edu)
Undergraduate
MIT EECS

Abstract

The game of Jenga is played such that players must take turns removing blocks from a tower without knocking it over. The game is played until someone knocks the tower over. WebPPL offers the option of a physics simulator within the language, such that a user can input some objects into a world with some dimensions, positions, and velocities, and then simulate the situation they have created. Using this tool, we decided to try to simulate Jenga-like scenarios, and compare the simulations with people's intuitions for the physics of a tower of blocks. Through this project, we discovered that people tended to give more conservative answers than the simulation, showing that they expected the towers to fall over more easily than they did. However, they were able to relatively accurately decide on the most optimal block to move, showing that they had strong intuition for where the most stable point in the tower was (relative to the point values of the blocks). This means their intuition for the tradeoff between point values and displacements was very accurate.

Keywords: intuitive physics; robotics; computational cognitive science

Methodology (Nada)

To conduct this experiment, we followed 3 basic steps. First, since WebPPL is unable to run simulations in 3D, we had to decide what our new version of Jenga would be like, and how it would be played. Second, we had to create instances of this game and collect data from people displaying what they believed to be the optimal move in the game, based on their own intuition of the physics of the game. Finally, we had to simulate these same instances and find the optimal solution to the game, and compare it to the human data to discover any trends in human intuitions about physics versus the actual results.

Gameplay

Our modification of Jenga is played as follows:

1. A tower is a 2-dimensional stack of blocks such that there is only one block per height. A visual of this tower can be seen in Figure 1.

2. Each block has a different point value assigned to it, such that the topmost block has a point value of 1, and each subsequent block has a score of 1+prevBlockScore. In this way, blocks on the bottom of the tower are weighted more heavily. In other words, the more dangerous a block is to move, the more points it is worth on our scale.

3. At a player's turn, they must maximize their score by moving one block as far as possible within the tower without knocking the tower over.

4. Score is calculated as blockValue*distanceMoved. In this way, the score considers both how much the player moves a block, and how 'risky' the block they chose was.

5. Gameplay is limited to moving one block per turn, and the game ends and the player gets a score of 0 if the tower is knocked over.

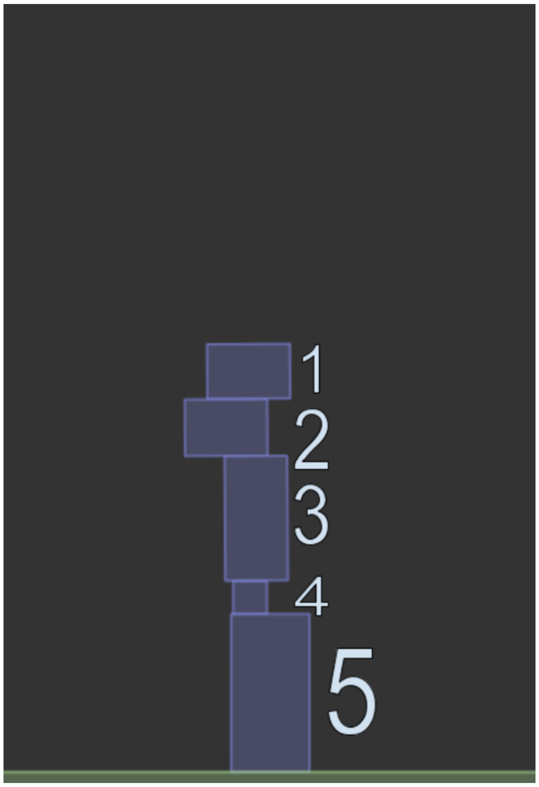

Figure 1: Here you can see an example of the towers used in this game. The numbers next to the blocks represent each block's point value in the game. Each tower was stable in its initial state.

Data Collection

In order to collect human data, we had to simplify the game as much as possible. To do this, we created a random generator in WebPPL that would come up with towers made up of a set number of blocks that it was given. We limited the number of blocks to 5 in order to ensure relative stability of the tower. From here, we selected 12 stable towers that it created and included them in a survey. We tried to vary the structure of tower we used, as well as the height of the tower, for more representation.

