Aalborg University

Medialogy

Simulating Object Stacking Using Stack Stability

Portfolio

Author:
Kasper Kronborg Thomsen

Supervisor:
Martin Kraus

AALBORG UNIVERSITY
DENMARK

CONTENTS

Chapter 1 Introduction

In this portfolio, documentation that is not directly related to the system proposed in the main report will be presented. The raw data underlying statistical analysis will also be presented. Several approached that were not used in the final version of the system are also explained here.

Chapter 2 Methods That did not Work

For the implementation of the method presented in the report, a few approaches were tried that did not work as planned. Before the general brick stability framework was implemented, it was attempted to try and reduce the problems with the normal physics engine by using an energy correction system. It was quickly found that this system could only solve a minor part of the stability issues and so this method was quickly abandoned. When implementing the function updating the brick data, at first a recursive approach was attempted, but was abandoned as it had problems propagating correctly through the stack. Lastly, when the basic version of the brick stability system was implemented, a lot of functionality was implemented in not optimal ways. Improving the code base was done before the implementation of the version for leaning bricks.

2.1 Energy Conservation

The first approach to solve the problems with stable stacking in physics engines was to impose more strict energy conservation of the system. When investigating the energy levels in the stack, it was seen that the energy fluctuated when the stack should be stable. It was assumed that at some point these errors lead to oscillations and these cause structure collapse. It was assumed that the jitter and swaying from the stack was the main reason for collapse, but as later discovered the main problem was the penalty system in the physics engine as discussed in [Report 3.1].

The general idea for the energy conservation system is to limit the amount of energy added to the system from errors and approximations. The system logs the kinetic and potential energy of each brick in every physics step. If this energy is higher in the current frame than in the last frame, a correction function is used to reduce the total energy back to the level of the last frame. This is done by reducing the kinetic energy to where the total energy is equal to the last frame. This is done using the normalization function from the equations described below. The expressions for kinetic, potential and rotational energies are shown in equation 1, 2 and 3. The kinetic energy goal is shown in equation 4. This is only valid if the total energy in the current frame is higher than the last frame. In order to find the goal velocity for the object a ratio between the current and goal kinetic velocities is used

giving equation 5. This can be reduced to equation 6 and the goal velocity can be isolated in equation 7.

$$E_{kinetic} = \frac{1}{2}mV^2 \quad [1]$$

$$E_{potential} = mgh \quad [2]$$

$$E_{total} = E_{kinetic} + E_{potential} \quad [3]$$

$$E_{kinetic\ goal} = E_{total\ last\ frame} - E_{potential} \quad [4]$$

$$Ratio = \frac{E_{kinetic\ goal}}{E_{kinetic\ current}} = \frac{\frac{1}{2}mv_{goal}^2}{\frac{1}{2}mv_{current}^2} \quad [5]$$

$$\sqrt{Ratio} = \frac{V_{goal}}{V_{current}} \quad [6]$$

$$V_{goal} = V_{current} * \sqrt{Ratio} \quad [7]$$

Using this system to control the total energy for each object in a stack, the swaying and jitter is reduced to a point where they cannot be seen. Stack stability, however, is not improved. It is apparent that the swaying is not a significant contributor to instability in the stack. It is likely that the stability problem is a result of penalty forces being used instead of the usual collision response system.

2.2 Recursive Approach to Brick Stability

When implementing the current method for testing the stability of a stack, the first approach was to implement it as a recursive function that ran top down through the structure both calculating the necessary data and testing each brick and partial structure. The algorithm would start at a brick with nothing resting on it. It would call itself for all bricks that the current brick rests on, and also parse the collected data in the function call. It would then calculate the data for the current brick based on the data received. This way the algorithm should cover the entire stack in a top down way.

