Viktor Johan Kolind s204166

02561 Computer Graphics

Texturing wavefront files (.obj)

Introduction

In this report, we describe the process of adding textures to a wavefront model using the OBJPARSER.js file. The OBJPARSER.js file is responsible for parsing wavefront OBJ files and extracting the necessary information to render the model. By default, the OBJPARSER.js file does not support reading texture coordinates, so it must be extended to enable this functionality. We outline the changes made to the OBJPARSER.js file to enable it to read texture coordinates and the implementation of these changes in the textureObject.js program. The results of our implementation are then discussed, including any issues encountered and the impact on the displayed textures on the model.

Method

The way to add textures to the wavefront model is by extending the OBJPARSER.js file so it can read the texture coordinates, also known as the uv. It does not do this by default, so it must be extended in order to allow the data to be read by the shaders. The program also needs to have initialized a texture buffer so it can read the texture data.

Implementation: There are several minor changes to the parser that allow it to read the texture data. These changes are listed below:

- The OBJDOC object has a new field called "textures".
- The line parser has a new switch case that reads the token "vt", which stands for vertex texture. It is followed by two floats that are the texture coordinates, which are then pushed to the object's "textures" field.
- In the face parser, there is a new condition that counts the texture indices.
- In the function that splits quads into triangles, there is some code that helps split up the texture coordinates correctly.
- In the getDrawingInfo() function, a new float array has been initialized.
- The uv's are added to the correct vertex indices in the getDrawingInfo() function.
- In the face object, new values have been added for texture indices.
- The DrawingInfo object has a field for textures.
- A new object has been added for the texture object, which contains the uv's.

The program that calls the parser is textureObject.js, which is an extension of worksheet 5.3. The changes made to this program include the creation of a new buffer that takes the texture coordinates and sends them to the vertex shader with the attribute a_Textures.

Results

Unfortunately, the parser did not work for models more complex than a quad. There is a bug somewhere that causes it to be unable to load the uv's properly. In the following picture, there is a quad:

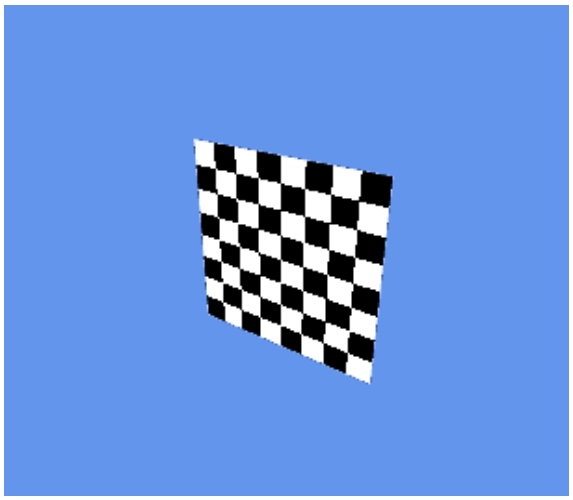

Figure 1 Quad

When the program read the texture data, it became unordered. As a result, the resulting model's textures were displayed incorrectly. For example, here is a cube:

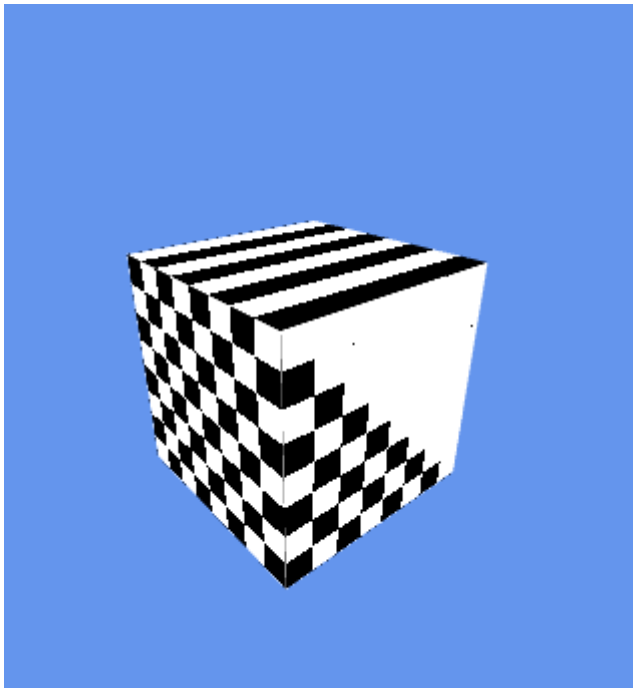

Figure 2 Cube

It is worth noting that both of these models had their uv's reset in Blender, so the correct way for the texture to be displayed would be flat on the model. However, this is clearly not the case.

Discussion

In conclusion, the OBJPARSER.js file was extended to read texture coordinates (also known as uv's) in order to add textures to the wavefront model. This was achieved through several

changes to the parser, including the addition of a new field for textures in the OBJDOC object and the initialization of a new float array in the getDrawingInfo() function to store the uv's. However, the implementation was not successful in loading the uv's properly for models more complex than a quad, resulting in incorrect textures being displayed on the model. Further investigation and debugging is needed to identify and resolve the issue.