

COMP3011 Computer Graphics

Assessment 3


Report Sheet (v5)


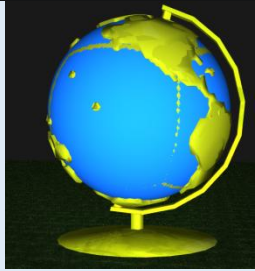
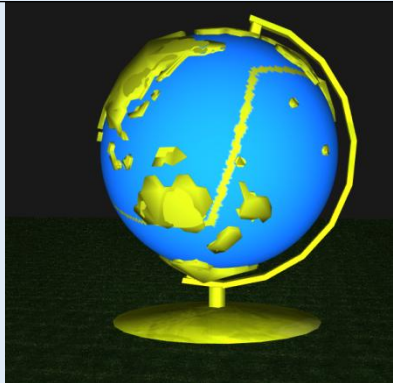
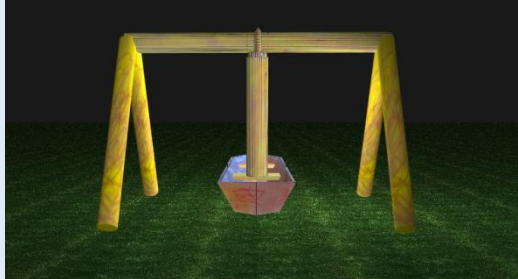
Use this table to help you prepare for your demo. You will also need to submit this report to Moodle.


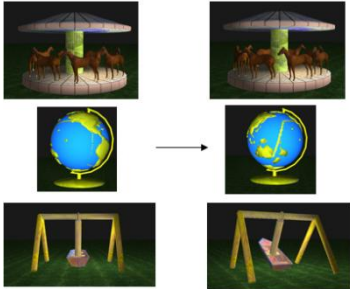


Student Name: Wendi Han

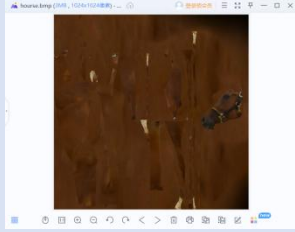

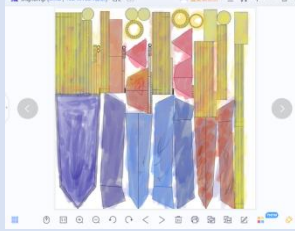
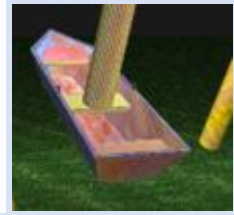




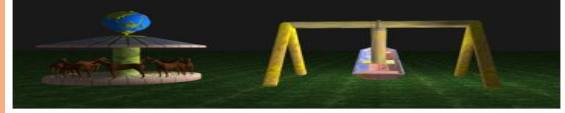
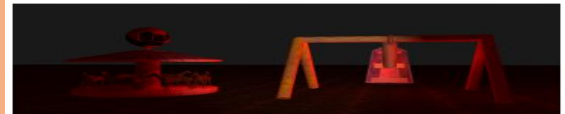
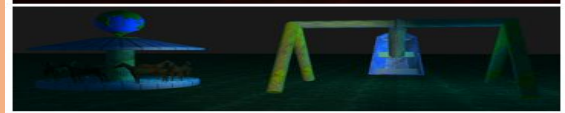
Student ID: 20126355