The recursive method worked well with the non branching stack; however, when branching was introduced, the recursive approach became less ideal. Due to the nature of branching, it is possible for the recursive algorithm not to have all the necessary data once it reaches a brick below a split in the stack. Due to this missing information the brick may be designated as unstable while it is in fact stable, or the other way around. Another problem with the recursive algorithm is that it may cover parts of the stack several times if the stack is not pyramid shaped. As the algorithm starts from every brick that has nothing resting on it, a stack with many top bricks, like a flat wall or a tower with several bricks in the top level, will waste many calculations on testing bricks that already have been tested in this frame. Due to these shortcomings it was decided to give up on the recursive approach. One

advantage of the recursive approach is how fast the system reacts to changes to the stack. As the recursive function calculates all stack data in one frame, any change will immediately take effect, where as the system described in the report will take between 1 frame and a number of frames equal to the number of contact areas in the stack, depending on the configuration of the stack.

Chapter 3 Optimizations of an Early Version of Brick Stability

After implementing the basic version of the stack stability system, see [Report 3.3], the code for the system was rewritten. This happened because much of the functionality had not been designed from the ground up to function in the framework and there were some duplicate functions doing the same calculations.

The main overhaul was to the stability test system. In the first version the different types of contact areas were each placed on different layers to avoid the ray cast in the stability test hitting the wrong type of contact area and returning a false positive. In the final version, however, the contact areas are all held on the same layer, but before the stability test the contact area type that is being tested for is moved to a separate layer, and moved back after the test. This simplification removes a lot of code and simplifies stability testing. In the first version of the system, there were also several functions for doing the stability test. One function handled individual bricks, another tested for individual contact stability and a third tested for stability of the stack . In the new version the stability test is collapsed to a single function that tests the brick stability against a specific contact area. The brick and the contact areas are input. The function always uses the centre of mass for the stack above, but due to the way the rest of the system works, when single object stability is needed the centre of mass for the stack above a brick and the bricks own centre of mass are the same.

Another place that is revised is the generation of contact areas. The first change is that two duplicate functions are joined together since they were virtually identical. The second change is the way contact points are send to this function. Before they were saved to the brick, now they are used directly as input in the function.

Chapter 4 Preliminary Test

4.1 Design

A preliminary test was carried out to test the system in its early state before the optimizations. In this test novice users were asked to play with the physical bricks for around 10 min. After this they were given one of two digital systems for 5 minutes. They were then given the other digital system for an equal duration. One was the proposed system and the other was the Unity 3d built in game engine with parameter modifications to gravity to allow for stacking. The physical brick session was always the first, in order to familiarize the participants with the physical properties of the bricks. The two digital systems were presented in counterbalanced order, so as to reduce potential learning effects of the controls, novelty effects and other confounding variables. During the test the participant had access to a leaflet with inspirational material.

The physical part of the test was filmed, with focus on the structures created. The video was mainly shot from a fixed camera, but structures that couldn't adequately be captured from the fixed angle were filmed using a hand held camera. The participant's screen was recorded using screen capture software. The goal of the filming was to examine the type of structures users construct and to find common structural elements in these. The digital screen capture was taken to identify potential errors in the system and to be able to replicate potential problematic structures. The screen that the participant saw had the prototype running, and a second screen, only visible to the test conductor, had a separate debug viewport that was also captured by the screen recording software. This debug view was used to get a second angle of the construction, and to see how the prototype handled the simulation of the structure.

After each test, a short semi-structured interview with the participant was conducted. This interview focused on the goals of the participant, comparisons between the two digital models, and the realism of these. The interview followed a preset script, but was extended if the participant had anything to add. The interview guidelines can be seen on the enclosed DVD in the folder "Test".

To allow users to interact with the prototype, an interface was developed. This consists of a 6 degrees of freedom controller, in this case the Razer Hydra [Sixsense, 2014]. The movement of the controllers is mapped 1 to 1 to the brick movement and rotation. The controller allows for picking up bricks, and for snapping rotation to 45 degrees intervals.

4.2 Results

8 people took part in the test, as well as one person carrying out a pilot test. Data from the pilot test were only partially usable, as modifications were made to the

software between the pilot and the main test. Therefore only data from the physical brick session could be used from the pilot test.

