Benjamin VIAL
29590765

COMP 371 – Final Project Report
Instructors: Sudhir P. MUDUR
Charalambos POULLIS
Pages: 6

Purpose

The application presented in this report is an extension of a graphics engine, written using the following:

- C++ 11
- OpenGL 3.3 (core profile)
- GLEW 2.1.0
- GLFW 3.2.1
- GLM 0.9.8.5

Its main goal was to provide learning grounds for modern OpenGL graphics programming. Following is a list of implemented concepts and techniques.

Implemented concepts and techniques

Category	Concept or technique	Comments
Lighting	Phong lighting model	Ambient, diffuse, and specular lighting
Lighting	Rim lighting	Additional lighting around the edges of an entity, dependent on camera position
Lighting	Day and night cycle	Dynamic sun and moon positions, dynamic light source color dependent on time of day
Lighting	Fog scattering and absorption	Squared exponential fog factor, color dependent on time of day
Shadows	Shadow mapping	Two-pass shadow computations for point shadows, using cubemap textures and grid-based PCF
Collision	Collision detection	Bounding spheres and pair-wise detection, $O(n^2)$
Collision	Collision resolution	Arbitrary deciding on a model to stay stationary until collision is resolved
Animation	Keyframe animation	Frame-dependent running animation
Animation	Constrained pathing	Frame-dependent entity pathing bound to the rendered ground texture
Modelling	Hierarchical modelling	Joint-based hierarchical modelling
Other	Skybox	Color dependent on fog density and time of day
Other	Grass	Billboard, star-shaped polygons instantiated and warped in the fragment shader
Other	Particle effects	Falling rain
Other	Procedural generation	Pseudo-random entity placement, orientation, and pathing
Other	Frame rate display	Window title updated periodically with current FPS
Architecture	Optimized VAO binding	Reduced number of VAO switches by having all cube-based, modelling entities share a unique VAO
Architecture	Object-oriented code	Focus on reusability and flexibility (to a lesser extent towards the end!)

User input

Custom implementation controls are listed below.

