# Apples to Oranges: Applications of Computer Graphics Methods

Esther Choe ‘22
Advisor: Lyn Turbak

## Abstract

In Spring ‘21, I took CS 307 (Computer Graphics) with Professor Ellen Hildreth. My final project was inspired by the phrase “comparing apples to oranges” and is comprised of a basketball game scene with apple and orange players, along with bunny spectators. The code involved working with the JavaScript library Three.js and the following graphics concepts were used:

- hierarchical modeling
- material, lighting, and shading
- curved lines/surfaces
- textures and texture-mapping
- animation

## Concepts

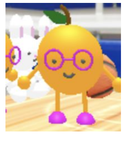
- **Three.js geometries** - sphere, torus, circle, cylinder, box
- **Bezier curve** - freeform curve created with 16 control points and based on parametric equations
- **Hierarchical modeling** - constructing individual geometry components and adding to a main object to manipulate
- **Cloning** - duplicates existing object, can manipulate separately once assigned to variable
- **MeshPhongMaterial** - affected by lighting by showing shine and specular highlights on its surface (unlike MeshBasicMaterial)
- **Texture-mapping** - array of pixels onto a plane
- **Ambient lighting** - lights all objects in scene uniformly, no shadows present
- **Directional lighting** - lights scene in a specific direction, casts shadows
- **Animation** - using several frames to create a dynamic scene
- **User interaction** - keyboard callbacks, mouse movement

### Overall scene

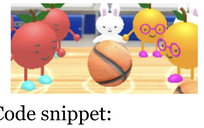
### Texture mapping on different surfaces

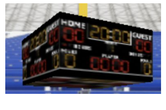
### Lighting (aerial view)

### Hierarchical modeling & Bezier curves



### Animation





→ Code snippet:

```
animationState.time += ballParams.deltaT;
// screen
screen.rotateY(0.01);
// ball movement vertically
animationState.ballHeight = setBallPosition(animationState.time);
// ball movement horizontally
ball.position.x += ballParams.motion;
orange1.position.z += orangeMotion;
if (ball.position.x >  60) {
    ballParams.motion = -1; // move left
    orangeMotion = -0.25;
}
if (ball.position.x < -100) {
    ballParams.motion =  1; // move right
    orangeMotion = 0.25;
```

(keyboard callbacks, e.g. ‘g’ key to start the animation and increase the speed of the bouncing ball, were implemented as well)

## Implementation

1. The floor, bleachers, and ceiling were produced from texture-mapping.
2. The basketball, hoop, and stand involved texture-mapping on geometry objects with Phong material. The hoop’s stand was created from three cylinders, where the base used two different radii.
3. The bunny, apples, and oranges were made with hierarchical modeling using Phong material. The geometry objects were mostly spheres that were scaled. The brown stem was made from Torus geometry and the green leaf was a Bezier curve of 16 points.
4. Ambient and directional light lit the overall scene.
5. The animation consists of three moving objects. The basketball is dribbled on the floor between the apple and the orange. The orange moves in the z-direction, bouncing the ball back to the apple. The scoreboard rotates around the y-axis.

## Applications

Computer graphics methods reflect many core ideas in computer programming. The following are some examples to illustrate this:

- Hierarchical modeling, where individual smaller geometries can be layered together to create a main component, is related to object-oriented programming, like inheritance. This may involve the use of different functions that convey the layers within a graphics scene, such as creating a new THREE.Object3D().
- Multivariable calculus and linear algebra concepts reoccur through the use of parametric equations, positioning of objects according to the axes, and mapping animation sequences to pi in relation to time.
- User interaction/experience is represented through taking into account the camera setup and positioning of the screen, as well as keyboard callbacks to interact with the animation.