5 of 8 participants felt that the prototype system felt more realistic than the unity system. The remaining 3 could not tell any difference. The 5 people that detected a difference mostly noticed the reduced gravity. Two commented on friction behavior. 3 participants expressed preference for the prototype version as the most realistic, the rest of the participants didn't notice a significant difference between the models. A point of concern was the interface used. The controls had a significant learning effect associated with them, to the point that 2 participants commented on the learning effect, and 2 participants felt a difference in the controls between the two versions. Two participants, however, commented that the interface was well functioning, and easy to learn.

3 participants commented on the lack of interaction in some form or another. Two of these asking for collisions while stacking, and one noting that the model does not take object impact into account when calculating stability (the participant dropped a brick on the wing of a T structure). Two participants asked for support for moving several bricks at once, and moving entire structures. For both this related to their real world stacking behavior, where they placed several bricks at the same time. One participant noticed the inability of the model to handle card house stacking.

The stacking behavior with the physical brick observed in the first part of the test showed several consistent elements. Almost all participants for the most part build with the bricks lying flat. Some also used bricks set on their ends as pillars, but none used bricks on the side in their structures. Most participants built using overlapping bricks when building, as well as alternating direction bricks two examples of these structures are seen in Figure 1. Four participants used or attempted to use slanted bricks. 2 of these had two bricks leaning on each other as an inverted V as a part of their construction.

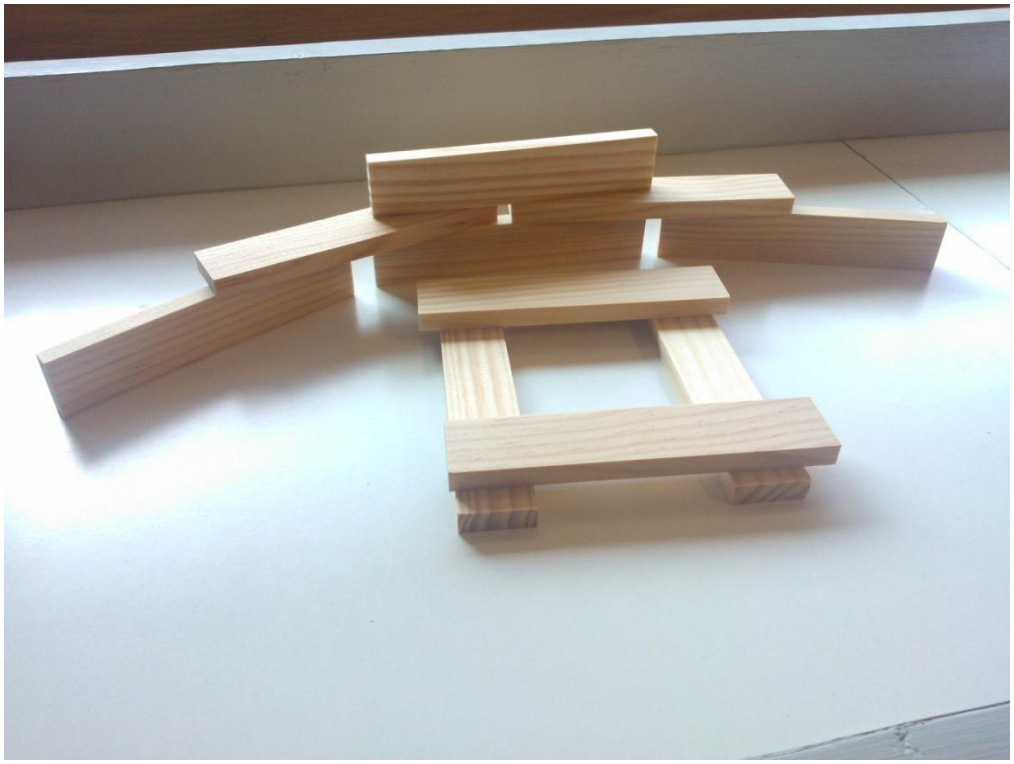

Figure 1: Two recreated stackin configurations.

