COMP SCI 5407
Final Project Description

Aneesh Pavan Prodduturu

Natural disasters on three sport court scenarios

Project Description submitted for approval:

For my final project, I'd want to design three sports court scenarios, such as tennis, soccer, or basketball courts, and place them in various terrains, along with other natural disaster components, random balls, and nets. I'm attempting to make it as realistic as possible. Finally, I'd like to combine them with a switch node and animate them as best I can.

For the first court, I'd want to make a soccer court in a desert environment using a "mountain" node and randomly scatter balls on the court using a "scatter" node that moves when animated.

For the second court, I'd want to make a basketball court with uneven ground topography using a "mountain" node and an earthquake using an "RBD material fracture" and an "RBD bullet solver."

For the third court, I'd want to build a tennis court with a ball on the court and seating for the audience around the court, and I'd like to experiment with Houdini test geometry objects. Using Popnet, add heavy rain or a windstorm to one of the courts. Finally, I will incorporate the majority of the topics presented in class to make the overall project interesting, and I am open to any suggestions for my project.

Project Description:

I've divided the project into three halves. The first section is a simple soccer court with randomly moving balls on it, and then I added a snowfall as a natural catastrophe aspect. On the second court, I built a basic basketball court and included an earthquake as a natural catastrophe feature. The third court is a tennis court, and as a natural disaster feature, I put heavy rain and a cloud. Finally, I animated all three courts by adding them to a switch node.

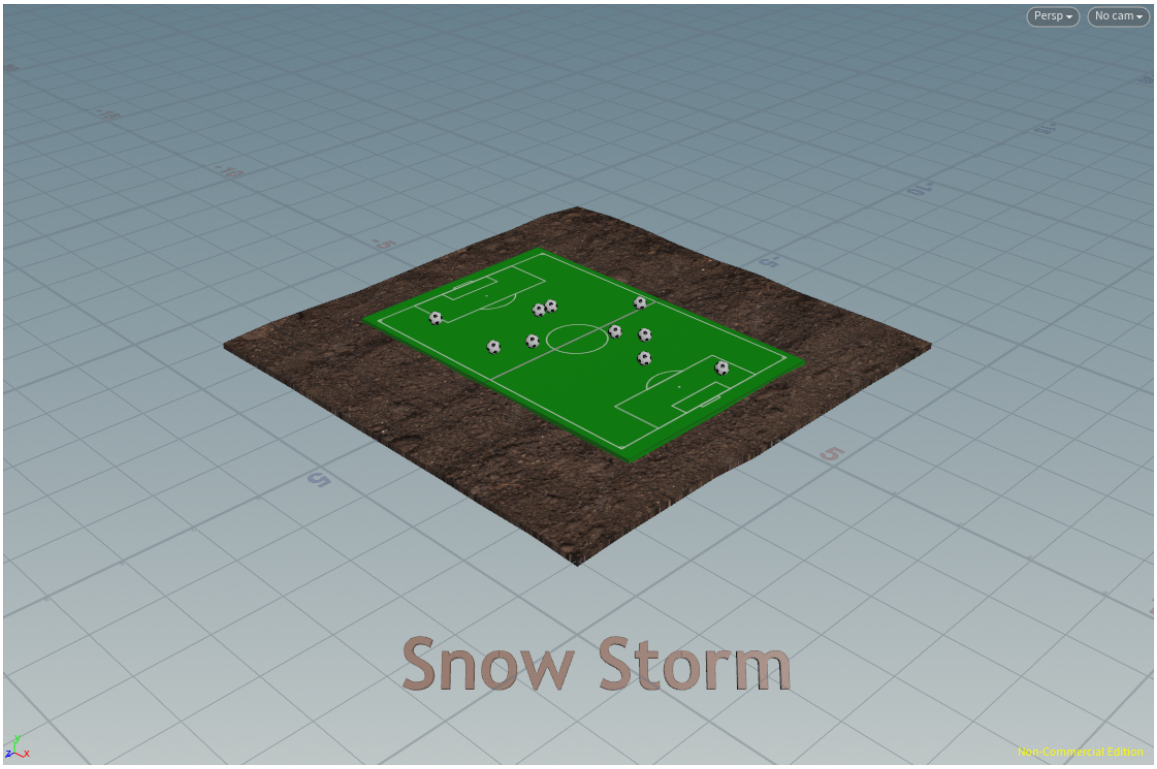
For the first half of the project, I need to build a simple soccer court, disperse randomly scattered balls over it, and then sprinkle snow over it. I began by constructing the network pane shape using objects and inserting a grid node. I modified the grid node's properties, such as increasing the size to 6 and 4, decreasing the center y-axis to 0.03, and increasing the number of rows and columns to 2 and 3, respectively. Then I added a polyextrude node and connected the grid's output to the input of polyextrude, increased the distance by about 0.076, checked the

output geometry and groups section again, and finally added the “uv quickshade” node to add the court image to the grid and connected the grid's output to the input of “uv quickshade” and added the image. This completes the development of the soccer court.