Username: scywh1

| | | |
|---|--|---|
| Introduction <i>Please explain the reason why you chose to implement this scene</i> | <i>Describe your inspirations</i> | <i>Provide a general description of the scene.</i> |
| <p>The first eligible object that comes to my mind is a windmill, however, I cannot find another object which is harmoniously shown in the same scene. Then I remember my favorite place from my childhood -- playground.</p> | <p><i>I used to play with my family in playground in my hometown. There are lots of movable facilities such as carousel and corsair ship. Also, a big fountain with a shiny rotate globe on the top. So, I want to implement it.</i></p> | <p><i>The whole scene simulates night and a positional light on the sky. The view contains a spotlight which simulate flashlight. There are a globe, carousel and corsair in the scene which can move or stop through key press. Also, a disco light can be shown by key.</i></p> |
| TR 1 – 3D modelling & 3D Transformations <i>Please give details for up to 3 objects</i> <i>Note – if you have programmed an API for parsing OBJ data then provide additional details on the last page</i> | | |
| Object 1 3D modelling <i>Please give a screenshot</i> | <i>reference specific code (filename and line)</i> | <i>Description of object</i> |
|  | <p><i>Code:</i> <i>obj/Carousel.cpp</i> <i>obj/Carousel.h</i></p> <p><i>Res:</i> <i>Bin/model/carousel</i></p> <p><i>(Parser see last part)</i></p> | <p>It contains two parts, one fixed bone on the pillar, shelter and base and one continually rotate seat (horse-like). The object is programmed from '.obj', '.bmp' and '.mtl' file. The file is modelling from blender.</p> |
| Object 1 3D Transformations <i>Please give a screenshot</i> | <i>reference specific code (filename and line)</i> | <i>Description of transformations</i> |

| | | |
|--|--|--|
|  | <p><i>Main.cpp</i> <i>Line 85-87 - cal the time different between each loop</i></p> <p><i>obj/Carousel.cpp</i> <i>Line 20 - cal angle base time</i> <i>Line 36 - rotate</i></p> | <p>It is clear that horses' position is different from last part. Therefore, I use rotate to change it position and keep it rotating. Its position is translated to place in the same scene with other objects. It is scaled to be harmonious with other objects. Also, it can be interacted by pressing '9' to stop/re-rotate it.</p> |
| <h3>Object 2 3D modelling</h3> <p><i>Please give a screenshot</i></p> | <p><i>reference specific code (filename and line)</i></p> | <p><i>Description of object</i></p> |
|  | <p><i>Code:</i> <i>obj/Globe.cpp</i> <i>obj/Globe.h</i></p> <p><i>Res:</i> <i>Bin/model/Globe</i></p> <p><i>(Parser see last part)</i></p> | <p>It contains two parts, one fixed bone on the middle pillar and base and one continually self-rotate ball. The object is programmed from '.obj', '.bmp' and '.mtl' file. The file is modelling from blender.</p> |
| <h3>Object 2 3D Transformations</h3> <p><i>Please give a screenshot</i></p> | <p><i>reference specific code (filename and line)</i></p> | <p><i>Description of transformations</i></p> |
|  | <p><i>Main.cpp</i> <i>Line 85-87 - cal the time different between each loop</i></p> <p><i>obj/Globe.cpp</i> <i>Line 20 - cal angle base time</i> <i>Line 36 - rotate</i></p> | <p>Globe is self-rotate to change it direction seen bone as center (image different from last part). Its position is translated to place in the same scene with other objects. It is scaled to be harmonious with other objects. Also, it can be interacted by pressing '9' to stop/re-rotate it.</p> |
| <h3>Object 3 3D modelling</h3> <p><i>Please give a screenshot</i></p> | <p><i>reference specific code (filename and line)</i></p> | <p><i>Description of object</i></p> |
|  | <p><i>Code:</i> <i>obj/Corsair.cpp</i> <i>obj/Corsair.h</i></p> <p><i>Res:</i> <i>Bin/model/Corsair</i></p> <p><i>(Parser see last part)</i></p> | <p>It contains two parts, one fixed bone - left/right/top pillar and one continually rotate boat with pillar connected it with bone. The object is programmed from '.obj', '.bmp' and '.mtl' file. The file is modelling from blender.</p> |
| <h3>Object 3 3D</h3> | <p><i>reference specific code (filename and line)</i></p> | <p><i>Description of transformations</i></p> |

