COMP3011 Computer Graphics

Assessment 3

Report Sheet (v4)

Student Name: Ziqi Yang

Student ID: 16521586

Username: zy21586

|  |  |  |
| --- | --- | --- |
| Introduction | *Inspiration and Instruction* | *Architecture* |
| *Inspired from Game Gris, I implemented this scene. All models are created by myself (including textures).*  *Key features are:  1. Engine (windowSystem, InputSysytem, Renderer, Free-look Camera …);*  *2.Graphics (Dir/point/spot Lights, Skybox, OBJ Model Loader, Batch Rendering, HDR, Post-processing, MSAA…)*  *3. Scene (Player, Boids, Tilemap, Ripple effect…)* | *Inspired from Game Gris and my own Game (currently developing).*  It contains a player, boids and interactive tilemap.  *Instruction:*  *1. press P to summon fish*  *2. press F to switch camera*  *3. use mouse to free-look*  *4. scroll to zoom*  *5. press ESC to end*  *6. use WSAD to control the player movement* | *1. All systems init and update in Engine class.*  *2. All the entities (Entity.h), lights (Light.h) are generated in Sence class.*  *3. All rendering process happens in Render class including UBO (all view, projection and light info are pass to GPU in UBO), batch rendering, skybox, HDR and post-processing.*  *4. All model/mesh/textures are loaded by Model class.*  *5. All shaders are loaded by Shader class.*  *6. Entities are stored in graph structure (though I didn’t add sub-entity in my demo scene)* |
| TR 1 – 3D modelling & 3D Transformations  *Please give details for up to 3 objects* | | |
| Object 1 3D modelling  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of object* |
|  | *Relevant class/file:*  *class Player : public Entity*  *Transform.h*  *Scene.h*  *mesh.h*  *model.h*  *Shader.h*  *utils.h*  *Shader:*  *player.vs, player.fs*  *Model:*  *player.obj, player.mtl* | *I created this model in Blender and export as obj.*  *This model is texture-less; I coded necessary parameters in the mtl file and its shader for three meshes (hat, body and cloak). This object is affected by dirLight and HDR but not point lights on purpose (player.fs Shader)* |
| Object 1 3D Transformations  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of transformations* |
| 1. *Movement (WASD)* 2. *Rotation by lerp* 3. *Floating by lerp* 4. *Please try out in the app, the screenshot doesn’t work for this 😊*   *All transformation is updated recursively from parent to children (if any) in Transform class.* | *Relevant class/file:*  *class Player : public Entity*  *class Transform*  *InputSystem.h utils.h (lerp & pingpong)*  *(math::lerp())* | 1. The player is control by Keyboard WASD to move but only when the camera is fouced (press F to foucs). 2. The player can rotate smoothly by linear interpolation. 3. The player can floating smoothly by time. |
| Object 2 3D modelling  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of object* |
|  | *Relevant class/file:*  *class Tilemap : public Entity*  *mesh.h model.h*  *utils.h*  *Shader:*  *tile.vs,tile.fs*  *Model & texture:*  *tile.obj, tile.mtl, diffuse.png* | This object is implemented as particle system using batch rendering technique to draw all the tiles in one draw call from CPU to GPU. (set transformation matrices as an instance vertex attribute, see Tilemap.cpp line 38 and line 223)  The tile object is rendered dynamically and affected by dirlight, pointlights, HDR and bloom. |
| Object 2 3D Transformations  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of transformations* |
|  | *Relevant class/file:*  *class Tilemap : public Entity*  *Shader:*  *tile.vs,tile.fs* | This tilemap simulate a ripple effect by ripple algorithm (*see Tilemap::Ripple()* line 79)  The tile object is interactive with the player’s position.  The tile object is rendered according to a light level cubically related to its current height (y value) (see tile.vs line 40) |
| Object 3 3D modelling  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of object* |
|  | I’m not sure the skybox counts for object so I will introduce fish instead.  *Relevant class/file:*  *class Fish : public Entity*  *Shader:*  *fish.vs, fish.fs* | Similar as the player, the fish is loaded, rendered texture-less and affected by dirlight, HDR and bloom.  Each fish is bind with a pointlight rendered dynamically and passed using UBO. |
| Object 3 3D Transformations  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of transformations* |
| *Please try out in the app, the screenshot is similar as the above 😊*  *Press P to start play animation of the boids.* | *Relevant class/file:*  *Class Fish : public Entity*  *Fish::Move()*  *Fish::Turn()*  *Transform.h*  *utils.h (math::lerp())* | The boids can swim inside the zone of tilemap.  Each fish can turn direction and float up or down. The Turn() func is called by time interval to control the next rotation and floating;  we lerp its transform smoothly in each frame. |