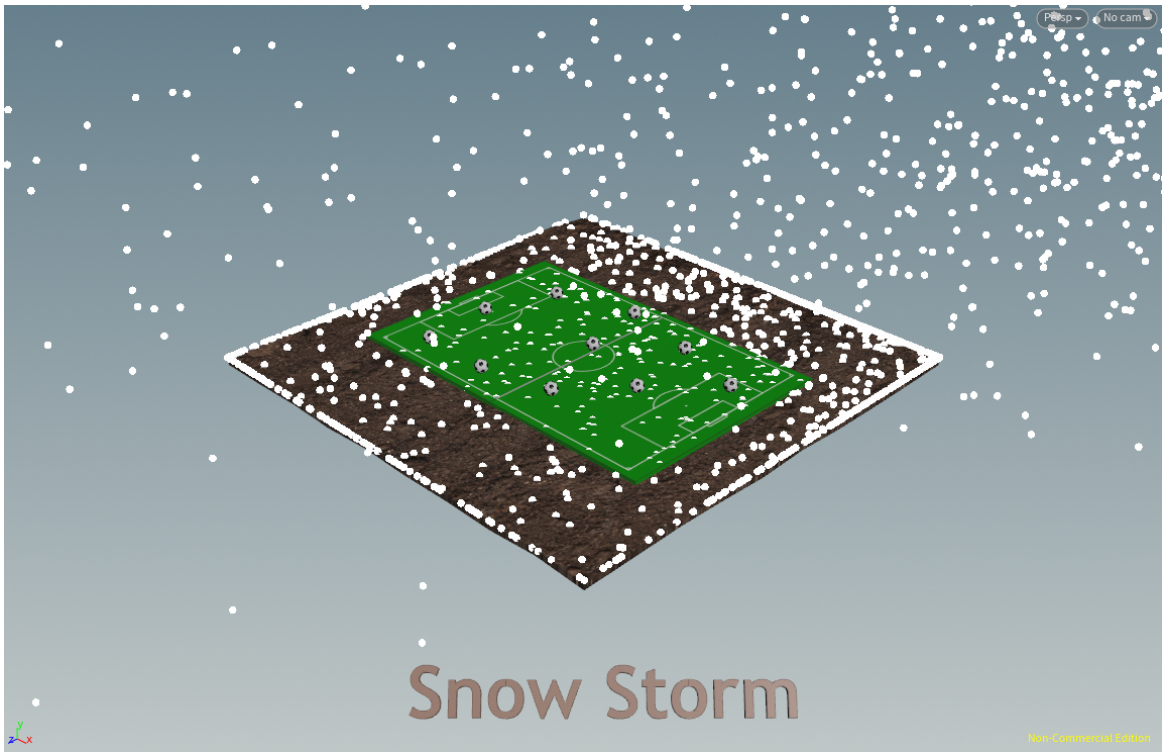
Now, add the balls. I investigated a new node called the platonic solids node, which contains a solid type called a "soccer ball," which generates a soccer ball automatically. In the settings pane, I changed the radius to 0.12 and the y-axis position to 0.191, and then I wanted to make them many, so I added a copy to the points node and linked the output of the platonic solids node to the “copy to points” node.

Then, I wanted to create a black soil terrain, so I started with a grid of 8 by 8 and used the help of a mountain node to uneven the grid node. I adjusted the amplitude node to 0.1, the noise pattern to “Worley cellular F1,” and the pulse duration to 10; I also connected the output of the mountain node to polyextrude and increased the distance by 0.1 to make it more attractive; and finally, I added a uv quickshade node to add a terrain image and simultaneously connected all of the end nodes to a merge node to see if everything was going as planned.