| | | |
|--|--|--|
| <h2>Transformations</h2> <p>Please give a screenshot</p> | | |
|  | <p><i>Main.cpp</i> Line 85-87 - cal the time different between loops</p> <p><i>obj/Corsair.cpp</i> Line 21,34 - cal angle base time Line 40 - rotate</p> | <p>One pillow and ship are rotate to change its position Its position is translated to place in the same scene with other objects. It is scaled to be harmonious with other objects. Also, it can be interacted by pressing '9' to stop/re-rotate.</p> |
| <h2>TR 2 - Animation</h2> <p>Please give a screenshot</p> | | |
|  | <p>(Objects transformation have been shown in previous part.) <i>Scene.cpp</i> Line 104-109,137-140 , change a value "moveOn" to decide the value of transformation and thereby, stop or restate it.</p> | <p>The object is keep transforming over time. And key '9' can be used to stop/re-start its transformation.</p> |
| <h2>TR 3 – Lighting</h2> <p>Please give details of up to 2 lights</p> | | |
| <h3>Light 1</h3> <p>Please give a screenshot</p> | <p>reference specific code (filename and line)</p> | <p>Description of light 1</p> |
|  | <p><i>Bin/shader/model.fs</i> Line 82-100 <i>Scene.cpp</i> Line 70-77</p> | <p>Spotlight from view simulate flashlight.</p> |
| <h3>Light 2</h3> <p>Please give a screenshot</p> | <p>reference specific code (filename and line)</p> | <p>Description of light 2</p> |
|  | <p><i>Bin/shader/model.fs</i> Line 66-80 <i>Scene.cpp</i> Line 79-83</p> | <p>Directional light from top to ground. Can press '0' to change it color.</p> |
| <h2>TR 4 – Texture</h2> <p>Please give details of at least 2 textures</p> | | |
| <h3>Texture 1</h3> <p>Please give a screenshot</p> | <p>reference specific code (filename and line)</p> | <p>Description of texture</p> |

| | | |
|---|--|--|
|   | <p>Main: Obj/Carousel.cpp -> Draw() FrameWork/TTexture.h/.cpp Relate: FrameWork/TMaterial.h/.cpp FrameWork/TObjLoader.h/.cpp FrameWork/TMesh.h/.cpp FrameWork/TShader.h/.cpp FrameWork/TBaseShader.h/.cpp</p> | <p>Obj->Mtl->Text Load the texture for object in .bmp file. All the objects use texture in my project. The detail of loading it based on its coordinate will neglect here since it is repeated with last part. (also grass ground). Mipmap is using for all texture.</p> |
| <h2>Texture 2</h2> <p>Please give a screenshot</p> | <p>reference specific code (filename and line)</p> | <p>Description of texture</p> |
|   | <p>Main: FrameWork/TTexture.h/.cpp Obj/Carousel.cpp -> Draw() Relate: Same as previous</p> | <p>Obj->Mtl->Text Load the texture for object in .bmp file. All the objects use texture in my project. The detail of loading it will neglect here since it is repeated with last part.</p> |
| <h2>TR 5 - Interactive camera</h2> <p>Please give a screenshot</p> | <p>reference specific code (filename and line)</p> | <p>Description of interactive camera</p> |
|     | <p>Scene.cpp Line - 28,29</p> <p>FrameWork/TCamera.cpp FrameWork/TCamera.h FrameWork/TFreeCamera.cpp FrameWork/TFreeCamera.h</p> | <p>Two camera, perspective view and orthographic view, change using 'LEFT_SHIFT'. 'WASD' -> front, left, back ,right 'LEFT_CTRL' -> down 'SPACE' -> up</p> |
| <h2>TR 6 - Interactive object</h2> <p>Please give a screenshot</p> | <p>reference specific code (filename and line)</p> | <p>Description of interactive object</p> |
|    | <p>(stop/restate transform has been shown in TR2)</p> <p>Bin/shader/model.fs Line 51-64 Scene.cpp Line 52-56 change the value of 'pointLight' to change its color. Used in Line 63-68</p> | <p>Use '9' to stop/restart objects' motion. Use '0' open/close disco light (change the color of proportionate light). Also, you can use "v" to show your cursor.</p> |
| <h2>Conclusion</h2> <p>Please describe what you perceive to be the strengths and weaknesses of your project</p> | <p>Describe what aspect of it you are particularly proud of, and what you think would need to be improved.</p> | <p>Reflect on what you have learned during this project that you can apply in future projects to improve your performance.</p> |
| <p>Nearly all the objects using textures maybe one advantages of my project, they make objects seems better. Also, instead of using api such as 'assimp' and 'tinyobj', I implement an object parse to do that job which is time-consuming but</p> | <p>The proudest thing is implementing object parser by myself and refactoring the mess file to separate each function to different class. These are</p> | <p>I may not implement another object parser in the further, but I think it is helpful for me to understand the workings</p> |

