Roller Coaster Simulation

CAP4800 , CAP5804 Computer Simulation
Dr. Paul Fishwick : FALL 2012

Team Leader:

Bharath Yarlagadda

Team Members:

Bharath Yarlagadda (CAP5805) - bharath.yarlagadda89@gmail.com

Juntao Zhu (CAP4800)- juntaozhu15@yahoo.com

Richard Daley (CAP4800)- rdaleyfl@ufl.edu

Goal of project

We want to simulate a real life event that would be challenging, interesting and fun. So we decided to simulate a queueing system of the crowd waiting for rollercoaster rides at an amusement park. In order to achieve this goal we must have proper tools, which is described below in Technical Requirements and Components section. This simulation consists of scenarios from crowd waiting in queue for the roller coaster ride to scenarios where the roller coaster ride is simulated and overall enjoyment factor with respect to the frustration of waiting in a queue is calculated.

We will guess the average fun rating a crowd can give to the ride based on the enjoyment they had on the ride and frustration they had waiting in the queue for the ride. We will be assigning some negative values for waiting in the queue if the time taken is more than the permissible waiting time. We will have some positive values for time they enjoyed on the ride and assign some ratings based on final value per customer and guess the ?ratings for the roller coaster.

1. Roles of Team members

Richard Daley's role consisted of research, design and building. He researched and studied the Open Source 3D Content Creation Suite called Blender. He used this suite to help create three roller coasters for this project. Before building the coasters, he made sure the roller coasters were of three uniquely different styles. This was to give the aspect of choice for the crowd. More on the constructing details of the coasters will be discussed later in this report.

Juntao Zhu's role consisted mainly of research, modelling, and texturing. He primarily researched both roller coasters and crowd dynamics in order to ensure that the model was accurate and consistent with real world data. He also worked on some of the modeling and texturing in Blender to help make the project look better and give it a sense of place. In addition, he also helped with other various aspects of the project such as contributing to some of the scripting of the model.

Bharath's role consisted of research and programming. He researched about the python scripting for Blender game engine and various features blender provides to generate crowd dynamics. He implemented the queueing system for the rollercoasters and also the intelligence of the humanoid when moving in the queue system for scenarios where one of the queues get filled then the natural tendency of humans is to go for another ride until the queue opens up or stay in the holding area until the queues are open again. In addition he also contributed in modelling of objects and creation of terrain in the project.

2. Abstract of report

For our project, we decided to create a fun and unique idea of a crowd and roller coaster simulation. The idea is to show the aspect of a crowd with three choices of three different roller coasters. The simulation consists of crowds entering entrance points for the three roller coasters and exiting exit points simulating the crowd behavior of lines at roller coaster rides.

In this simulation we have randomized the decision making behaviour of the crowd just like in real life and visualised the effect on the queues. So in a scenario where a certain ride is full a humanoid can choose to stay in the holding area or a pathway in the amusement park or

instead try another ride until the queue of other ride opens up. So scenarios like this show affect the size of crowd in the holding area and crowd in the queue and helps the authorities in making decisions to change the size of the queues or open up new exciting stuff to divert the crowd to avoid overcrowding of the pathways in the amusement park or control the flow of crowd into the system.

3. Introduction to the application domain area with citations to literature

For our project we decided to model the behavior of a crowd in a amusement/roller-coaster park. We did a wide variety of research on crowd behavior such as how crowds act when faced with a decision about which roller coaster to pick and how crowds act when the queue to a particular ride is very long. We did extensive research into crowd dynamics (3) and learned a lot about the psychology of how crowds behave and how a crowd tends to make decisions and act. We also learned more about the science behind how individuals in a crowd behave and what their behaviors has on the crowd as a whole (4). We also have researched boid particles in blender which are generally used to model flocking of animals(6) to check the feasibility of modelling a crowd.

In addition to researching about crowds we also did extensive reading about roller coasters and what makes a roller coaster work and also what types of roller coasters people are attracted too. We started by collecting a lot of more general information about roller coasters (2) and then by researching more in-depth about roller coasters in order to better build our understanding of the roller coasters and to ensure that our models are accurate with real world examples.