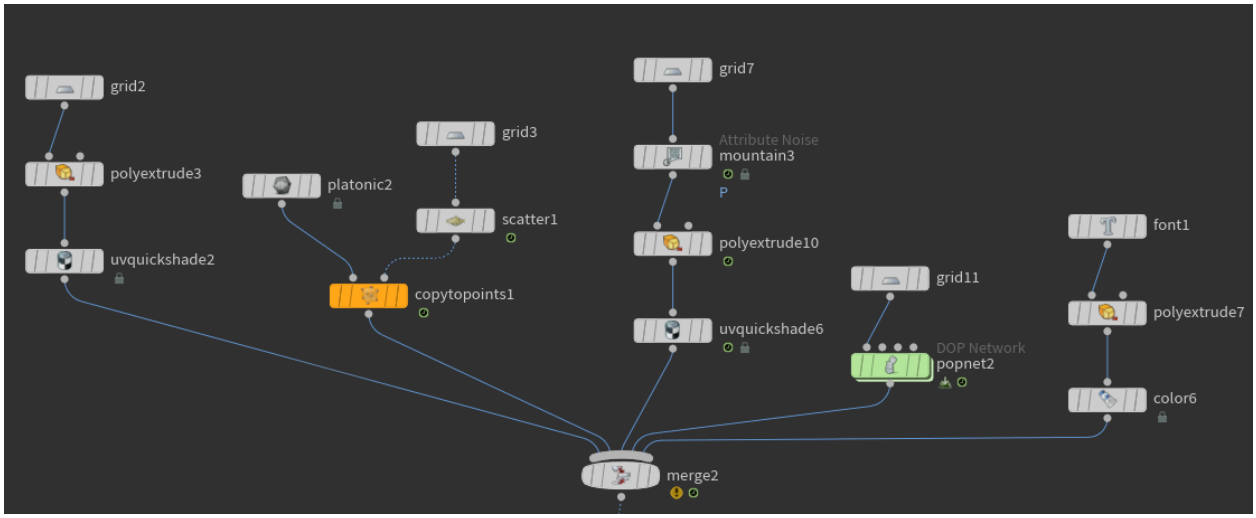
Now comes the snow. I made a grid node and changed the size to 5 and 5, the center x-axis to 7, the y-axis to 5, and the 5 and z-axis to -8.5, then rotated the x-axis to 30 and the y-axis to -40, and the left rows and columns to 10 and 10 as default, then added a new node called popnet to create particles and connected the output of the recently added grid node to popnet's first input. Then double-click on the popnet to access the popnet's dynamics in the network editor, where we may tweak various things to make it snow. As a result, I attached a gravity node between the merge and the output node that he uses to make the particles fall off the grid, which was previously linked to Popnet's second input. First, I made changes to the pop source node. In the birth section, we may specify how many particles the node should generate and the particle's life expectancy. I set the birth rate to 2000, the life expectancy at 30, and the seed to $F to change every frame. Then, in the popobject node, change the creation frame to 15, check "creation frame specifies simulation frame," "solve on the creation frame," and apply "object transform." Then, in the attributes section, I changed the velocity and variance to -7,-5, 8, and 10, 10, and 10 to steer the particles in the desired direction. I added two more nodes between the pop source node and the merge node: the pop force node, with force values of -7, -5, and 8, and the pop color node, with color set to white. These changes cause the snow to move properly, but it does not collide with the terrain and court we created previously, so I added another node called "ground plane" in the last merge just above the gravity node, which activates collision for the particles we just created, and then adjusted the ground plane node the same as the terrain from the geometry node so that when everything is combined, we see that particles have collided near the grid. In the collision behaviour section of the pop solver node, enable collision detection, impact data, and move to hit, and primarily set the reaction to stick so that the snow particles collide with the ground and adhere to it like snow. This completes the popnet dynamics adjustment, and we can press U in the network editor pane to return to the geometry node and connect it to the merge node. Finally, we added a font and adjusted accordingly so that we can see the disaster name using a poly extrude and the color node connected to the merge, and this completely completes the first part of what I planned.



And with the natural disaster element, this is how it looks:



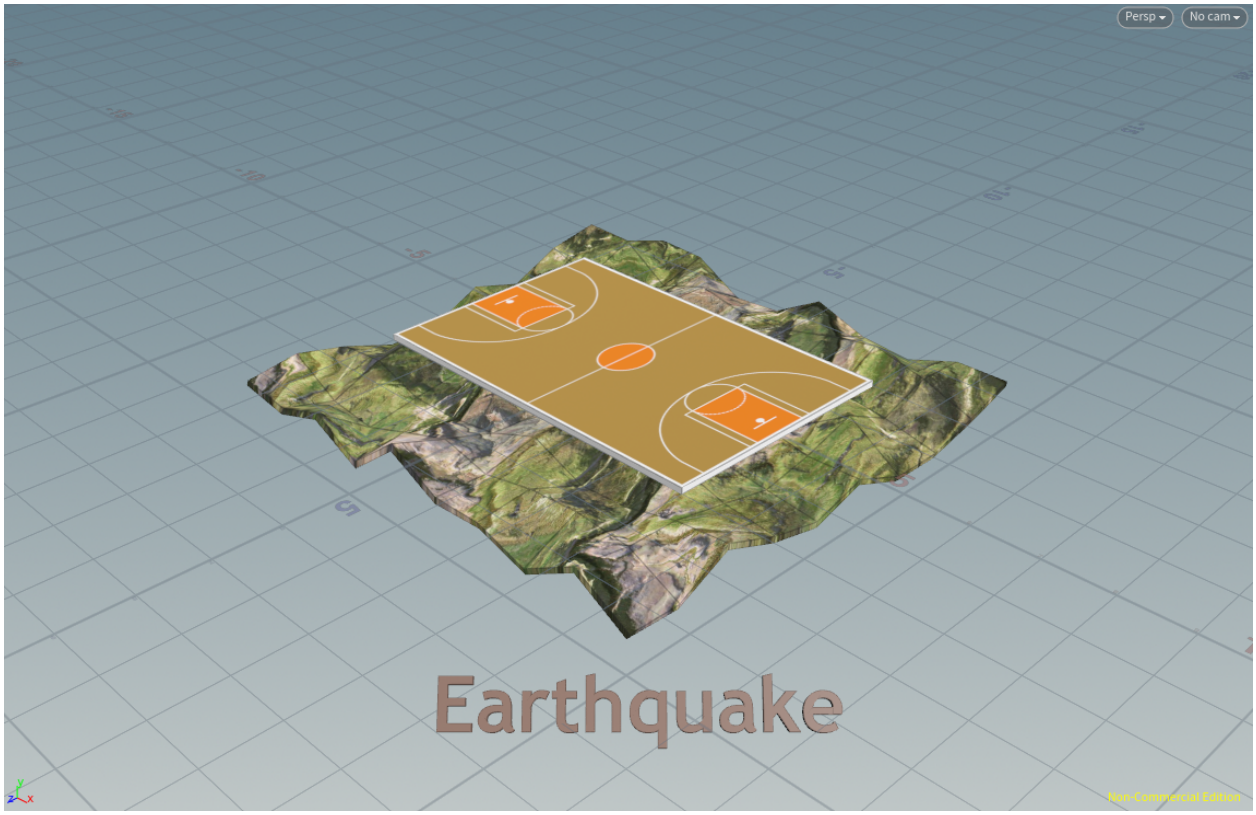
A screenshot of the network editor pane with all the nodes used:



