# Project 1

OBJECT TRANSFORMATION

Mitchell Couturier | CIS 367 | March 14, 2017

Purpose

The purpose of this program is to create a few different 3D Objects and arrange them into a scene. In addition, examples of 3D transformations are animations are displayed as well. For this program, a Star Wars scene is displayed. The three objects used in this scene are droids, sand huts, and randomly positioned rocks. The droids in the scene are meant to resemble the BB-8 droid as seen in the Star Wars movies.

Architecture

This program and the graphics generated by it are created using WebGL. WebGL is a Javascript library used to create and draw computer graphics on a canvas. In this program, the render.js file contains the method *main()* which creates and displays the Star Wars scene. The render.js program uses the three classes **Rock**, **Hut**, and **BB8** which represent the 3 objects displayed in the scene. These classes also use many other classes that define and draw basic shapes. The exact dependencies can be seen in the diagram below.
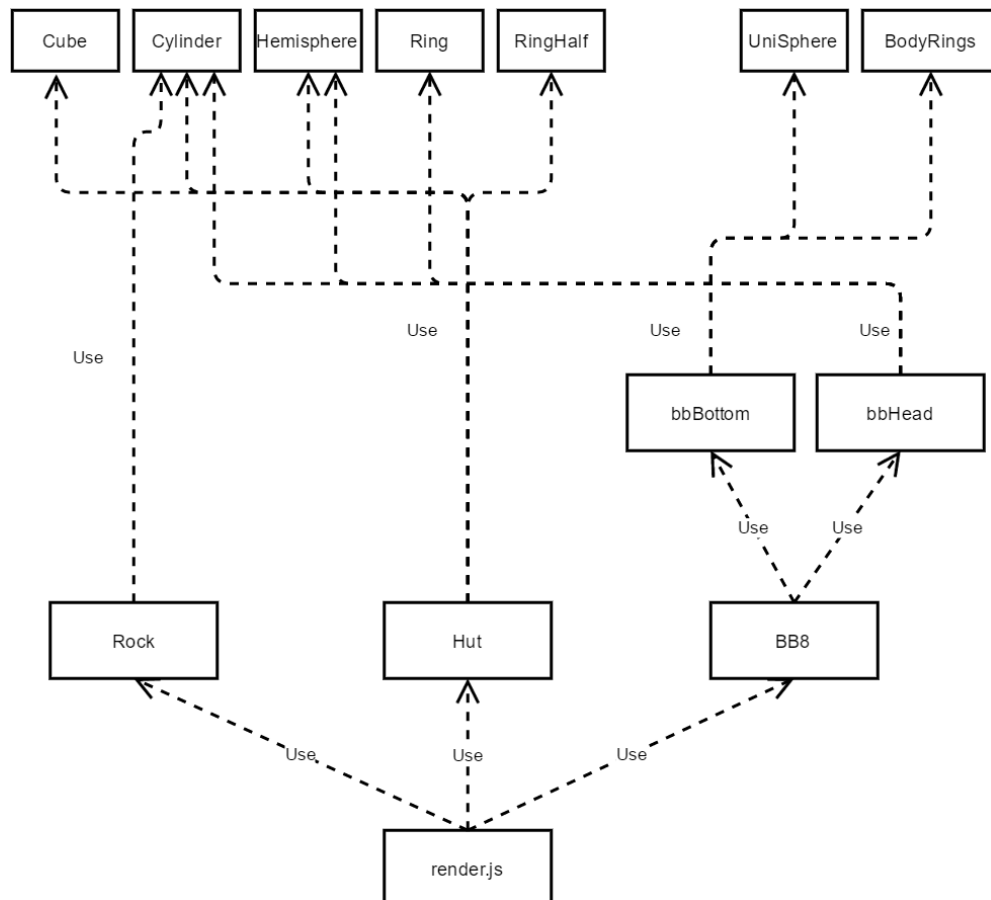
Building the BB-8 Droid

The BB-8 droid object is broken up into two parts, which are then each broken up into even more parts. The droid is first divided into a head object and a body object. The reasoning behind this initial step is to allow separate movement of each part, which will be described later in this document.