feel proud when I finished it. The weakness is the number of objects is not enough compared with a real playground. Also, at the first time, I want to implement an island surrounded by ocean but there is no time to do that.

difficult and cost lots of time to implement and debug, need to see lots of resources, but feel proud and relax when I finished it. The things need to be improved is add more facilities and change the ground to island surrounded by ocean (do not have enough time to do that).

inside these APIs. Also, mastering some basic knowledge of blender.

Time Consuming:

Nearly One week for modelling – blender to generate obj bmp mtl.
 One week for basic – window, vao, shader, ...
 One week for another things – camera, light, some interaction and modify previous code (i.e. add function, etc)
 Two weeks for programming the objects – parser, texture (Mesh, Model, Material, Texture, ...)
 All the time include debugging and some tasks interact are interspersed with each other, such as basic window, shader may be done at the same time I have started to do the objects parser.

TR 1 – 3D modelling & 3D Transformations

OBJ parser

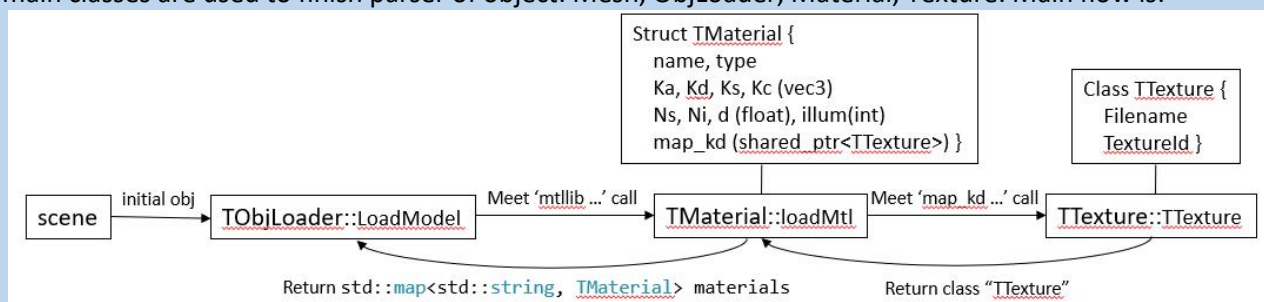
Please give details for how you programmed the OBJ parser function

Include all of the important aspects including

1. Memory allocation and freeing
2. Data structures you programmed (remember you can only use API which I provided in the course)
3. How you read the file
4. Logic for parsing the vertex attributes
5. Transferring the vertex attributes to OpenGL

Basically, it contains three types of files, .obj, .mtl, .bmp which contain one object's texture appearance, position of each point, shape and material of texture.