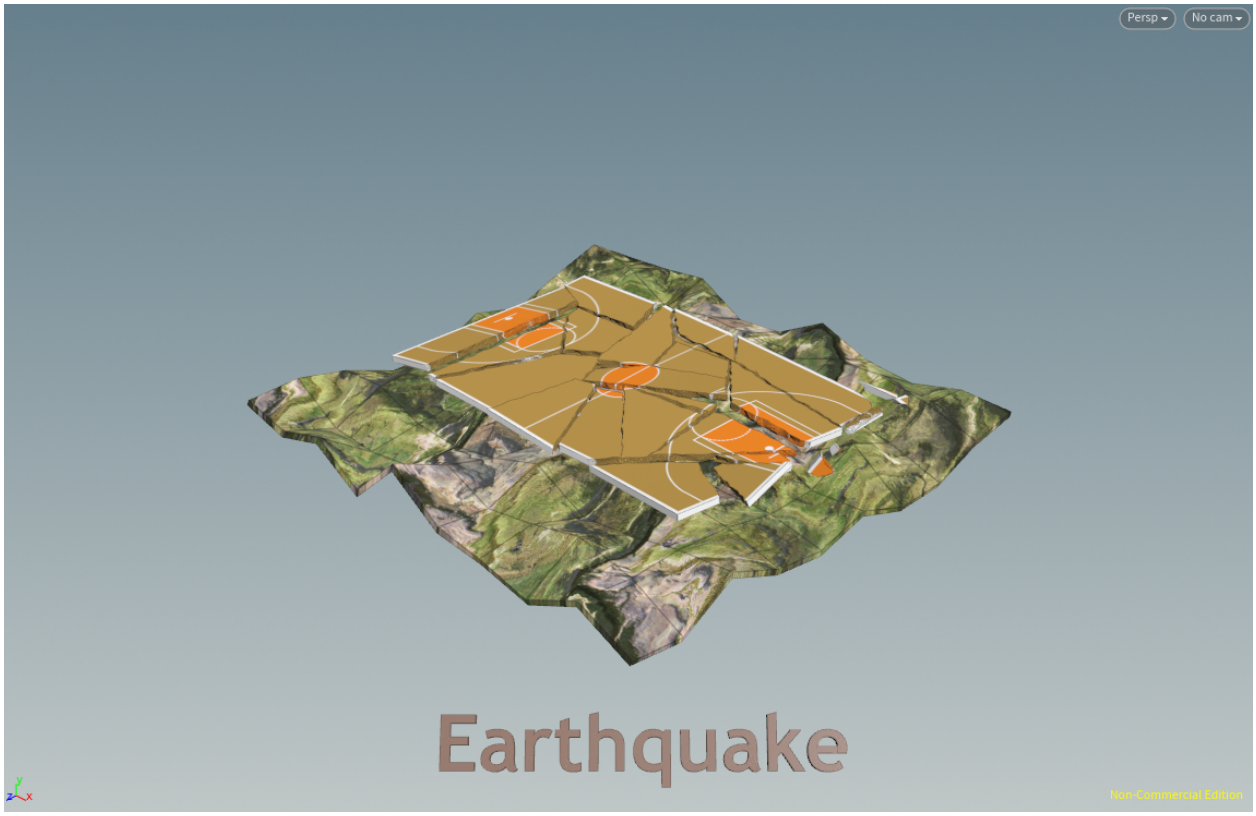
It's a simple basketball court, woodland landscape, and earthquake for the second half of the project. I began by creating a grid node. I changed the rows and columns to 2 and 2. Then I added a polyextrude node and connected the grid's output to the input of polyextrude, increased the distance by about 0.1, checked "output back" in the output geometry and groups section, and added the uv quickshade node to add the court image to the grid and connected the output of polyextrude to the input of uv quickshade and added the image, which completed the basketball court creation.