The head of the BB-8 droid is made up of multiple sections. The largest and first section is a hemisphere object, which makes up the bulky portion of the droid head. Underneath that, is a very thin cylinder of the same radius as the hemisphere, to increase the height of the head by a small amount. Underneath that cylinder is an upside-down truncated cone which serves as the bottom/base of the head, where the head of the droid connects to the body. On top of the head are two tall and thin cylinders which represent the antennas for the droid. And lastly, two ring objects are used toward the middle of the head object to represent the eyes.

The body of the BB-8 droid is much simpler, and is only made up of two parts. The first part is the bulk of the body, which is just a sphere object. The second part is the body designs, or rings, which cover the outside of the sphere. These rings are still rounded/angled so as to cover the surface of the sphere.

The head object is then mathematically placed just on top of the cylinder, so that the two objects become connected. The BB8 class takes a parameter for the radius of the body's sphere, and the head's height its adjust accordingly so it is placed at the top of the sphere. This is shown in the scene where there is a medium-sized and small-sized droid.

Building the Sand Hut

The sand hut object is much simpler to build, and is made up of four different types of shapes. The first portion of the hut is a hemisphere, which upper body of the hut itself, and makes up most of the mass of the object. Underneath the hemisphere, there is a truncated cone, which serves as the bottom-half of the hut. The door of the hut is then made up of two shapes, a half ring and two cubes. The half ring is the top of the door, and the cubes serve as the sides of the entrance. The cubes must be stretched to be more rectangular to match the length and height of the half ring. The door is then placed at the edge of the hut body to create the final product.

Building the Rocks

The rocks are the most simple objects, and are only made up of 4 cylinders. The cylinders are sized, rotated, and positioned together to create a seemingly random clump, in order to resemble a rock which would commonly be seen in a desert landscape.

Randomizing the Rocks

The rocks displayed in the scene are randomized by position, orientation about the z-axis, and size. First, an array of rocks is created, each with a random size. Next an array of 4x4 matrix transformations is created, each with a translation with random X and Y position and a z-rotation of a random angle. Before each rock is drawn, it's transformation is applied, creating the effect of multiple rocks randomly skewed along the ground.

New Basic Shapes Created

In order to create some of the objects in this Star Wars scene, many basic shapes had to be used. While many shapes were initially supplied for this project, some new shape classes had to be created. These new shapes included the hemisphere, half ring, and body rings. The hemisphere object was a class that was pulled over from the previous project, which started out as a sphere, and was simply cut in half to only draw half of the sphere. The half ring object was created the same way, by only creating half of the vertices of the ring and slightly adjusting how it is drawn. The body ring, similarly, was created by only drawing one layer of a multi-layer hemisphere, to instead create an angled ring. Multiple instances of this ring are created, and rotated around the center of the coordinate frame to create the full ring design for the body.

Head Movement

The droid has a head movement that occurs automatically, or independently from the user's input. This movement is to simulate the droid constantly "looking around" the scene. This movement is generated by rotating the head about the z-axis back and forth. Each time the head changes direction, the time for the head to rotate is randomized, and it does not change direction again until that time has been reached. In addition, to have the head pause before it switches direction, the head only rotates during the middle 50% of the rotation time.

Body Movement

The body's movement was a bit more complicated, and was totally reliant on the user's input. Keyboard input allows the body to roll around the scene in the direction specified by the user, with the head remaining on top, so that the entire droid can smoothly move around. This movement is broken up into two parts: position changes and rotation. First, when a user wants the body to move in a specific direction, a translation is applied to the entire object in that direction. While a keyboard button is held, the object will continue moving in that direction. Directions can also be combined, to create a diagonal

movement. Second, the body section of the droid will rotate in the direction of the movement. Since multiple rotations about an object's X and Y axes will not always result in rotations in the desired direction, matrix multiplication had to be done to account for this. To solve this, any time a direction is changed, a new vector which the object should be rotated around is created. This vector is created by creating a vector on the global axis and multiplying it by the inverse of the object's coordinate system. This new vector will allow the body to always roll in the direction that the user intended. Once again, this rotation along with the position changes create a rolling effect as seen in the Star Wars series.

Screenshots