4. Implementation details with snapshots and discussion

Blender gave us the ability to work on different parts of the simulation simultaneously. This was possible because Blender has a feature that allows users to access different objects, meshes, scripts and more from different .blender files. So with this feature we could build the roller coasters while another would work on the crowd aspect at the same time. This gives us the opportunity to discuss the different design techniques Blender offers and our implementation of our simulation.

The roller coaster implementation at moments was a graphical design headache and at other moments a fun exciting challenge. The process first started with the track design of the roller coaster. The implementation involved multiple circle meshes extended into cylinder type shapes seen in Figure 1. We decided to make the tracks look like metal roller coaster tracks instead of the flat wood tracks. We only needed to create a short part of the track, because Blender had a special modifier called *Array*. This modifier creates an array of copies of the object, with each copy being offset from the previous one in a number of possible ways, like Pinocchio's nose when he lies. Vertices in adjacent copies can be merged based on a merge

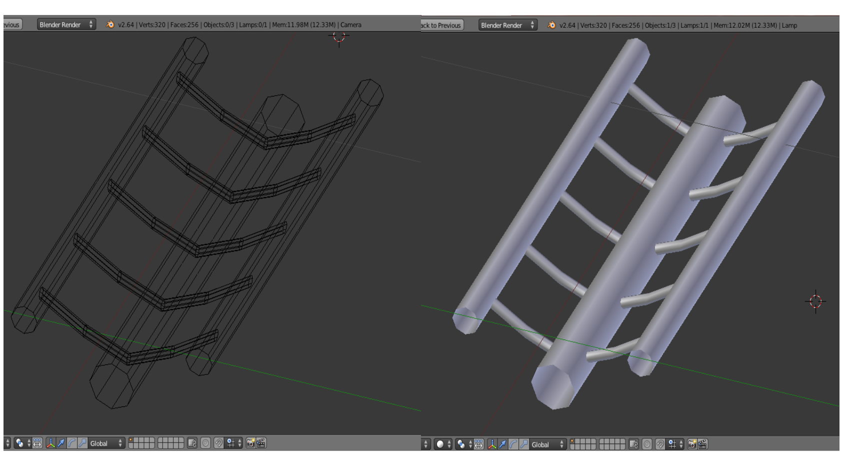


Figure 1. Here we can see the track in wire form (left) and solid object form (right).

distance, allowing smooth subsurf frameworks to be generated.

After the track was built, the next process was to create the roller coaster path. This was done by adding a curve. In Blender, curves are objects just as meshes are objects except they are expressed in terms of mathematical functions, rather than as series of points. An example of the curve object can be seen in Figure 2. The curve has many features that allowed us to extend it, move it in 3-dimensions, and change tilt. The process of changing these features took quite awhile. When moving/extending these curves we needed to adjust each point, created by the extension, appropriately in three dimensions. This required an exhaustive effort of adjusting point's x-axis, y-axis and z-axis to give the track our up, down, turning, flipping and spiraling motions. Once the curves were completed, all that was left was to fill out the curves with the track object created above. We can see a completed roller coaster track in Figure 3. He we can see the adjusted dimensions and tilt of the curves in more detail. When adding the track object through the Array function, we were about to see the tilt of said coaster. This is were diligent effort took place to adjust the proper tilt for each coaster. The tilt had to be adjusted properly for turns, flips and spirals, though we weren't using any physics script for the coaster, we still wanted the proper tilt for an accurate coaster design.