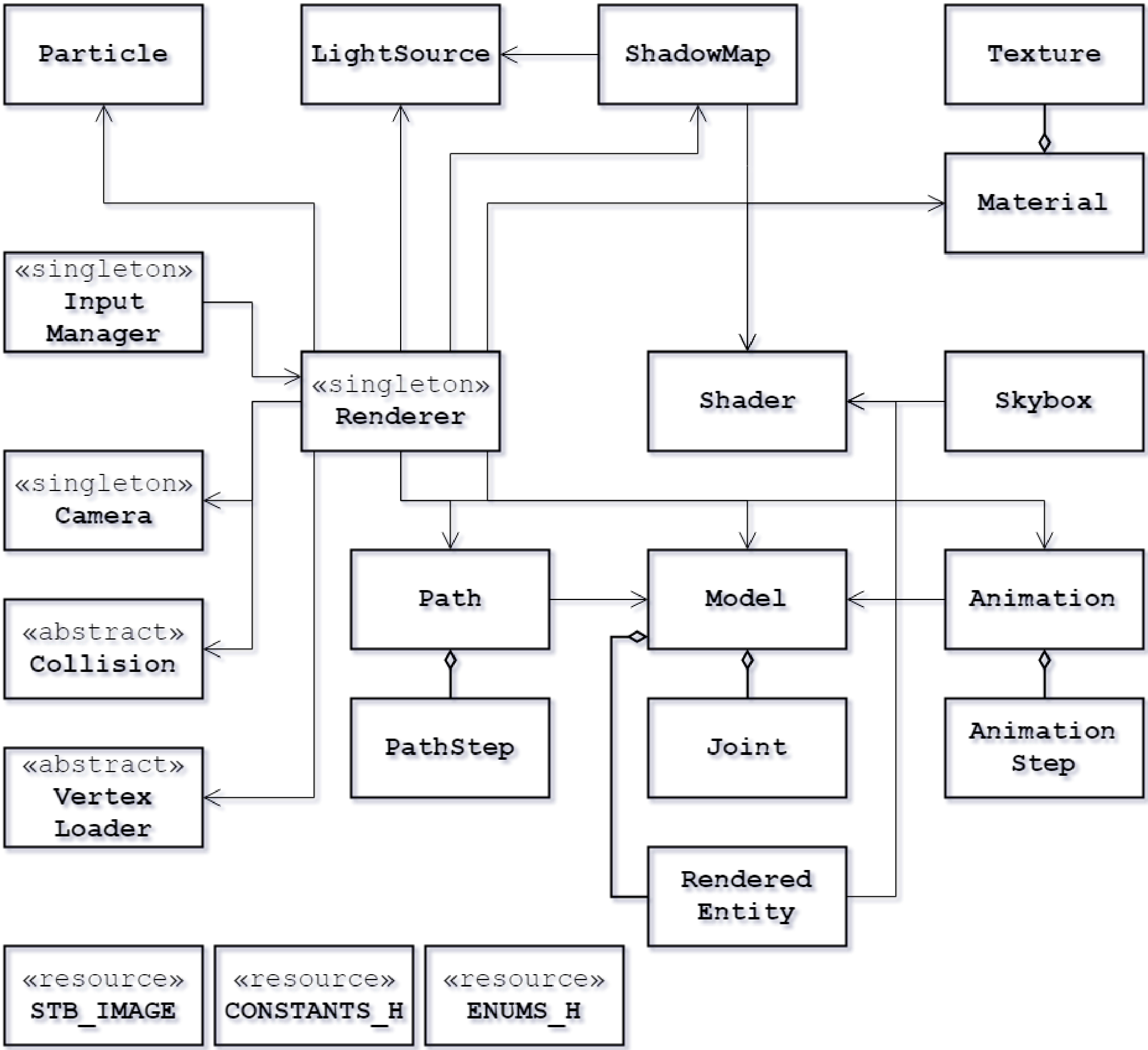
Input	Action	Comments
=	Change shadow map PCF grid samples	Increase (hold SHIFT to decrease)
[	Change shadow map maximum bias	Increase (hold SHIFT to decrease)
]	Change shadow map minimum bias	Increase (hold SHIFT to decrease)
;	Change shadow map PCF grid factor	Increase (hold SHIFT to decrease)
'	Change shadow map PCF grid offset	Increase (hold SHIFT to decrease)
Y	Toggle fog	
I	Toggle rain	
F	Toggle day and night cycle	Shader-generated grass animation is enabled if day and night cycle is enabled
V	Toggle shadow map debug quad	Render cubemap negative-Y texture

Actions that are requisites for the project (as in, not extending assignment guidelines) are listed below for reference.

Input	**Action**	**Comments**
UP	Rotate scene around the X-axis	Upwards rotation
DOWN	Rotate scene around the X-axis	Downwards rotation
LEFT	Rotate scene around the Y-axis	Leftwards rotation
RIGHT	Rotate scene around the Y-axis	Rightwards rotation
ESCAPE	Quit application	
SPACEBAR	Set main model position	Pseudo-random position, clamped to ground
0	Rotate head	Clockwise (hold SHIFT for counter-clockwise)
1	Rotate neck	Clockwise (hold SHIFT for counter-clockwise)
2	Rotate right shoulder	Clockwise (hold SHIFT for counter-clockwise)
3	Rotate right elbow	Clockwise (hold SHIFT for counter-clockwise)
4	Rotate right hip	Clockwise (hold SHIFT for counter-clockwise)
5	Rotate right knee	Clockwise (hold SHIFT for counter-clockwise)
6	Rotate left shoulder	Clockwise (hold SHIFT for counter-clockwise)
7	Rotate left elbow	Clockwise (hold SHIFT for counter-clockwise)
8	Rotate left hip	Clockwise (hold SHIFT for counter-clockwise)
9	Rotate left knee	Clockwise (hold SHIFT for counter-clockwise)
-	Rotate torso	Clockwise (hold SHIFT for counter-clockwise)
H	Toggle pathing	Animations are enabled if pathing is enabled
B	Toggle shadows	
Z	Toggle lighting	Shadows are disabled if lights are disabled
X	Toggle textures	
C	Toggle frame and grid	
R	Toggle animations	
U	Scale main model up	Clamped between 0.1f and 5.0f
J	Scale main model down	Clamped between 0.1f and 5.0f
P	Change rendering primitive	GL_POINTS
L	Change rendering primitive	GL_LINES
T	Change rendering primitive	GL_TRIANGLES
HOME	Reset camera position and rotation	
END	Reset main model position, rotation, and scale	
W	Transform main model	Clockwise rotation around Z-axis (hold SHIFT to move along positive X-axis)
A	Transform main model	Counter-clockwise rotation around Z-axis (hold SHIFT to move along negative X-axis)
S	Transform main model	Clockwise rotation around Y-axis (hold SHIFT to move along positive Z-axis)
D	Transform main model	Counter-clockwise rotation around Y-axis (hold SHIFT to move along negative Z-axis)