Tower	Height
1	5
2	5
3	4
4	3
5	2
6	5
7	5
8	5
9	5
10	5
11	5
12	5

Table 1: Tower Heights (in Number of Blocks)

For each tower, we showed participants an image of the tower, and gave them no exact dimensions. We then told them how to compute their score (blockValue*distance), and from here asked them which block they chose to move, selected by the blockValue (which also represented the position of the block relative to the top block in the tower). We then asked them to decide how far they wanted to move the block.

Because it was hard to come up with an absolute unit of measurement for distance that was intuitive and easy to reason about, we had to come up with an analog. We asked participants to give us the distance they wanted to move the block as a fraction of the width of the widest block in the tower. We know the actual dimensions of these blocks, so from there we were able to easily convert their choices into numerical answers. This allowed the participants to be able to give us simple, intuitive answers that we could then easily convert to numerical data to compare to the simulation.

Simulation Design

To simulate this scenario, we recreated each of the towers that we gave the participants, representing them as lists of objects. From here, we fed each tower through an analysis function. This function would go through each block in the tower. For each block, the function would have a list of possible velocities to apply to the block as an initial instantaneous velocity, since WebPPL only allows an intial instantaneous velocity, rather than a constant one. From here, the function would simulate the world by applying each of the possible velocities to the block in the tower. It would then determine whether or not the tower had fallen. If it had, then this particular run would return a score of -1, so as not to be considered as a valid run. If it had not fallen, we made sure to check that the displacement of the block had been less than the width of the widest block, since we gave this restriction to our participants. From here, if the choice was still valid, we returned the score by multiplying the block by the distance.

We then repeated this for each block in the tower, which gave us a maximum score for each possible block's movement. We could then easily determine the overall maximum score by taking the maximum of each individual block's score. This became the optimal solution for our tower.

We stored this data as block and displacement pairs. From here, we were able to compare the simulation's answer to the human data for each of the towers.

Human Data (Nisha)

The towers that we gave the participants were mostly 5 blocks tall, with some being shorter to have more scenarios represented. Once we had collected enough data (approximately 30 participants), we determined the average and mode of the data for each tower for comparison to the simulation.

Tower	Block	Distance
1	3.78	14.687
2	4.21	15.9316
3	3.08	15.337
4	2.69	20.997
5	1.73	15.835
6	1.86	17.237
7	2.86	12.125
8	4.30	25.716
9	3.91	22.519
10	2.86	14.600
11	2.72	16.195
12	3.61	20.399

Table 2: Human Data Averages

Tower	Block	Distance
1	5	12.992
2	5	7.633
3	4	20.997
4	3	20.119
5	2	13.390
6	2	22.442
7	4	12.871
8	5	20.256
9	5	23.978
10	4	16.790
11	1	13.196
12	3	19.650

Table 3: Human Data Modes

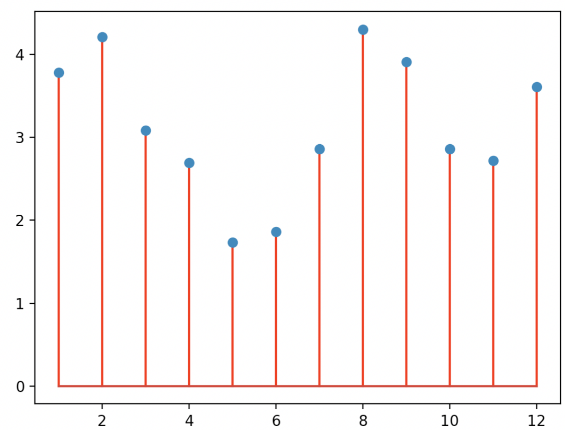

Figure 2: Here you can see a plot of the human average block choice. The x axis is the tower number, while the y-axis is the average block number chosen.