4.3 Discussion

The test showed that several additional features are needed to be implemented to achieve better realism. The most noticed unrealistic element is the lack of physical interaction between stacked and non stacked bricks. Important structure types are the already supported overlapping structure and inverted V stacking which is not supported in this version.

Chapter 5 Test Systems

This chapter describes systems other than the bricks stability system used for the two tests. These are functions that were used for the test but are not directly associated with the brick stability system.

5.1 Brick Generation

One of the first pieces of extra functionality was a system to generate new bricks. This function adds the brick into the Unity scene, and also links the brick to the data structure for the brick stability system. The function also handles input related to adding bricks, both from the 6DOF controller and from the keyboard. The controller has the functionality to add a single brick at a specific location in the scene, and this functionality is also available from the keyboard.

Keyboard keys are used to generate the different structures used in the capabilities test [Report Chapter 4]. One key press generates one brick in the appropriate position in the structure. The position for the card house structure is hardcoded, but for the other structures the position is found using an equation based on number of bricks currently added to the structure.

5.2 6 DOF Controller

For user interaction in the tests a 6 degrees of freedom (6DOF) controller was used, namely the Razer Hydra controller [Sixsense, 2014] which is a dual 6DOF controller with 7 buttons and an analogue stick on each controller. This is used as the main method of manipulation of the system. The main button on the controller is used as the grab button. When the 3d cursor (a sphere) is in contact with a brick and the grab button is pushed, the brick will bind to the 3d cursor and match rotation to the controller. It can now be moved around by moving the cursor. Letting go of the grab button will drop the brick. While the brick is held, gravity is disabled for the brick, and its position is constantly set to the controller. This means that anything the brick bumps into will be subject to high penalty forces, if it is controlled by the physics engine. While the brick is grabbed, two snap buttons are available to be pressed. One snap button angles the brick in 45 degrees intervals, while the other snaps the brick to 80 degrees leaning in a fixed direction. The direction of the leaning is determined by the controller which holds the brick. This is to help users create the card house stack, as the two bricks will be able to form the supports of the card house. The snapping is implemented as a button hold but should be implemented as a toggle for better usability. As mentioned in the previous section, another button will generate a new brick when pressed.

Another important function of the controller is the clutch. When this button is pressed, the position of the cursor is fixed, and the controller can be moved independently. This is to let the user avoid real world obstacles by moving the controller away and around them. It also enables two controlled bricks to be close

to each other, as the controllers would otherwise bump into each other. While the button is pressed, the movement of the controller changes an offset variable which is added to the movement when the clutch is not held.

Camera controls were partially implemented but were deemed unnecessary for the test. A button for the translation of the camera works in the same way as the clutch. Support for camera rotation is also built into the controls. By rotating the input from the controller equal to the rotation of the camera, y axis rotation is enabled. In the current implementation the rotation happens with some offset, and the "centre" of the controls is also moved by this rotation. This leads to the controls becoming off centered by rotation. Enabling camera movement and rotation can help users better line up bricks, as they will be able to view the stack from different angles.

5.3 Settings for the Modified Physics Engine

For the preliminary test, the brick stability system from before the optimization is used. It has the same functionality as the basic version described in the report [Report 3.3] but some implementation is different as described in Chapter 3. For comparison in the preliminary test, a system based on the physics engine with parameter modifications is used. The scene in the program is set at a scale of 1 unit = 1 cm. the default 1 unit = 1 meter scale cannot be used as the graphics engine has difficulty operating at these scales, and the interface of the program also is difficult to use at this scale. The gravity in the scene is set at 98.1 cm/sec^2 which is equal to 0.981 m/sec^2 which is 0.1 times normal gravity. At this level of gravity the physics engine is able to generate somewhat stable stacking for reasonably sized stacks. The bricks have a friction coefficient of 0.35 and a mass of 0.02 kg which is similar to the real bricks, and the physics engine is running at 60 physics steps per second.