Now, I wanted to make a green forest terrain, so I started with a grid of 8 and 8 and used a mountain node to uneven the grid node, adjusted the amplitude node to 1, the noise pattern to "Worley cellular F1," and the pulse duration to 10, and also connected the output of the mountain node to polyextrude, increased the distance by 0.1, and added a uv quickshade node to add a forest terrain, and simultaneously connected all of the end nodes to their respective

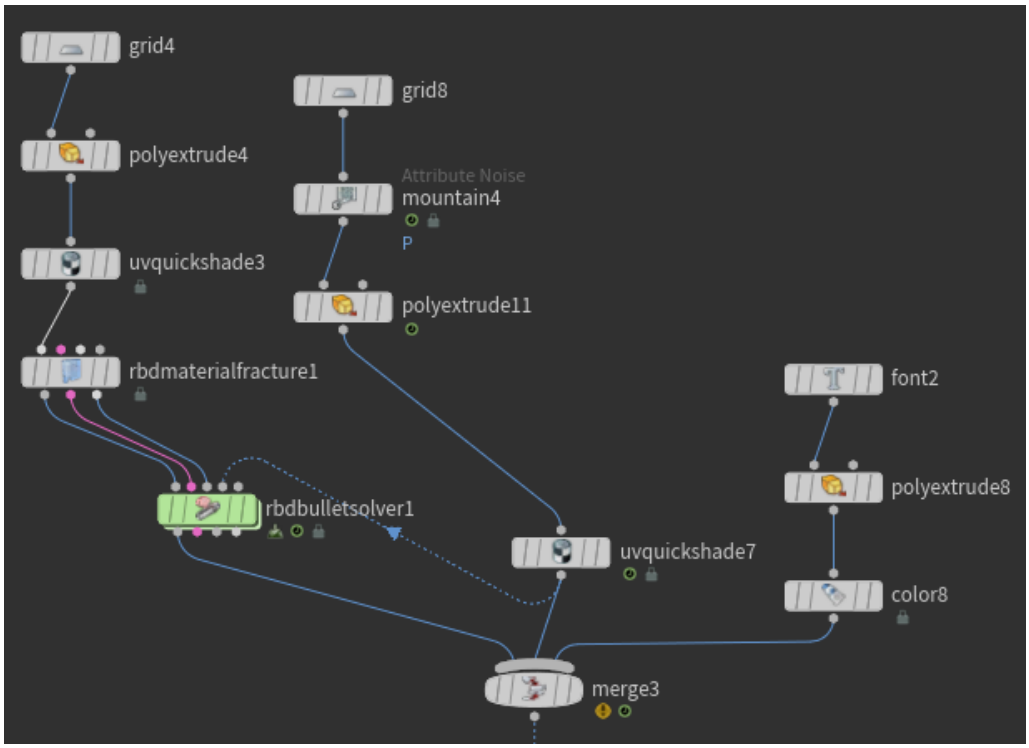
Then, to add the earthquake, I looked into two often used nodes named "RBD material fracture" and "RBD bullet solver," which would aid in fracturing and flinging the basketball court into bits. I began by adding the rbd material fracture node and connecting the output of the basketball court's uv quickshade node to the first input of the rbd material fracture node, which is a geometry input. I changed the material type to "concrete" in the parameters pane, scatter points to 5 in the cell points section, and noise type to "Alligator Noise" in the fog volume section, volume resolution to 75, noise frequency to the x-axis to 1, the y-axis to 1, and the z-axis to 1, and connected all three outputs to inputs of the rbd bullet solver node where the first output is geometry input, the second output is constraint geometry, and the third output is proxy geometry where Adjust collision padding in the properties section of the rbd bullet solver node, and also increase the bounce to 1.19 inside the physical section, then enable collision geometry in the collision section and change the collision type to static, the shape to the cylinder, and padding to 0.02, bounce, and friction to 1 and 1 inside the physical section. Finally, I added a font and altered it so that we can see the catastrophe name using a font, polyextrude, and color node attached to the merging, which completes the second half of what I planned.



And with the natural disaster element, this is how it looks:



A screenshot of the network editor pane with all the nodes used:



It's a simple tennis court, desert environment, test geometry, and severe rain for the third portion of the project. I began by building a grid node and adjusting some of the grid node's properties, such as size to 6 and 4, center y-axis to 0.095, and rows and columns to 2 and 2, correspondingly. Then I added a polyextrude node and connected the output of the grid to the input of polyextrude, increased the distance by 0.1, and checked the output back in the output geometry and groups section. Finally, I added a uv quickshade node to add a court image to the grid and connected the output of polyextrude to the input of uv quickshade and added the image, which completed the tennis court creation.