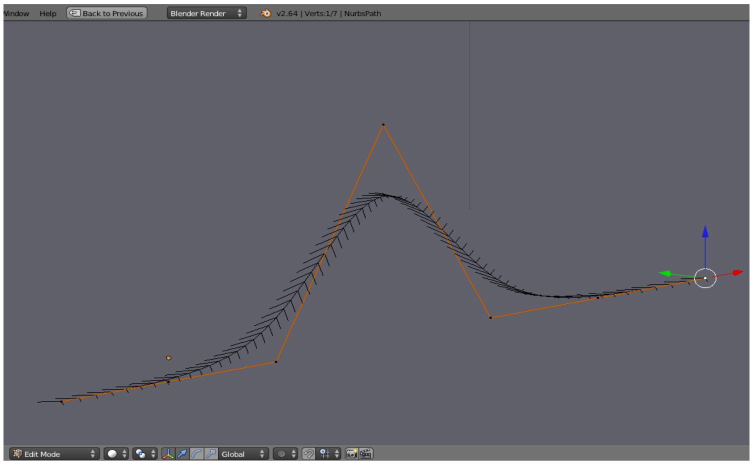


Figure 2. Example of a curve object with adjusted points to make a hill

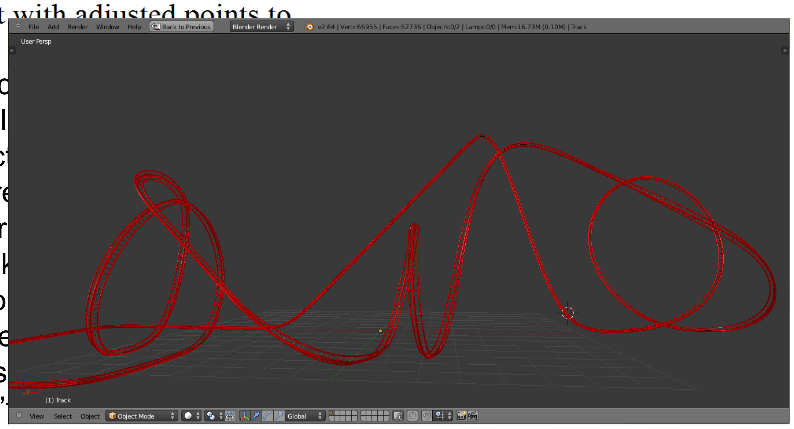


Figure 3. Example of a finished roller coaster

As the roller coasters were designed and we looked into queueing models and event scheduling(7) models to simulate the crowd. The first step was to create the map of the flow of crowd in the system. The second was to induce randomness into the decisions taken by the crowd in the queue system. As shown in Figure 4. Here we set up the basical pathing for the crowd objects to move around in this amusement park system. Initially we programmed the flow of the crowd in this map.

After finishing the basic modelling of the paths and objects we focused on simulating the crowd in such a way that they would react more like a crowd does in the real world by randomizing which roller coaster each member would take and and also by randomizing whether each member of the crowd wants to stay in the amusement park or exit the park. In general when we go to an amusement park if a queue to a ride is full we tend to go into another rides queue and come back later or stay in the pathway until a spot opens up on the queue. But if the ride is really in demand at that point of time the spot goes to whoever responds fast. So such random decisions are taken by the people when they are in a queueing system. So we programmed this into every humanoid of our crowd using python scripting in the Blender game engine. As this is a hypothetical amusement park we have assumed that a person exiting the park right away after a ride has a probability of 0.2 and has a high probability of 0.8 for going into the queueing system. The no of people in a queue for each rollercoaster is displayed for the user. This can help the amusement park operators in introducing new stuff to divert the crowd or controlling the inflow of the crowd based on the numbers. We can watch the entrance and exit of amusement park system by pressing C on the keyboard to switch between the cameras.