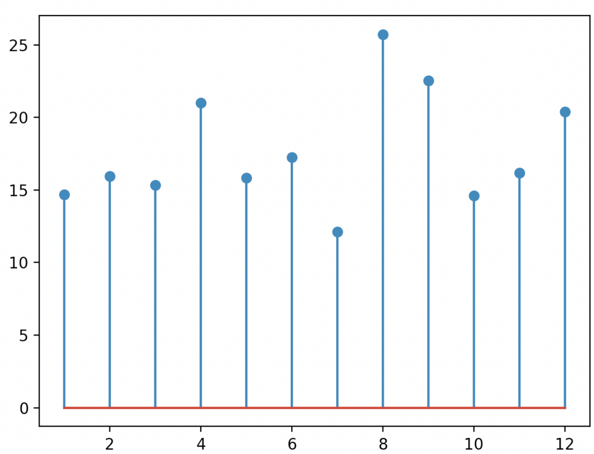

Figure 3: Here you can see a plot of the human average distance moved. The x axis is the tower number, while the y-axis is the average displacement.

In general, participants were able to agree upon a single block that they chose to move. In 7/12 of the given towers, a single block garnered over 50% of participant votes for the block they would move, indicating that there may be some level of uniformity in peoples' assumptions regarding intuitive physics. However, for 5/12 of the given towers, participants were unable to decide on a single block that they would move. This seemed to directly correlate with the number of blocks participants were given to choose from. The more blocks there were, the more uncertainty there seemed to be. Participants seemed to be far less certain about how far to move the block. No single distance had the majority of participants' votes, and most answers were widely distributed.

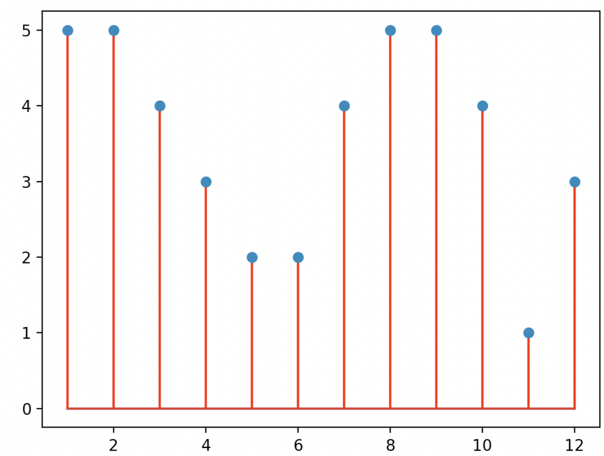

Figure 4: Here you can see a plot of the human mode block choice. The x axis is the tower number, while the y-axis is the mode block number chosen.

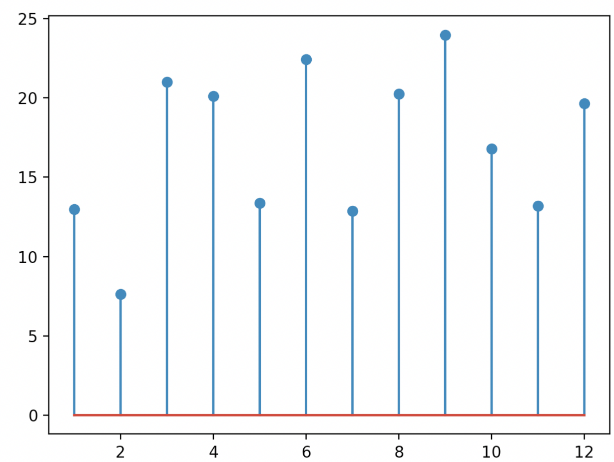

Figure 5: Here you can see a plot of the human mode distance moved. The x axis is the tower number, while the y-axis is the mode displacement.

Between the mode and average of the human data, we can see that the data in average block choice vs mode block choice is very similar. This tells us more confidently that users were mostly able to agree on the most optimal block to choose, which gives us more of an idea of a possible trend in the physics intuition of the participants. In the mode of the human block choices, we also see that participants often chose the bottom block to choose, which was surprising. It seems that people considered the tradeoff between the score multiplier and the distance limitations, and still decided to move the bottom block in order to get the multiplier.

The data in average distance vs mode distance looks more different, which confirms our belief that users were less confident in their intuition behind how far they could move a block, as opposed to which one to choose, since we were less able to see distinct trends in the distances chosen. However, in general, the participants never chose to move blocks more than half the width of the widest block, showing some level of caution when participants were deciding how far they could safely move things in the towers.

Simulation Data (Nada)

After following the process described earlier, we were able to collect the optimal solutions for each of the towers that we gave to our participants. This was the combination of block and displacement that maximized the score. From here we were able to compare the optimal solutions with the human data.