Four main classes are used to finish parser of object: Mesh, ObjLoader, Material, Texture. Main flow is:



LoadMtl and TTexture contain read file “.mtl” and “.bmp”. For material file, main loop is read line one by one and split it. After splitting, the first string is its type and according to the type of line read its content saved in TMaterial. For TTexture, it generates and save texture Id () and bind it. Then, define repeat and linear filtering and convert bmp to rgb. Finally, specify 2d texture using glTexImage2D(). Therefore, we finished the processing of ‘.bmp’ and ‘.mtl’ file and we return its data “materials” to “LoadModel” method.

The memory here is read file content will free when ttexture class deconstruct and reading material we do not save its content (except line data using unique_ptr and data need to be returned) as string pointer.

Next step is to parser ‘.obj’ file. Similarly, a figure as bottom:

Temperate variable:
 vector<vec3> vList
 vector<vec2> vtList
 vector<vec3> vnList
 vector<Vertex> currVertices
 Vector<unsigned int> currIndices
 Tmaterial currMaterial
 map<string, TMaterial> materials

Struct Vertex {
 vec3 Position;
 vec3 Normal;
 vec2 TexCoords;
 vec3 Tangent;
 vec3 Bitangent;}

Struct ObjParsedLine {
 token (line type), strData,
 vec2Data, vec3Data,
 face (Face)}

Struct Face {
 indices (vector<int>)
 textures (vector<int>)
 normals (vector<int>)}

TObjLoader::LoadModel

1. Read each line, call processLine()
2. Analyze ObjParsedLine's token:
 - mtllib -> materials = TMaterial::loadMtl()
 - usemtl -> currMaterial = materials[mtlName]
 - v/vn/vt -> vList/vnList/vtList.append(Data)
 - f -> polygon to triangle, select point and add it
 - AddPoint()**
 - a) create Vertex v based on parser data (includes index points to vList/vnList/vtList).
 - b) Get vec from vList/vnList/vtList, save in v.
 - c) Add v in currVertices
 - Add its indices in currVertices to currIndices
 - o -> a. Create TMesh (currVertices, currIndices, currMaterial)
 - b. save TMesh into meshes (vector<TMesh>) for further draw

Read each line and analyze

Get struct ObjParsedLine

processLine:

1. if not empty, spilt and save first string as token
2. Analyze token:
 - #, s -> return
 - mtllib, usemtl -> strData = 2nd string
 - o, g -> strData = 2nd string
 - v, vn -> vec3Data <2nd string, 3rd string, 4th string>
 - vt -> vec2Data <2nd string, 3rd string>
 - f -> a. loop each string
 - b. split using '/'
 - c. based on its format (i, i/t, i/t/n, i//n)
 - push value into indices/textures/normals

TMesh::Tmesh

1. Per adjascent three points in currVertices is a triangle, so loop currIndices / 3 times to assign index to line_indices
2. Setup Mesh: bind buffer, VAO, VBO, EBO, specify location and data format
3. setup Line: similar to last last step.

```
line_indices.resize(indices.size() * 3);
for (int i = 0; i < indices.size() / 3; ++i)
{
    unsigned int i0 = indices[i * 3 + 0];
    unsigned int i1 = indices[i * 3 + 1];
    unsigned int i2 = indices[i * 3 + 2];

    line_indices[i * 6 + 0] = i0;
    line_indices[i * 6 + 1] = i1;
    line_indices[i * 6 + 2] = i1;
    line_indices[i * 6 + 3] = i1;
    line_indices[i * 6 + 4] = i2;
    line_indices[i * 6 + 5] = i0;
}
```

I believe above figure can explain clearly how to parser a '.obj' data. The process of reading file using "unique_ptr" so do not need to be freed manually. The other data each line in the file saved in the buffer in VAO, VBO, EBO. GlfwTerminate() will clean it.

The stream_using unique_ptr to create, can release its memory by defining deleter to close the file stream and delete

```
unique_ptr<ifstream, void(*)(ifstream *)> pIfs(new ifstream(mtlFileName), [](ifstream *ifs)
{
    ifs->close();
    delete ifs;
});
```