Architecture

Because the architecture turned into a hefty beast over the last few months, and because of a wish of keeping this report concise, it has been resumed into the UML class diagram below, which hopefully conveys the strategy employed for concept encapsulation. Implicit relationships have been left out for the sake of legibility.



Challenges and obstacles

Implementing hierarchical modeling was the first road block encountered during the development of this application. Eventually, a map for a given model was established between an entity and its parent, provided they make up the same joint. The model matrix for the entity is then computed by pushing model matrices, excluding scaling, upon a stack, traversing the map until the pointer to a parent is null. The stack is then unwound, adding joint rotations in the process.

The next major obstacle was point shadows, and the difficulty in rendering a proper cubemap texture. Although a traditional, one-texture approach would have been fine for the requirements of assignment 2, it had already been decided at that point that a day and night cycle was to be implemented, which would be tremendously more difficult to do with a single texture. Eventually, the external tool RenderDoc (highly recommended: https://renderdoc.org/) was used to debug shader uniforms and the textures and buffers written to by the graphics pipeline.

Lastly, with the addition of several other models to the scene, frame rate began to drop, and time-dependent implementations began to behave erratically. It was decided to opt for a time step taken from the minimum of the time taken by the rendering loop, and the screen refresh rate (1/60). This solved some, but not all issues, and pathing kept on being severely affected by frame rate, sometimes causing the models to, for lack of a better expression, pirouette wildly in place before the path step was updated.

But, perhaps more than all of these, the main challenge was dealing with a mixture of constant self-criticism, the will to refactor code, and wanting to implement as many features as possible. This project provided the perfect learning grounds in the field of computer graphics, and it was difficult to cut down on features to polish the main requirements.

Further development

As they were implemented last, shader-warped blades of grass and particle effects could have been made prettier. Fog scattering and absorption would also have looked more natural if it had been calculated as a volume instead of a color on a flat texture, and if the skybox texture had been affected by it in a more arbitrary fashion (such that only the horizon was covered in a thick fog). Shadows suffer from severe perspective aliasing, which cannot be solved by simply tweaking bias and PCF sampling. Cascaded shadow maps seem like a reliable alternative.

Finally, because it seems like a constant, nagging voice in the back of my head, extensive refactoring should be done to the engine to allow for a more flexible and reusable structure. Building a graphics or game engine from scratch is a (very!) long-term project, and this course has been an excellent incentive towards pushing me in undertaking that endeavor.

Final thoughts

As a final note, I would like to share my gratitude for taking the risk to change the structure of the course, and how much I appreciate making all submissions individual. I believe the goals of the instructors, which were to encourage personal learning and to discourage unequal contributions, were reached far and beyond. Thank you for an excellent course.

UML class diagram drawn on *draw.io*.
Resource credits included in the `credits/` folder.
I certify that this submission is my original work and conforms to the Faculty's Expectations of Originality.

Benjamin Vial, on 2018-04-12