Tower	Block	Distance
1	5	12.323
2	5	37.900
3	4	34.732
4	3	28.803
5	2	58.879
6	3	17.932
7	5	23.811
8	5	19.735
9	5	41.027
10	5	27.933
11	5	20.158
12	5	32.221

Table 4: Simulation

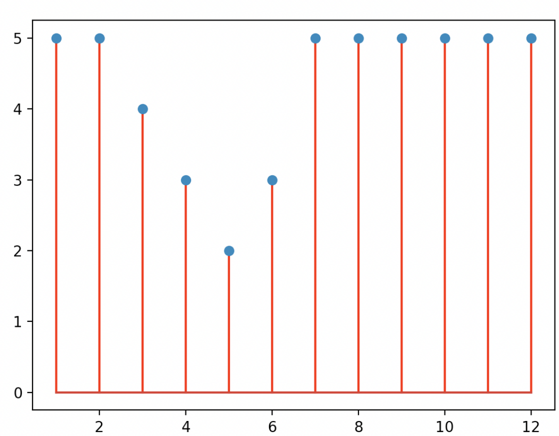

Figure 6: Here you can see a plot of the simulation's block choice. The x axis is the tower number, while the y-axis is the block number chosen.

An important trend that came up in the simulation data was that the simulation always chose the bottom block in the

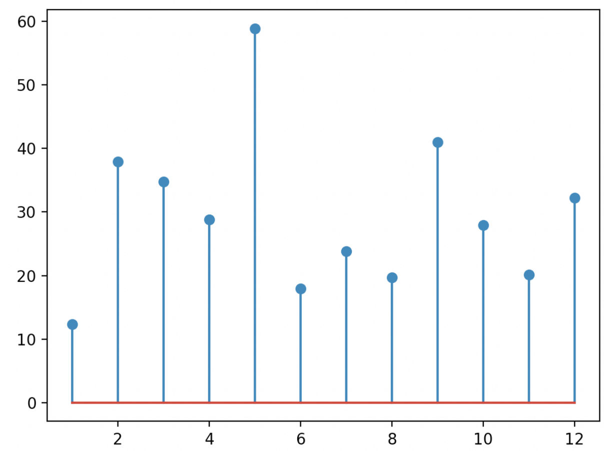

Figure 7: Here you can see a plot of the simulation's distance moved. The x axis is the tower number, while the y-axis is the displacement.

tower as the optimal choice. After thinking about this solution, we were able to realize that this is because the physics of the simulation often meant that if a block was moved, all of the blocks above it would move as well, but the ones below would not. In this sense, it becomes clear that moving something in the middle of the tower has less range of motion than the bottom block, since the block can only be moved as far as the block below it before the tower falls over. Since the bottom block was on the ground, it could move much further and still potentially not knock over the rest of the tower.

The simulation was limited in its movement the same way the human data was: a block could only be moved a maximum of the width of the widest block in the tower. With this limitation, our simulation was still able to push the blocks very far within the tower without knocking the tower over. When we looked at the data within the simulation, we were able to see that the simulation moved blocks by at least half of the width of the widest block, if not more. Part of this was most likely due to the fact that the simulation always chose the bottom block as the optimal solution, and the bottom block inherently had one of the highest ranges of motion out of any block in the tower, due to being on the ground rather than on top of a block of some significantly smaller width.

From here, it became clear that there were two optimal solutions for the towers if we only considered distance: this was the top block and the bottom block (the top block having no effect on anything below it, and the bottom block being able to potentially shift the entire tower). This is illustrated if we look at one tower and the trends that are seen in its movement distribution over the blocks. We chose to use tower 2 for this analysis, since it had 5 blocks and represented the simulation choice distribution well. This tower started off very stable, and also looked visibly stable since all of the blocks were more or less aligned along their centers. Using this tower, we were able to see the way that the simulation ran, and compare the distances the simulation was able to move each block in the tower successfully, without knocking the tower over.