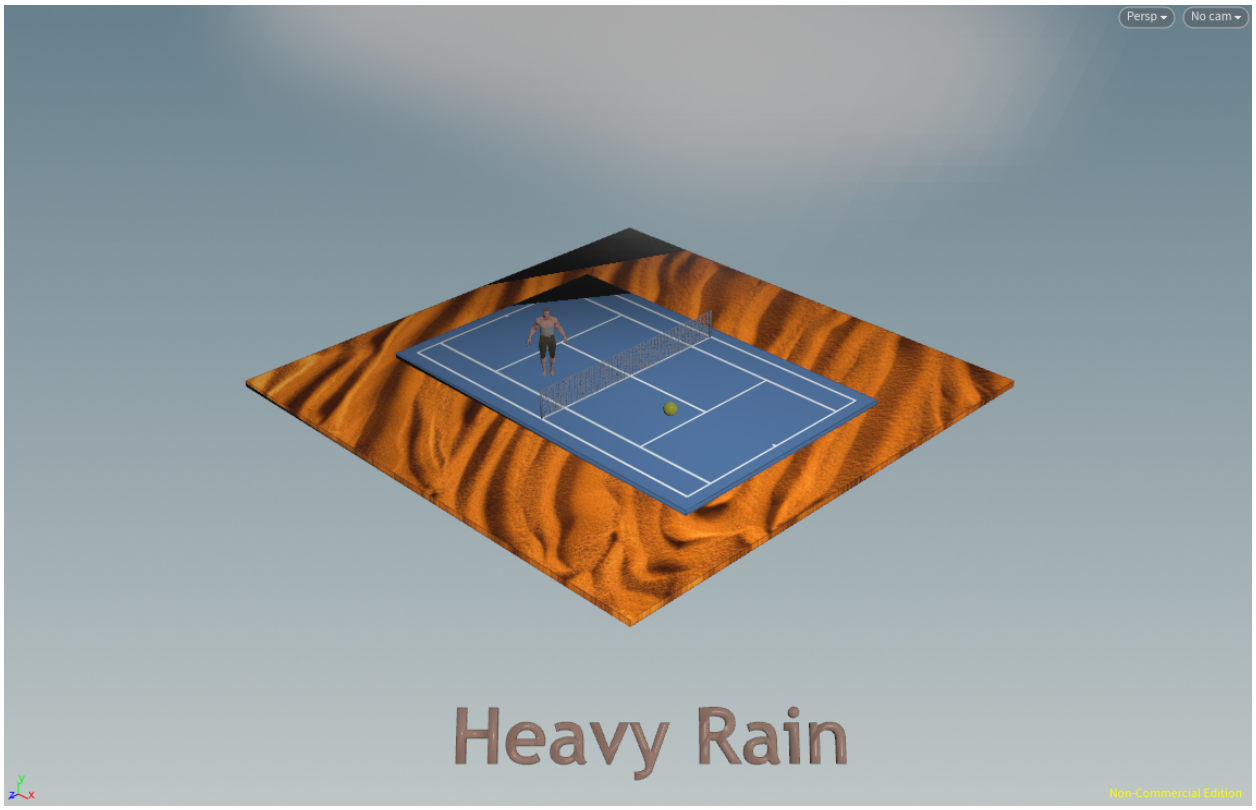
I also intended to make two tubes and a net to put in the center of the tennis court. For the tubes, I simply created a tube node with a center y-axis of 0.395, and a y-axis of 1.74, enabled caps, decreased the radius scale by 0.016, increased the height to 0.5, and decreased the rows and columns to 2 and 10, respectively, and later added a color node and connected the output of the tube node to the input of the color node and added blue color to it; for the other side of the tube, I simply cloned the tube and color node and then added a grid node to the network. I used sizes 3.46 and 0.5 and adjusted the center correspondingly so that they stayed linked to the tubes I already made. I set the center x-axis to -0.004, the y-axis to 0.395, and the left z-axis to 0. After increasing the number of rows to 20 and columns to 100, I added a polyextrude node where I utilized distance and twist to give a pattern to the grid and make it appear like a net. I first separated the polyextrude into individual parts, set the distance to 0.008, the twist to 1, and disabled the output front and rear so that just the output side was enabled, creating the appearance of a net.

Then I created a ball and a human and placed them on the court. For the ball, I just used a sphere node with radius 1 in all x,y and z axis and center x-axis to 1.2, y-axis to 0.2929 and z-axis to 0.40 and decreased the uniform scale to 0.1 and then added a color node to color the ball with neon color and for the human, I tried using test geometry called Tommy and adjusted translate x-axis to -0.9, y-axis to 0.197 and z-axis to 0.8, rotate y-axis to 50 and uniform scale to 0.6 and merged all of these to the merge node which I was previously using for this court scenario

