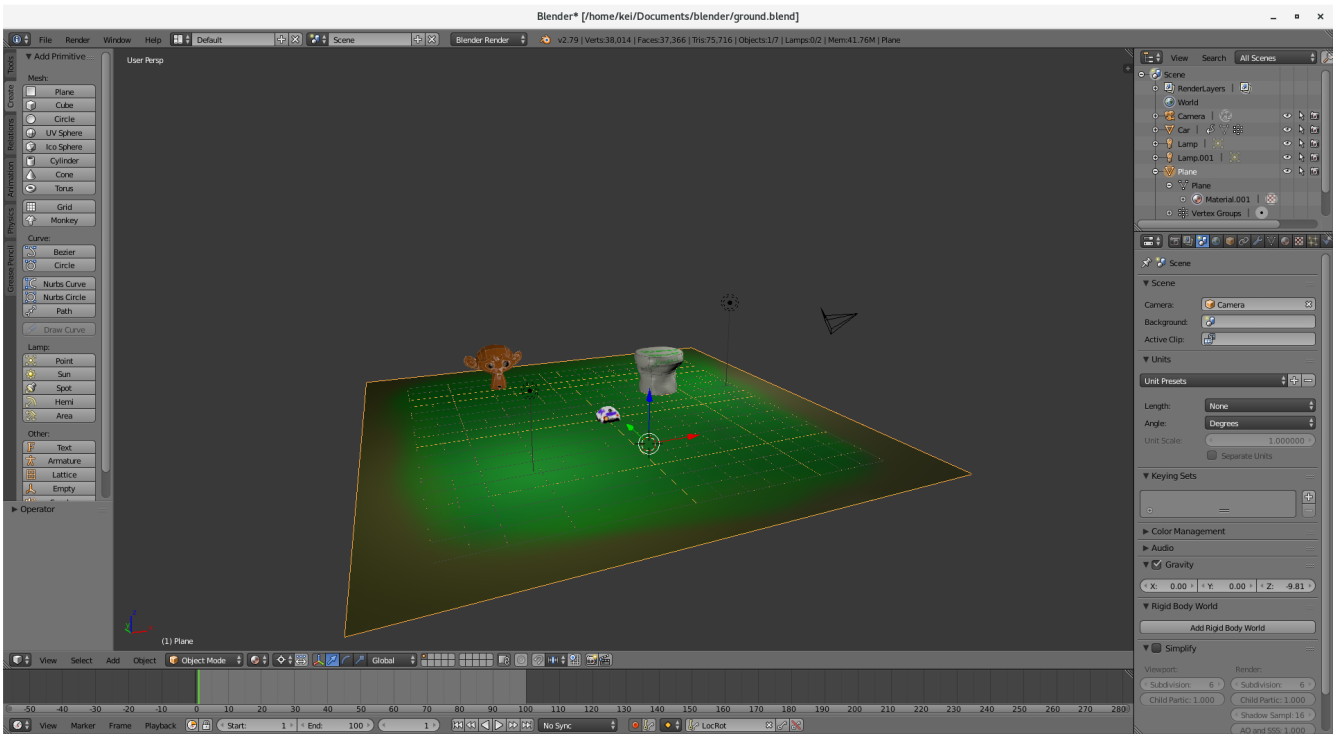
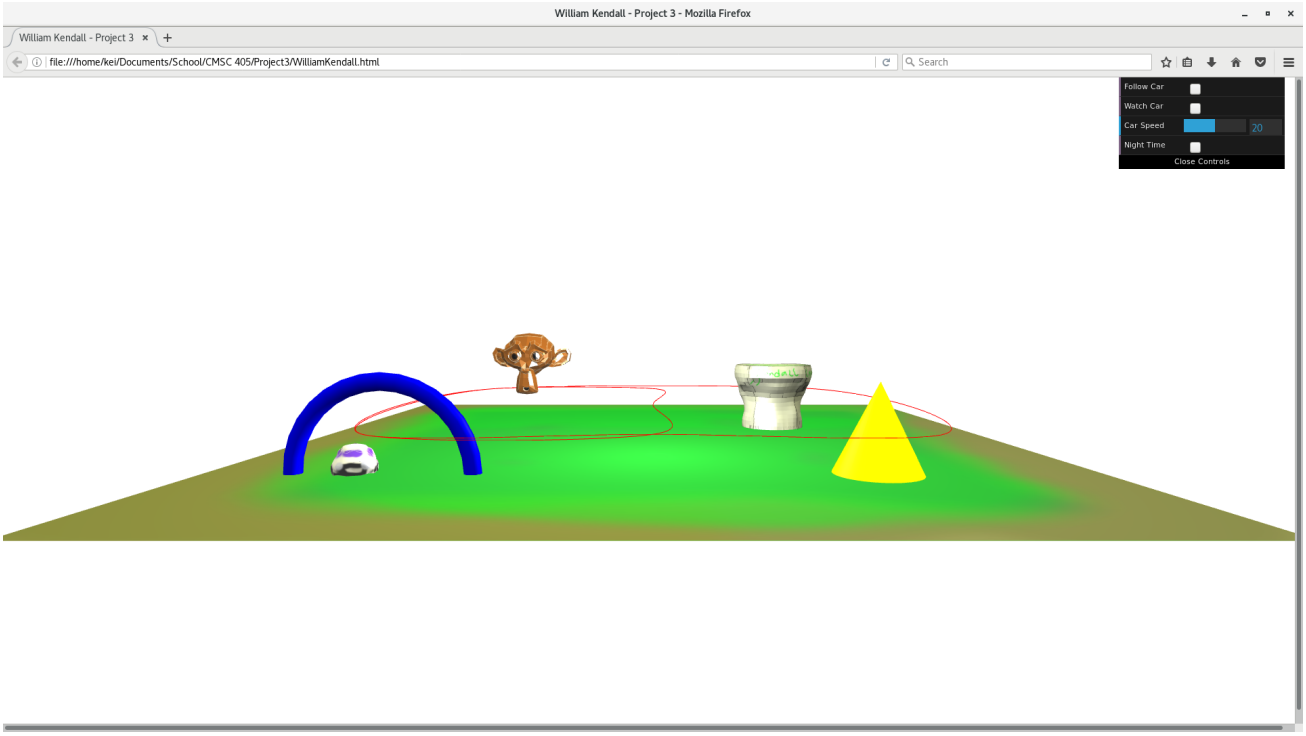
William Kendall
CMSC 405
Project 3