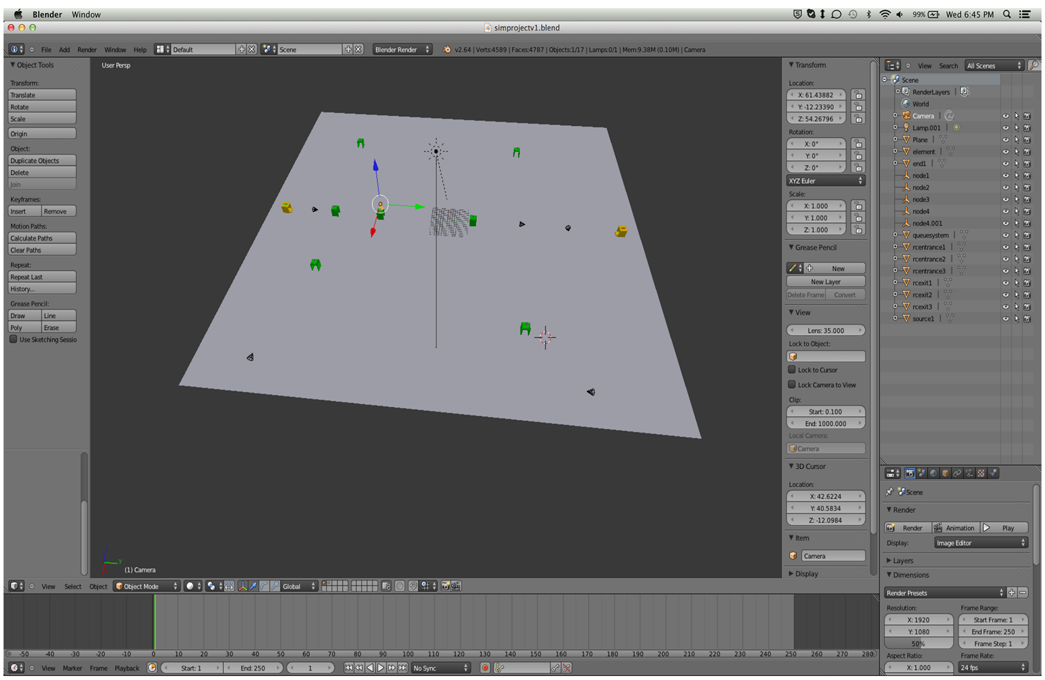


Figure 4

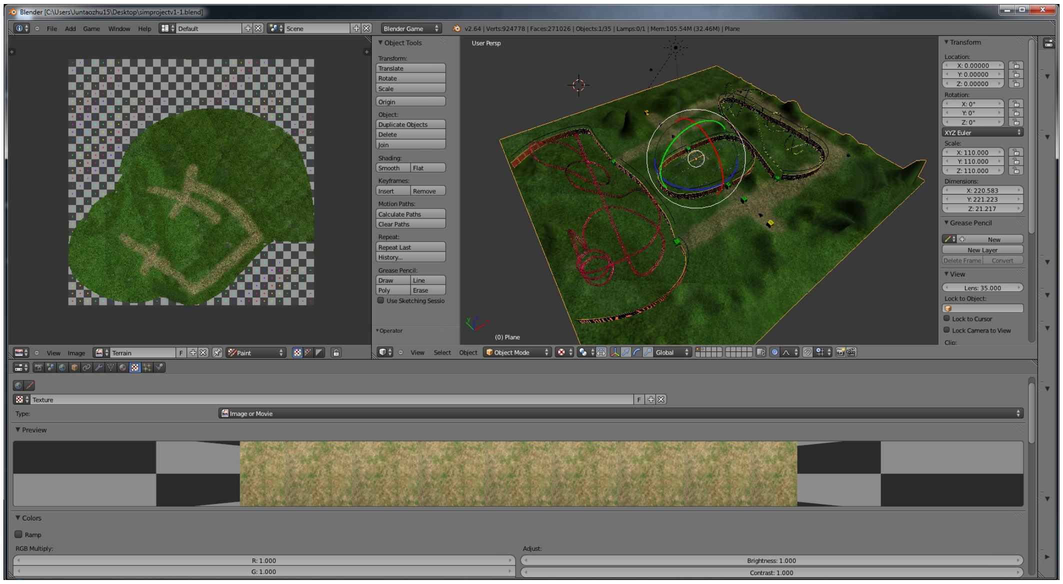
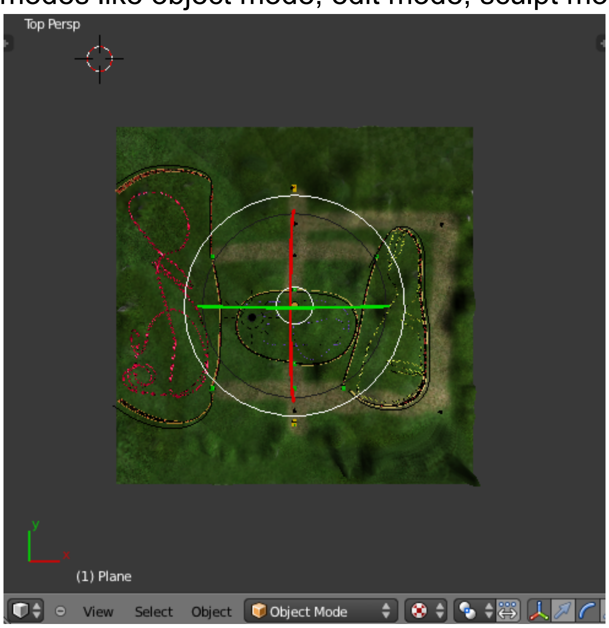