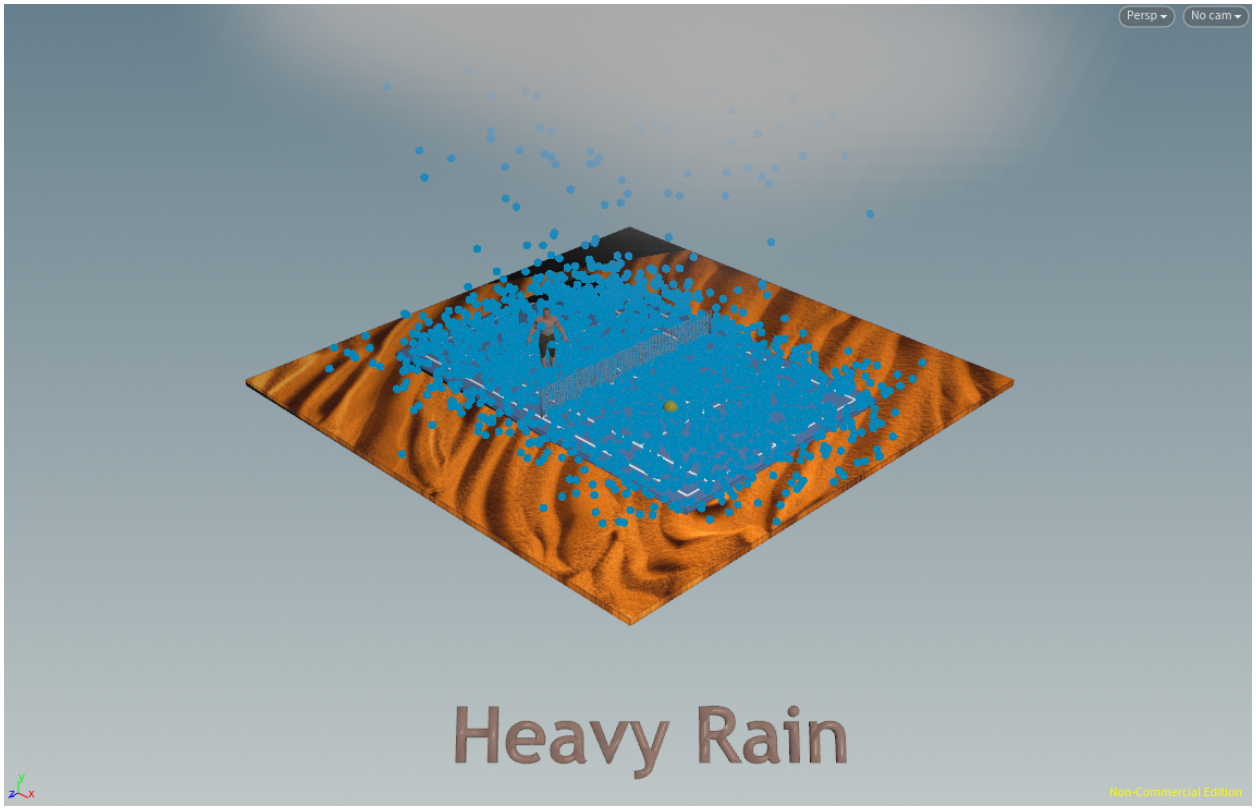
I also wanted to add a cloud to the top of the court, so I made a grid with the dimensions 7.5 and 5, set the center y-axis to 5.5, rotated the x-axis to -5, and attached the grid's output to a new node called iso offset to make the cloud. Where I modified the offset to 0.64 and changed the output to fog volume and mode to point cloud, the cloud on top was completed.

Coming to the catastrophe aspect, I used a grid, popnet, and color nodes to produce severe rain. In the grid, I changed the size to 6 and 4, respectively, and the center y-axis to 5, and connected the output of the grid node to the popnet node. To make changes inside the popnet node, double-click on the popnet to open the popnet dynamics in the network editor, where we can adjust many things to create heavy rain. Starting with the pop object node, I increased the creation frame to 165 based on the planning of my switch node at the end, and in the collisions section, I increased the tolerance to 0.0028. Moving on to the pop source, I changed the emission type to "scatter onto surfaces" and the geometry source to "use first context

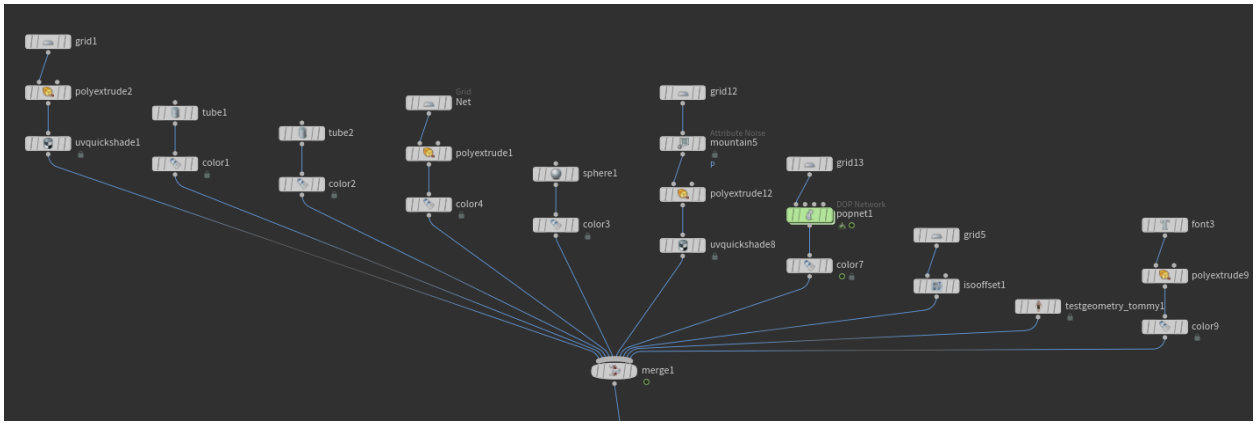
geometry," and in the birth section, I increased the birth rate to 8000, enabled max points per frame and set it to 50, increased the life expectancy to 80, and created and adjusted the variance to 1 for all axes. The particles then fall as planned but do not collide with the ground, so I researched it and made some changes in the popsolver node, such as in the collision section. I enabled collision detection, add hit attributes, move to hit, and response to slide so that when the rain collides it will slide on the ground like water, and enabled impact data and enable collision feedback, and to successfully work it out I also need to add a new node called ground plane and pop solver output as input and output of merge node to the output. This solved the major collision problem, but when I returned to the geometry, I could also see the ground plane, so in the popnet node, I had the option to select what was needed and changed the object to a pop object in the popnet object merge and connected the output of popnet to a color node to add blue color to the particles that I had just created. Finally, we added a font and altered it so that we can see the disaster name using a font, polyextrude, and color node attached to the merging, completing the third section of the project.