| TR 2 - Animation  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of animation* |
| *Please see transformation sections from 3 objs above.* |  |  |
| TR 3 – Lighting  *Please give details of up to 2 lights* | | |
| Light 1 (Dir Light)  (Blinn-Phong)  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of light 1* |
| *I turn off the HDR and p-p to make it clear.* | *All light info is passed from CPU to GPU as UBO stream to save memory. (same as view/proj matrices)*  *Relevant class/file:*  *Light.h (DirLight struct)*  *Shader:*  *player.fs*  *fish.fs*  *tile.fs* | *All lights are added in Scene.*  *All lights’ specular value is implemented by Blinn-Phong model instead of Phong to make it more natural. (See player.fs line 113)*  *You can also do it in RenderSystem::Render() to call HDR::Unbind() to disable the HDR and Postprocessing.* |
| Light 2 (Point light)  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of light 2* |
|  | *Relevant class/file:*  *Light.h (PointLight struct)*  *Fish.h*  *Shader:*  *tile.fs*  *See line 135 CalcPointLight() in fs* | *Each fish object holds a ref to a point light and everything point light follows its own fish and is rendered dynamically.*  *I only make the point light affects Tile object (Because I used HDR and Bloom already, the player and fish are bright enough!)* |
| TR 4 – Texture  *Please give details of at least 2 textures* | | |
| Texture 1  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of texture* |
| See Object 2 3D modelling | *Loading Textures for each Mesh:*  *Model.h line 804, line 833*  *Texture is stored in:*  *Model*  *|\_\_ vector<Mesh>*  *|\_\_material*  *|\_vector<Texture>* | *1. Gamma correction (Model.h line 833)*  *2.Mipmap (line 862)* |
| Texture 2  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of texture* |
| 1. Skybox(cubemap) 2. Multi framebuffer and textures in HDR and post processing (if count) | *Relevant class/file:*  *Skybox.h*  *Skybox::Init()*  *Skybox::LoadCubemap*  *Skybox::Draw()*  *HDR.h* | *The skybox is randomly generated from [website](https://tools.wwwtyro.net/space-3d/index.html" \l "animationSpeed=2.951835877925188&fov=71&nebulae=true&pointStars=true&resolution=1024&seed=2jwd2epyqo80&stars=true&sun=true)(click)* |
| TR 5 - Interactive camera  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of interactive camera* |
| Press F to switch between free-look and following player | *Relevant class/file:*  *Camera.h*  *See setting in line 14*  *Transform.h* | *1. press F to switch camera*  *2. use mouse to free-look*  *3. scroll to zoom*  I implemented a Follow() func to make the camera follow an entity/transform. You can simply add a code Follow(anyfish.transform) to randomly follow a fish! |
| TR 6 - Interactive object  *Please give a screenshot* | *reference specific code (filename and line)* | *Description of interactive object* |
| Please see object transformation 2 (Ripple effect) | *Tile.h*  *Tile::Ripple()* | The tile object is interactive with the player’s position. |
| Conclusion  *Please describe what you perceive to be the strengths and weaknesses of your project* | *Describe what aspect of it you are particularly proud of, and what you think would need to be improved.* | *Reflect on what you have learned during this project that you can apply in future projects to improve your performance.* |
| Strengths:  The scene looks awesome to me😉 (even few objects)  *Weaknesses:*  *Too many…see sections in the right->>>* | *Proud of:*  1. Customized Shaders for each object.  2.*Skybox, OBJ Model Loader, Batch Rendering, HDR, Post-processing*  *3. UBO (opengl version 3 is dumb), it doesn’t allow arbitrary data to be streamed as UBO, so u will find a lots of dummy float variables in my light struct in the shader😉*  (U can only pass vec4s or mat4s as UBO, yes, not even vec3 for memory binding issue I guess) | 1. How to build a renderer in an OOP way.  2. Design patterns useful for renderer  3. How light is simulated by math genius  4. A lot of linear algebra and its geometrical meaning!  Need to Improve:  1.Better OOP architecture in order to enhance this CW to build a renderer with more features.  2.Need to add a GUI system  3.The graph structure of Entity and Transform class used for updating frame still is not much robust.  4.All light structs are actually a bit dumb or more like static lights. Need to improve if I want light to be dynamic (will implement deferred rendering in the future)  5. Post processing and HDR should be separated (As I only gave a try for bloom, so I put them together inside the HDR class)  6. Default MSAA is not good enough.  7. I made a loads of data member public because time limitation and I’m lazy….Bad habit! |