For the last two tests only small changes were made to this setup. The version described in the report as the unmodified physics engine had gravity set at 981 cm/sec^2, friction at 3.5 with a mass of 0.02kg. All other values are default. For the version described in the report as modified physics engine, the gravity is 98.1 cm/sec^2 and the friction coefficient is increased to 0.7 and the mass increased to 0.06 kg in order to make the bricks more stable in the reduced gravity. Both systems run the simulation at 60 physics steps per second.

The settings used for the brick stability system is similar to the ones used in the unmodified physics engine. 981cm/sec^2 gravity, a brick mass of 0.02 kg and a friction coefficient of 3.5.

Chapter 6 Raw Results from Tests

The results from the preliminary test can be seen on the DVD in the folder preliminary test. The videos from the capabilities test can be found on the DVD in the folder capabilities test. The answers from the questionnaires from the user test can also be found on the DVD, and the results of the Likert scales can be seen below.

Leaning Tower

Participant	Bricks fall	Tower stability	Object contact	Tower Collapse	Easy to construct	Overall realism
1	2	1	3	1	5	2
2	2	1	2	2	5	1
3	2	2	2	1	2	2
4	4	2	3	4	3	4
5	4	2	2	3	5	4
6	2	4	4	2	4	3
7	3	5	1	5	5	5
8	2	2	1	2	1	2
9	1	1	1	3	5	1
10	5	5	4	5	5	5
11	1	1	1	3	5	1
12	2	2	2	3	2	2
13	2	4	4	3	5	4
14	2	4	3	2	4	2
15	2	3	2	3	5	2
16	4	1	4	3	5	5
17	2	5	3	1	3	2
18	1	1	1	2	5	1
19	1	2	2	2	5	4
20	2	1	3	2	3	1
sum	46	49	48	52	82	53
mean	2.3	2.45	2.4	2.6	4.1	2.65
variance	1.273684211	2.260526316	1.2	1.305263158	1.673684211	2.134210526
standard deviation	1.128576187	1.503504678	1.095445115	1.142481141	1.293709477	1.460893742
standard error	0.252357307	0.336193866	0.244948974	0.255466549	0.289282233	0.326665772

Raw results for the leaning tower, 1 is preference for Unity physics engine, and 5 is preference for the stack stability system.

Card House

Participant	Bricks fall	Tower stability	Object contact	Tower Collapse	Easy to construct	Overall realism
1	3	1	4	4	5	4
2	2	1	1	1	5	1
3	2	1	2	1	5	1
4	2	2	2	2	2	2
5	3	4	2	2	5	2
6	3	3	3	2	5	3
8	2	1	2	1	5	1
9	1	1	1	1	5	1
10	5	4	5	3	5	4
12	3	2	1	2	5	2
13	2	4	4	2	5	4
14	3	4	3	3	5	4
15	2	3	2	3	4	3
16	1	2	1	1	4	1
17	1	1	3	1	5	1
19	1	1	1	1	5	2
20	2	3	3	3	5	3
sum	38	38	40	33	80	39
mean	2.2352941	2.235294	2.352941	1.941176	4.705882	2.294118
variance	1.066206	1.5662	1.492656	0.935625	0.595606	1.470606
standard deviation	1.0325726	1.251479	1.221743	0.967277	0.771755	1.212686
standard error	0.250436	0.303528	0.296316	0.234599	0.187178	0.294119

Raw results for the card house structure, 1 is preference for Unity physics engine, and 5 is preference for the stack stability system.

The comments from the user test can be seen on the scanned questionnaires on the DVD in the folder Test. The interview notes from the preliminary test can be found here as well.

Results from the capabilities test [Report Chapter 4] can be seen in the videos on the DVD in the folder Test.

Chapter 7 BIBLIOGRAPHY

Sixsense. (2014). Retrieved May 21, 2014, from http://sixense.com/hardware/razerhydra