

GingerBread Winter Scene

Gabriel Etim

gabriel.etim001@umb.edu

University of Massachusetts Boston



Figure 1: Gingerbread house and tree.

ABSTRACT

In this project, I create a winter scene with a customizable gingerbread house and Christmas tree. Raycasters were used to place candy of different colors on Three.js geometries.

KEYWORDS

WebGL, Three.js, landscape, geometry, winter, Christmas

ACM Reference Format:

Gabriel Etim. 2022. GingerBread Winter Scene. In *CS460: Computer Graphics at UMass Boston, Fall 2022*. Boston, MA, USA, 2 pages. <https://CS460.org>

1 INTRODUCTION

In the spirit of Christmas and winter, I sought to create a scene of the Christmas related objects like a gingerbread house and christmas

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

CS460, Fall 2022, Boston, MA

© 2022 Copyright held by the owner/author(s).

ACM ISBN 1337.

<https://CS460.org>

tree that could be decorated with Three.js's geometrical shapes. I wanted to be able to allow the user to pick a variety of designs and use their creativity to make their own unique landscape.

2 RELATED WORK

WebGL, Three.js, Javascript

3 METHOD

My project involves a brown gingerbread house and a green Christmas tree in a snowy winter environment. Both the Christmas tree and the gingerbread house can be decorated with candy by holding down shift or ctrl and moving the mouse. The former gives a gumdrop while the latter is icing. The control panel on the right-hand side of the screen is used for control of the camera and the candy color. The camera can also be controlled using the arrow keys as well as Q and W.

3.1 Implementation

I implemented the decoration feature using raycasters that determine how the camera and mouse intercept certain geometries in the scene.

```
window.onmousemove = function(e) {  
    if (e.shiftKey) {  
  
        pixel_coords = new THREE.Vector2(  
            e.clientX, e.clientY );  
  
        vp_coords = new THREE.Vector2(  
            ( pixel_coords.x / window.innerWidth ) * 2 - 1,  
            ( pixel_coords.y / window.innerHeight ) * 2 + 1 );  
  
        vp_coords_near = new THREE.Vector3(  
            vp_coords.x, vp_coords.y, 0 );  
  
        raycaster = new THREE.Raycaster();  
        raycaster.setFromCamera(vp_coords_near, camera);  
        intersects = raycaster.intersectObjects(A);  
        geometry = new THREE.SphereGeometry(5, 32, 16);  
        material = new THREE.MeshLambertMaterial(  
            { color: controller.threejs_color } );  
        snow = new THREE.Mesh( geometry, material );  
        scene.add(snow);  
        snow.position.set(intersects[0].point.x,  
            intersects[0].point.y, intersects[0].point.z );  
    }  
}
```

3.2 Milestones

How did you structure the development?

3.2.1 *Milestone 1.* Adding lighting and plane. Looked for possible textures and colors for all objects

3.2.2 *Milestone 2.* Experimented with different lighting arrangements to get colors correct.

3.2.3 *Milestone 3.* Experimented with both trackball and orbit cameras, wrote code to move the camera around using GUI and keyboard as well as the mouse.

3.2.4 *Milestone 4.* Created gingerbread house using boxGeometry and rotations.

3.2.5 *Milestone 5.* Wrote code to create icing with mouse drag and added a second candy shape.

3.2.6 *Milestone 6.* Created Christmas tree using cone Geometries and cylinder Geometries.

3.3 Challenges

- Challenge 1: One of the reasons I had trouble adding more candy shapes is because the orientation of each candy would be upright regardless of what house wall it was placed on, making it look awkward and out of place. I tried to add a rotation factor that changed depending on what house surface the candy was placed on and I found the options for manipulating raycaster.intersectObjects to be very limited.
- Challenge 2: I had trouble getting the lighting of the scene correct. I could not add a triangle geometry as easily as I

could a box geometry and the method that did eventually work did not reflect light properly. I had to use MeshStandardMaterial for the house to have a consistent color.

4 RESULTS

I was able to create a 3D structures that could be changed to the liking of the user with multiple colors and shapes.

<https://gabrieletim001.github.io/final/index.html>

5 CONCLUSIONS

Overall there were many things that I wish I had still gotten to add to this project to make it more interesting and better reflect the concepts I know how to implement. Still, I enjoyed working on it and troubleshooting the problems I encountered. I think in the future I would add a lot more motion to make the area feel more alive and interesting.

REFERENCES