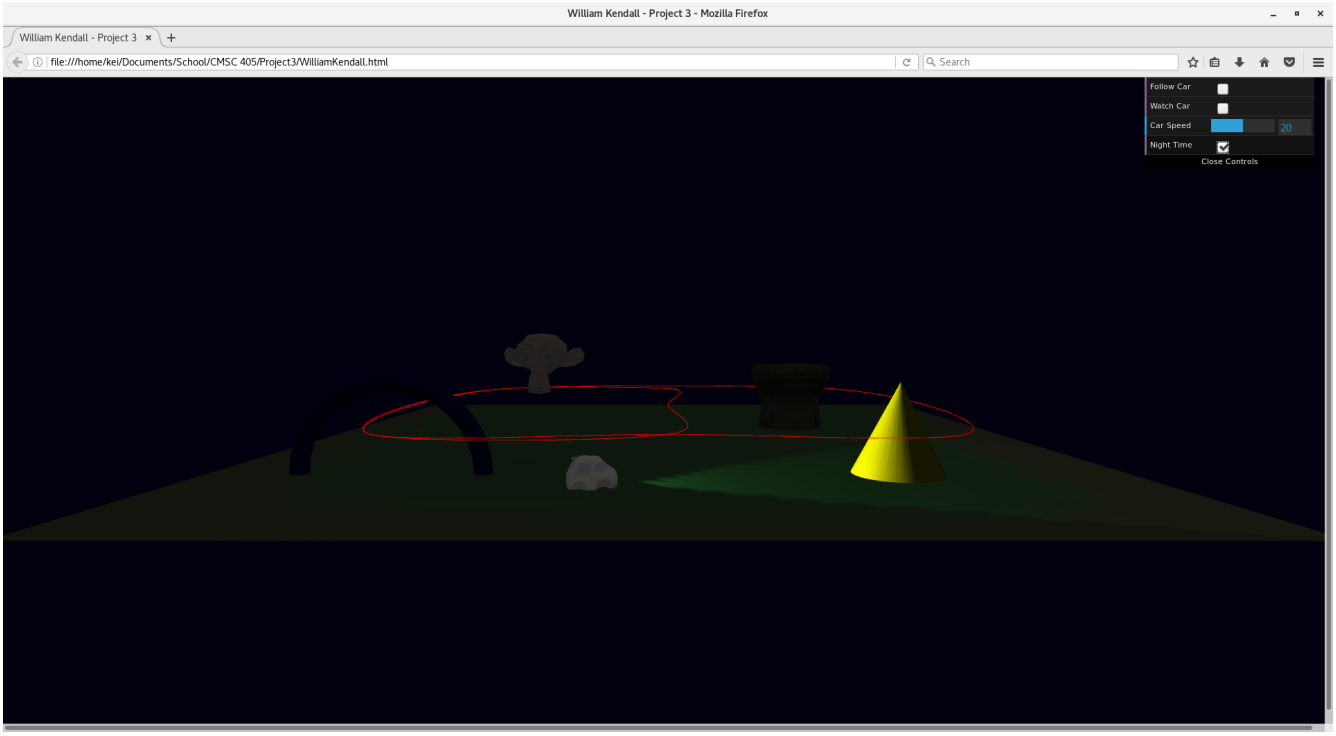
Kendall Land



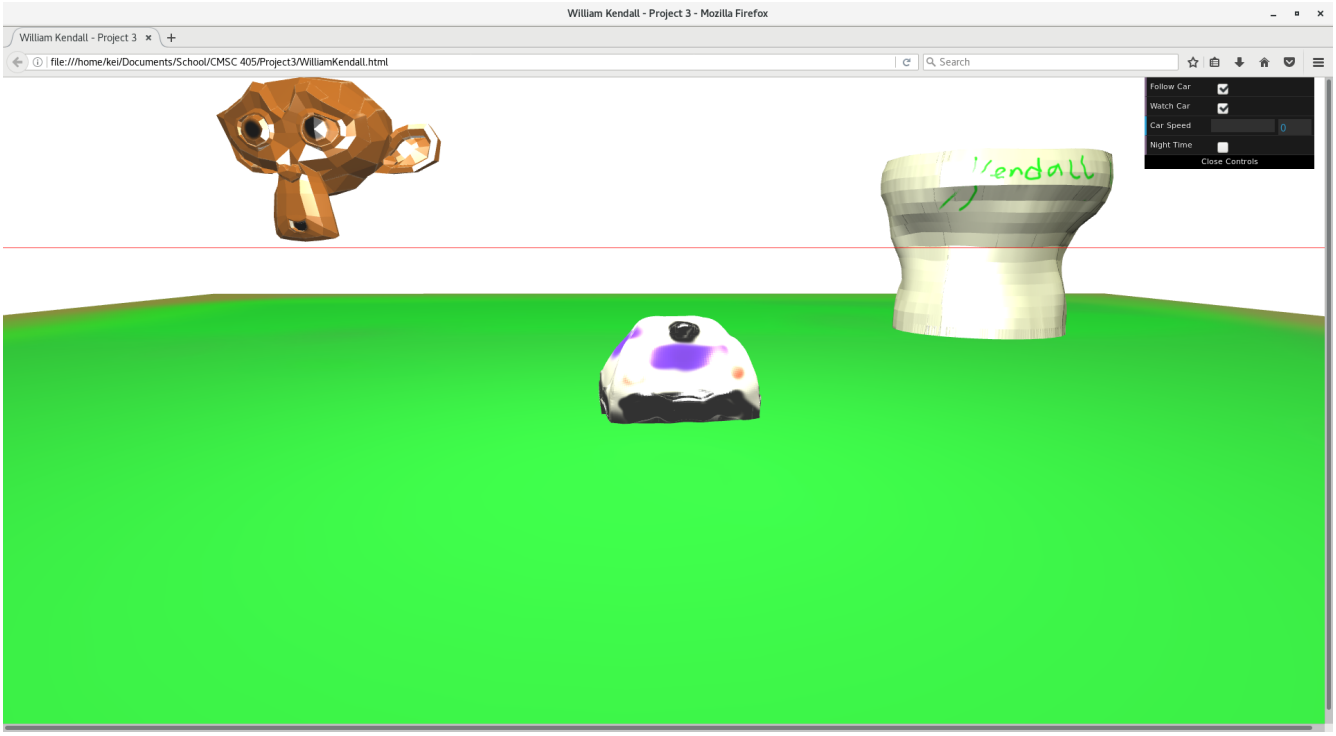
My original intention for this application was to design everything in blender including the animation. I have no really knowledge of web programming or blender. After a little research, I found that three.js could load blender models. I started with this approach until I ran into problems, which where my lack of knowledge of blender and animations. For some reason, the animations that I designed would not export. I was able to get skeletal animation to export, however I was not able to run them in three.js. I think this is due to me doing something wrong in blender.