And with the natural disaster element, this is how it looks:

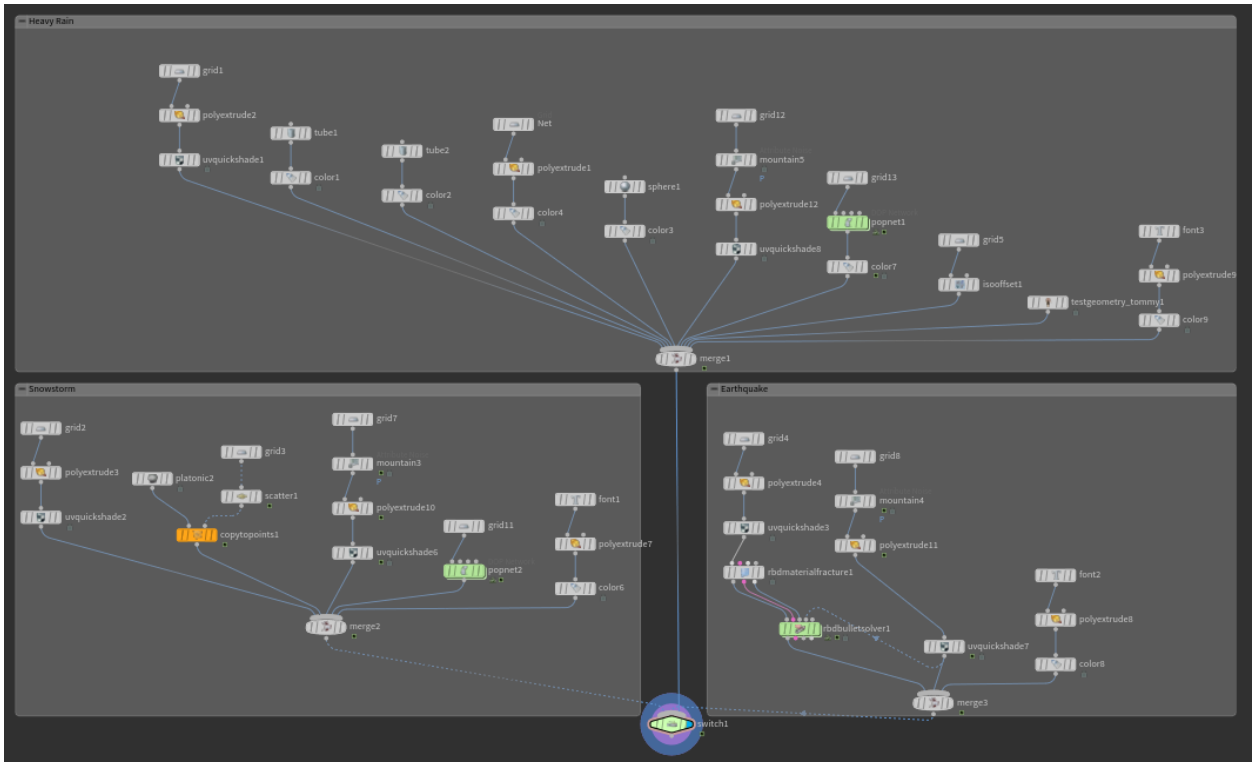


A screenshot of the network editor pane with all the nodes used:



Now I want to construct an animation and play the movie to demonstrate three sections of my project one after the other, so I introduced a new node called switch and used it to go from one part to another after a particular amount of time. I attached all three sections' end merge nodes to the switch node and set the input value animation to “floor($F/80)” so that it switches parts every 80 frames, dividing the 240 into three parts for my three court situations. Finally, my entire project is complete, as anticipated. In order to keep the project as organized as possible, I utilized sticky notes.

Here is a snapshot of the whole project network editor pane:



References:

- Class Lectures
- https://www.sidefx.com/forum/
- https://www.tokeru.com/cgwiki/index.php?title=HoudiniDops
- https://mrkunz.com/blog/08_22_2018_VEX_Wrangle_Cheat_Sheet.html