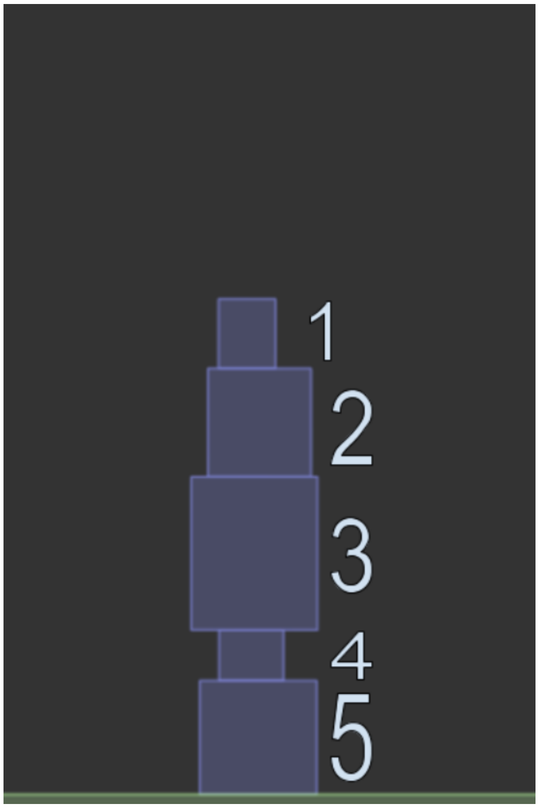

Figure 8: Here you can see an image of Tower 2 for reference as we discuss the data related to it.

Block	Distance
1	36.417
2	18.990
3	12.818
4	23.642
5	37.900

Table 5: Tower 2 Distance Distribution

In Tower 2, we can see that the simulation moved the first and last blocks in the tower much further than it moved any of the other blocks. From here, the optimal solution became block 5, since it had a multiplier of 5 on the score compared to the top block. Given this information, it's clear to see why the simulation always favored the bottom block – it was able to move any of the blocks so far that the multiplier always won in the tradeoff of block choice versus movement. This led us to believe that we may have poorly chosen our multiplier values, and should have made them smaller so that the last block would not always win out as significantly.

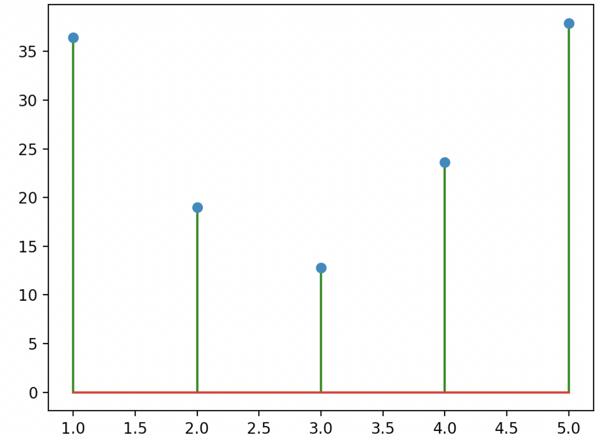

Figure 9: Here you can see a plot of each block in Tower 2 versus the maximum distance the simulation could move it. The x-axis represents the block, while the y-axis represents the distance it was moved.

Results (Nada and Nisha)

We ran the simulation on all towers, testing 'pushing' all blocks in each tower by applying an instantaneous velocity from $\_range(0, 400, 1)$. We then returned the max score that the simulation calculated for each block, and determined which score was the highest. We bounded movement of the block by the width of the largest block to align with the choices that we gave the participants we gathered data from. Then, given the block with the highest score, we divided out its "point value" from the score to get the distance it moved in the simulation, so we could consider both the distance as well as the total score in analyzing trends.

One observation that we immediately made was that the block that the simulation chose every time without exception was the bottom-most block, which was the one with the highest block score. We found this to be interesting because generally, the human data seemed to be a bit more conservative - in other words, people were too afraid to move the bottom-most block too much. This may be because its higher score implied that there was a greater risk to move it, and participants were subconsciously dissuaded from choosing it.

We also observed that the range of velocities we chose was important because the maximum score would change with the velocity. In WebPPL, an instantaneous velocity of 10 is almost nothing. We had to keep incrementing the range of velocities upwards until it was sufficient to push the block as far as it physically could move in the simulation. We eventually decided on a range of instantaneous velocities from 0 to 400 because it seemed sufficient enough to get close to knocking down most of the towers.

