Voxel Placer

Ryan Dang
Ryan.Dang001@umb.edu
University of Massachusetts Boston

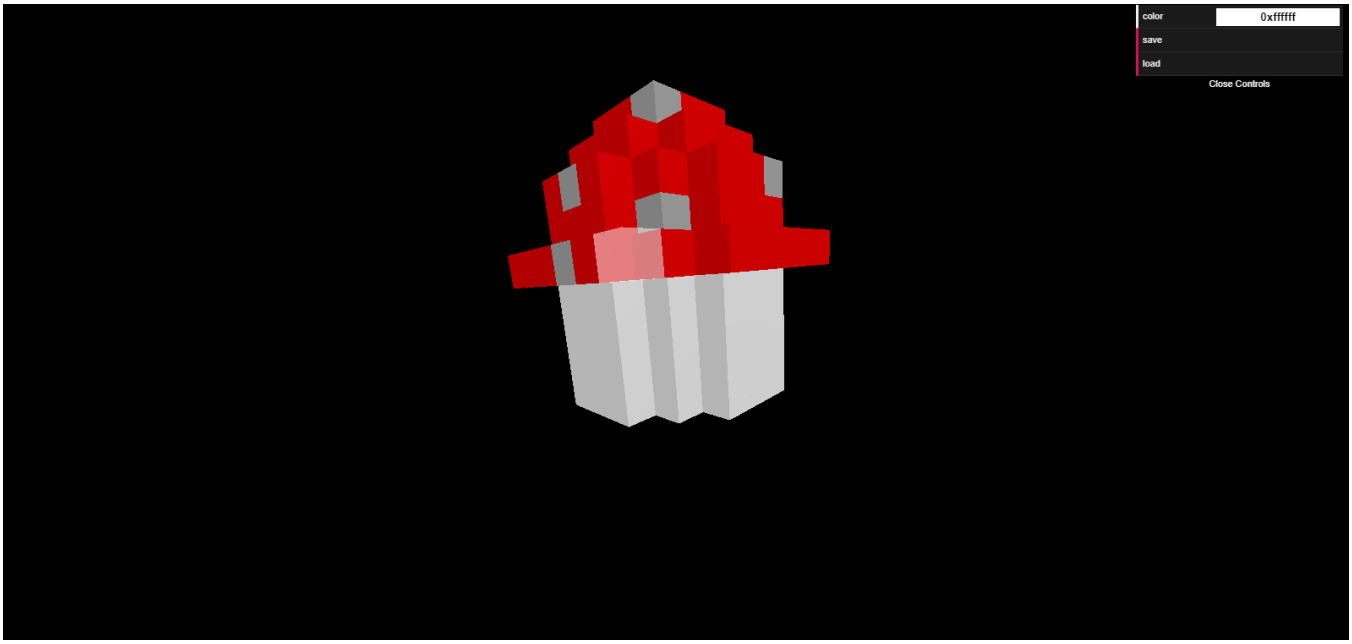

Figure 1: Voxel Mushroom created with the program

ABSTRACT

This program aims to create a simple web-based voxel creator, it allows users to create simple voxel geometry. It also allows saving and loading with a txt file.

KEYWORDS

WebGL, Visualization, 3D Model, Voxel

ACM Reference Format:
Ryan Dang. 2019. Voxel Placer. In *CS460: Computer Graphics at UMass Boston, Fall 2019*. Boston, MA, USA, 2 pages. https://CS460.org

1 INTRODUCTION

This project is important because it gives users the ability to quickly create a quick model without using any sort of external software.

CS460, Fall 2019, Boston, MA

ACM ISBN 1337.
https://CS460.org

2 RELATED WORK

This project was created with the use of Three.js, it also used dat-gui for the menu options. It also used FileSaver.js library for saving strings into txt files.

3 METHOD

The program uses Three.js to handle the rendering, and it uses the raycasts in Three.js to determine where to place the cubes. Once it fires a raycast it selects the second object it hits (this is becasue the cursor or "ghost" cube is also hit by the raycast and we want to ignore that). It then selects the face it is hitting and then places the cursor there so the user will know where the program intends to place the voxel. The user then can do shift+click or shift+right-click to add or remove a voxel. There is also a color option in the dat-gui menu.

3.1 Implementation

This section takes the raycast array of intersections and then checks if the object is the cursor, which it will skip to the next one and then it will select the next object. It also takes the face as well.

```
for (var i = 0; i < intersects.length; i++){
    if(intersects[i].object == cursorObject){
        continue;
    };
    selectedObject = intersects[i].object;
```

```
        objectFace = intersects[i].face;
        break;
}
```

The "ghost" cube or cursor is placed depending on which face normal it is hitting and the face normals will either be -1 or 1 depending on the axis and direction so we can just multiply it by a value to offset it from the selected object's position.

```
var newPosition = new THREE.Vector3(
     selectedObject.position.x + objectFace.normal.x * 10,
     selectedObject.position.y + objectFace.normal.y * 10,
     selectedObject.position.z + objectFace.normal.z * 10);
```

3.2 Milestones

How did you structure the development?

3.2.1 Milestone 1. Brainstormed ideas on ways of implementation (mainly how to deteremine where to place the cubes)

3.2.2 Milestone 2. Implemented the raycast selection + placing

3.2.3 Milestone 3. Implemented saving/loading

3.3 Challenges

Describe the challenges you faced.

- Challenge 1: Figuring out how to save/load files took some resarch, and I found an external library for saving.
- Challenge 2: Figuring out what could and couldn't be accessed from Three.js took some experimenting

4 RESULTS

The final result of the project is a simple to use website that allows voxel creations that can be saved and loaded from a txt file.

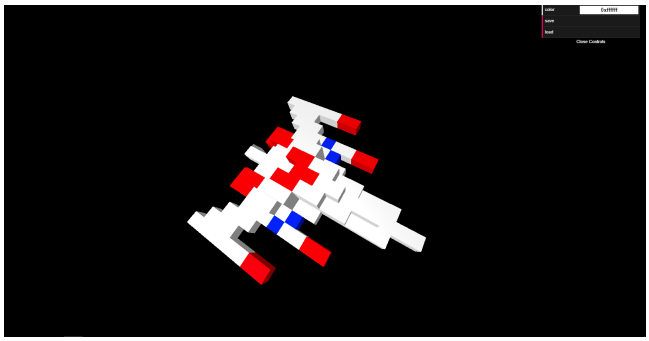

Figure 2: Galaga ship recreated

5 CONCLUSIONS

Overall the project came out well, a few things could be further optimized like the materials, currently with the way it's set up it will create a new material per cube even if they are using the same color so an improvement that could be made is optimization. The control scheme for the scene could be better as trackball controls may make it difficult to get the object in the position people would want.

Your references are loaded in BibTex from references.bib!

REFERENCES