Figure 5

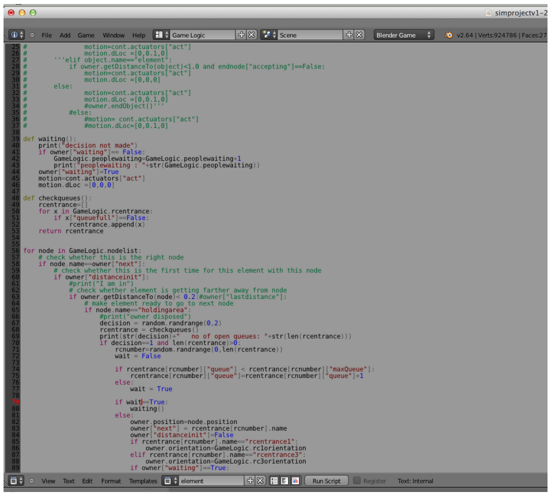
The final step in the modelling and simulation was to texture the map with a grassy texture and to texture the pathways that the individuals of the crowd would travel. This was done to create a more realistic and pleasing view of the simulation and to more accurately represent the real world environment of this simulation.

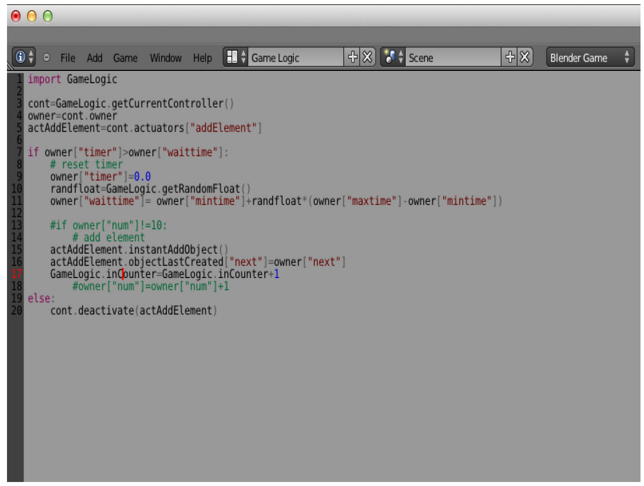
5. Details of interaction with the project, and interface snapshot.

This is a 3D window used to model various objects we used in our project. The building of objects and setting size, shape orientation of objects is done in this view by setting various modes like object mode, edit mode, sculpt mode and texture mode.



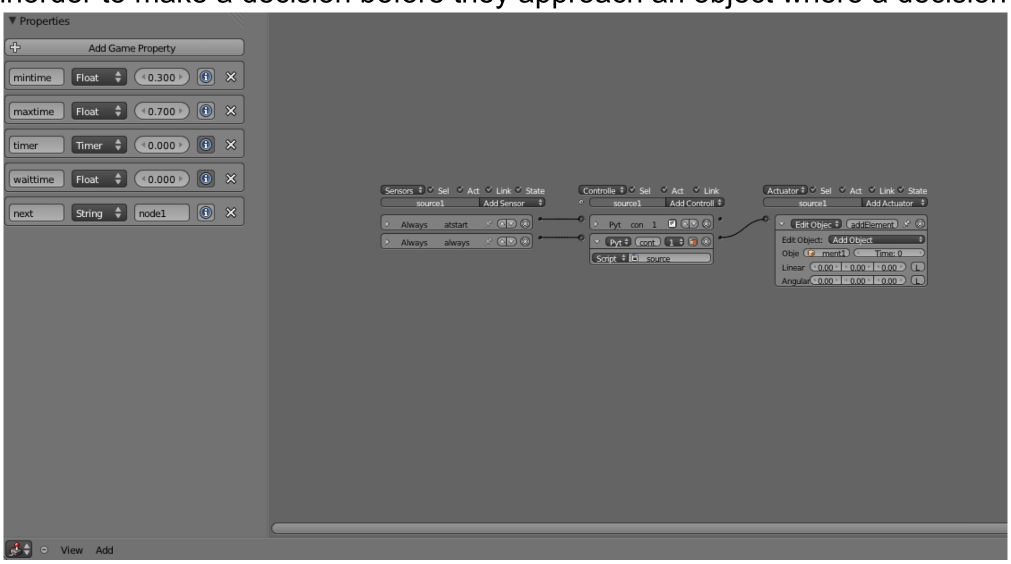
This snapshot is text editor window used for scripting. The python scripts we write are used as controllers which have the function of decision making or incrementing counters to generate new humanoids are intervals.