Once we had gathered comparable data, we were able to compare the human data to the simulation data. When we compare both the mode and the simulation data, we are able to see that the block choices surprisingly matched each other in a lot of the runs. Aside from towers 11 and 12, the simulation and the mode of the human data was mostly exactly the same, if not simply off by 1. This was very interesting to see, as it means that people were often able to see that they might be able to get away with pushing the bottom block enough that the extra score multiplier would be worth it. Because we were also able to see earlier that the human average and human mode were very similar, we can also say the same about the human average and the simulation. In general, there was a relatively high correlation between the simulation data and human data when it came to which block was the optimal choice.

However, when we came to the actual displacement of the blocks themselves, the data did not match at all with the human data. This surprised us, especially since we had limited the simulation to only move a block as much as the widest block in the tower. If we look at the displacement graph for the simulation compared to the human data, we see that the simulation moved the blocks significantly further than the participants in our survey did. This means that people tended to underestimate how far they could move blocks in the tower before it would fall over. Numerically, the simulation overall tended to move blocks as much as twice as far as the participants chose to.

However, if we look at the data run by run, we can see that there were some towers for which the human data actually was very accurate compared to the simulation's optimal solution. For example, in tower 1, the human data mode is almost exactly the optimal solution, choosing to move block 5 by 12.922. This, compared to the optimal solution of moving block 5 by 12.323, seems to be very accurate. Similarly, towers 4 and 8 were also very close to the simulation data.

In comparison, tower 2 was a particularly interesting case. In both the human and simulation cases, block 5 was chosen. However, the human data only wanted to move the block by 7.633, while the simulation moved it by 58.879, almost the entire width of the tower. This says a lot about people's intuition about these towers, as this means they thought the tower was too unstable to move far, while the simulation was easily able to move the tower quite a distance without any trouble.

When we looked at the particular towers that the data corresponded to, we were able to find a possible correlation. In towers where the optimal solution could not move the block very far, the human data closely matched the simulation. However, more stable towers tended to not match the human data, since the participants seemed to not trust any tower to be able to be moved very much, and erred on the side of caution all the time, while the simulation could always find the optimal solution. In this sense, we were able to see that humans were very cautious during this experiment, and were generally not confident that the tower would not fall, even when it was more stable.

Discussion

Gameplay (Nisha)

We ran into a few challenges over the course of this project. Our initial project intentions were to simulate a robot playing Jenga, the 3-dimensional version of the game we ended up with. However, the WebPPL simulator did not allow for this many dimensions, so we had to think about an analog example that would be representative of a game of Jenga while only requiring 2 dimensions. Because of this, we came up with the idea of having a simpler tower of stacked blocks, where the way to move the blocks is by applying a horizontal force to any given block.

An important difference between Jenga and our analog is that in Jenga, the goal is to remove a block on each turn. We realized that in two dimensions, the game would become trivial because players would simply remove the top most block of the tower on every turn. Because of our two dimensional constraint, we decided to change the goal of the game. In this way, we were asking people to move the tower as much as possible without removing any blocks entirely, as opposed to asking them which block was optimal to remove.

Human Data (Nada)

We also struggled with figuring out the best way to quantify people's intuition, especially when it came to how far they wanted to move the blocks. We wanted them to be able to answer us as simply as possible, in order to not complicate the situation and potentially bias the data. We also wanted to make sure to keep the distances absolute, rather than relative to any given block. We ended up coming up with a compromise, in that we had participants give the chosen displacement a number between 1 and 10, where 10 was a distance equivalent to the widest block's width, and 1 was 1/10th of the widest block's width. This was still not optimal, as the players would have less ability to represent the movement of bigger blocks vs smaller ones. However, it was the most intuitive option for our participants, so we decided to use it. This ended up working out better than we anticipated, especially since our simulation had the same maximum displacement limitation. However, it is still important to note that when a block had less range of motion, users were not able to express their displacement choices as precisely as when the block they chose had more range of motion (based on the width of the block beneath it).