After many problems with getting the models to load, such as not being able to find the location of the models, to texture problems, determination pays off. The three objects in Kendall world that are not from blender are the torus, cone, and the red path line. I added the monkey from blender to just show that I was using blender. Since I was not able to get animations to export from blender, the alternative was to hand code animations. Being that it is not really interesting to have hard coded animations. I made a path that could be changed to anything and the car will follow it. This is the point of having the red line to show the movement of the car.



Hey, what happened? Did someone turn out the lights? Yes.
It is at this point after loading all the objects and getting the car (that has no wheels) to load, I needed to make a scene that the users could change. I have seen some examples of three.js that have a really cool looking control panel. As luck would have it, this panel comes with three.js. And is called dat.gui.js
This was just what I needed, Dat Gui can update variables by itself. I used this for things like the car speed, I added a variable to the gui, set the limits and the gui updates the variable when the user changes it.



Its my horrible attempt at making a water tower, sorry I don't know blender. I also made the car and painted all the textures (monkey included).

The follow cam was one of the hardest (everything about this project was hard) things to get working. I have a path that the car follows, so I was cheap and just made the camera follow the same path. The reason this was difficult was because of my lack of knowledge on three.js about moving object and some bugs I was making on the way. I don't do any javascript programming so this was really a first.

Test Plan

Objective	**Attempt**	**Pass/Fail**	**result**
Three.js loads	1	Pass	Three.js works out of the box
Three.js loads	2	Fail	After trying out thing, I needed a new version of three.js
Three.js loads	3	Pass	Bugs in three.js are fixed in new versions
Models load	1	Fail	Unable to load local models
Models load	2	Pass	Changed models to Base64 and loaded as a dataURI
Models load	3	Pass	Models fixed in blender, resource location changed models load as files
Animations work	1	Fail	Something I am doing wrong, just dont know what it is.
Animations work	20000+	Fail	I give up
Animation work 2	1	Fail	Car flys around like crazy

Animations work	2	Pass	Researched about how three.js moves with time across paths. Found a cool function lerp.
Animations work	3	Fail	If the program lags, the delta is huge the math gets funny
Gui	1	Pass	Dat.gui.js works out of the box, this thing is great.
Car Headlights	1	Fail	Almost the same issues all over again as the path for the car
Car Headlights	2	Meh	Attached the light and target to the car.

Lessons Learned:

This project was all about NEW NEW NEW; well its all new to me. This is my first somewhat real javascript program, first time using blender, first time with three.js. I find that three.js is nothing like OpenGL and I have no idea what I am doing. The best lesson is that I was able to learn a little of everything; from making a cool 3D car model, texturing models, loading them in a web page, and making an interesting scene from them. Is it pretty? No, but it works.