(Add some scripting pics with game logic too, and discuss them)

This snapshot is the node logic window where we connect the controller with sensors and actuators to make the system work. The sensors trigger the controller which is generally the python script. The sensors trigger the controller of humanoids in the crowd certain frequency inorder to make a decision before they approach an object where a decision should be made.



In order to run the project

i. open the .blend file of this project with Blender (new version)

ii. set the blender workspace to Blender Game.

iii. make sure that the 3D view is set to object mode.

iv. Now press “p” on the keyboard and the simulation starts running

v. We can switch between cameras in the simulation and view the size of the queues by pressing “c” on the Keyboard

6. Time schedule of actual effort with discussion of differences from planned effort

The time schedule we planned was not fully matched with what happened throughout the project. The research portion took awhile for our project, because we wanted to use a new tool that best fit our project. Blender was a struggle to comprehend for a little while. It took some trial and error to understand some of the tools within Blender. This only delayed our second task even more. Once we got started with creating the simulation we discovered that python was taking much longer to pick up than we expected and that because none of us are familiar with a scripting language there was a lot of trial and error with coding the details of the simulation itself.

The dynamics of the crowd and the queuing system also took some research and trial to work out how to get the ratios of the crowd to never overflow and how to accurately simulate a crowded system in an amusement park.

7. Challenges that occurred during the lifecycle of the project

In this project, we came across a few challenges using this new 3D creation suite Blender. First, the time and effort of research and practice to understand the loads of different tools within Blender including all the various options for modeling, texturing, simulation, and scripting. The time we spent just familiarizing ourselves with the tools and scripting was much higher than we anticipated. We also ran into a lot of challenges with deciding on exactly how we wanted to simulate the crowds and the rollercoasters and what exact aspects we wanted to focus on.

We also came across a variety of challenges with regards to what and how exactly we were going to model being as that we had trouble finding much data about modelling either crowds or the rollercoasters themselves. In the end we decided to focus on modelling the crowds which came with another set of challenges of deciding on exactly how we wanted the crowd to behave and deciding on how the crowd should act depends on the size of the line (queue) and exactly how each member of the crowd would decide which ride to choose and whether to re enter the queue or leave upon the completion of a ride.

8. Areas for future improvement

Some areas of improvement would be modeling and design, smoother interface and more complex simulation. Though, we must admit it was difficult to master a 3-D tool within the time we had. With more time, we could have possibly added more to the modeling aspect of the simulation. Possible roller coaster beams could have been added to make it more realistic. We could have improved on the cart design, and made it properly follow the roller coaster path with proper physics implemented. Another track style, possibly wood, could be implemented next time to give even more unique options for the crowd. We could have also made our model more simulator to a full fledged amusement park with vendors and other environmental objects to better convey the overall feel of the amusement park and to provide a more realistic visual simulation.

Another fun improvement for future work would be the scripting aspect. If we had more time to familiarize ourselves with Blender and python scripting, we could have possibly added another simulation to the project. The simulation of the physics behind a roller coaster, we could possibly simulation the speed of the roller coaster itself and the physics acting upon it. We could also simulate the physics of the cart and track and possible measure the differences between a empty and full cart. In addition, we could create a more realistic human form and also simulate more crowd dynamics and possibly have some people change their minds/take more time to decided which ride to take, or even randomly generate the chance of each person going on each ride so that everyone has a difference preference. We also can simulate groups of people in a crowd which represent a family or a group of friends that stick together throughout their stay in the amusement park.

9. List of references cited within the document

1) http://science.howstuffworks.com/engineering/structural/roller-coaster.htm
2) http://en.wikipedia.org/wiki/Roller_coaster
3) http://www.uni-kiel.de/psychologie/ispp/doc_upload/Reicher_crowd%20dynamics.pdf
4) http://www.economist.com/node/21541709
5) http://wiki.blender.org/index.php/Doc:2.4/Manual/Data_System/Linked_Libraries
6) http://wiki.blender.org/index.php/Doc:2.6/Manual/Physics/Particles/Physics/Boids
7) http://blenderartists.org/forum/showthread.php?191883-Event-Simulation-In-Blender-GE