Simulation (Nisha)

Furthermore, simulating the scenarios proved to be more difficult than we anticipated. Because we wanted our simulation to give us an optimal score (by deciding which block it should move and by how much), we needed a way for our simulation to run through many different possibilities before deciding on the optimal solution. This was complicated since WebPPL does not support for loops. However, we ended up being able to fix this problem by pre-computing a list of possible velocities to use. We then used a mapping structure such that we mapped each block to an optimal score, where the optimal score was found by mapping each possible velocity to a score, and finding the maximum for each block.

We also had trouble with simulating the motion itself of the block. The way that WebPPL runs simulations is such that we can only apply an initial instantaneous velocity to an object, rather than giving it a slower constant velocity. Because of this, we had to vary the displacement of a block simply by giving the block a lower or higher velocity. Because we did not give users the ability to tell us how slowly they wanted to move the block they chose, this means there may be some disparity between our result and the result that would occur if we used constant velocities rather than instantaneous ones. This might be part of why our simulation data did not match the human data quite as well. Users probably would have elected to move a block more slowly so as not to disturb the tower as much, but the solutions we compared their data to moved the blocks much more forcefully.

After observing the results of the simulation, we want to note that we do not believe that the simulation reflects 'real-life' trends with complete accuracy. Real Jenga takes into account the complexities of 3D space, which the WebPPL simulation does not necessarily seem to. Furthermore, humans playing Jenga have a visual feedback loop with their environment. In other words, they can slowly push a block until the tower starts becoming unsteady, which is something that most humans can see. The simulation has no such feedback loop, and simply 'pushes' a block with no regard for what might happen. This is a notable difference in the way that the simulation plays this game and the way that a human one. If we actually had the simulation imitate a human playing Jenga, where it constantly was checking for whether or not the tower was unsteady, our simulation results may have been closer to the human data we obtained. We believe that having a built in 3D engine for WebPPL would have helped us a great deal; however, as of yet, this extension to the WebPPL language has not been written.

Improvements (Nada)

If we were to improve this project, we would try to collect more data. First, we would simply get a larger sample size of data. Furthermore we would ask for more data for each tower. In this version of the data, we simply asked users which block they want to move, and how much they want to move that particular block. This made it difficult for us to see the trends in how far people thought they could move various blocks, since we only had their answer for one. If we reran this experiment, we would want to collect data such that for each block in the tower, we ask users to tell us how far they would move it. From here we could figure out their maximum score, and we would also have their entire distribution of the intuition they had for moving each block. This would have allowed us to see the trends in how far down a block was, and how much a participant wanted to move it. In this way, we would have been able to compare the simulation to the human data more directly, and more easily see the tradeoffs that people used in placement of a block versus how far they thought they could move it. This also would have allowed us to play with the block point values ourselves, and see which ones would have been most optimal, rather than having to work with potentially too-biased data due to poor choice in point values.

If we had more time, we also would have looked into using a different platform for this simulation that would have allowed us to apply constant velocities as opposed to simply instantaneous ones. This would have allowed us to choose an optimal solution that also considered different ways of moving a block to the same displacement, which would have allowed us to get higher scores in the simulation. This also would have better resembled the human intuition of wanting to move a block slowly so as not to disturb the tower. Alternatively, we could have told the participants that they could only displace a block by applying an instantaneous velocity. Both of these solutions would have ensured that the simulation and the human data was more similar in scenario, which would have made the data more valid.

Conclusion (Nada)

Although there was much room for improvement in this project, we were able to successfully create a simulation that represented a real world situation and gather comparable human data to observe trends between the real world and people's intuition.

We were able to conclude that human intuition for the best block to move given a weighted scale pretty closely resembles the optimal choice of block to move. However, we also concluded that the amount that people moved blocks much less than they actually could have, showing that human intuition for these towers was more conservative than it needed to be.

Acknowledgments (Nisha)

We would like to thank our TA advisor Tristan for always being there to answer our desperate emails about this project. We also extend our thanks to the entire teaching staff of 6.804 for being there in our times of need, and for helping us struggle through WebPPL syntax errors. Finally, we would like to thank the bartenders at The Muddy Charles for helping us hit our Ballmer peak, which was the necessary push we needed to come up with our modification to Jenga.

References

Peter W. Battaglia, J. B. H., & Tenenbaum, J. B. (2013). Simulation as engine of physical understanding. In *Proceedings of the national academy of sciences nov 2013.*