OpenGL City Scene

Szymon Jackiewicz [150249751]

November 2019

1 Scene Overview and the minimap

Screenshot taken at the end of the animated sequence that displays an overview of the city. The city is randomly generated and loaded when the program is run.

There are 3 different directional lights illuminating the scene from different angles, each casting it's own shadows. Due to this, in the next screenshots, 3 different shadows can be cast from one object.

In the bottom left corner, a small display can be seen, showing an amplified version of the depth map from the perspective of the helicopter spotlight. This feature has been toggled off in other screenshots. (See Section 6.1)

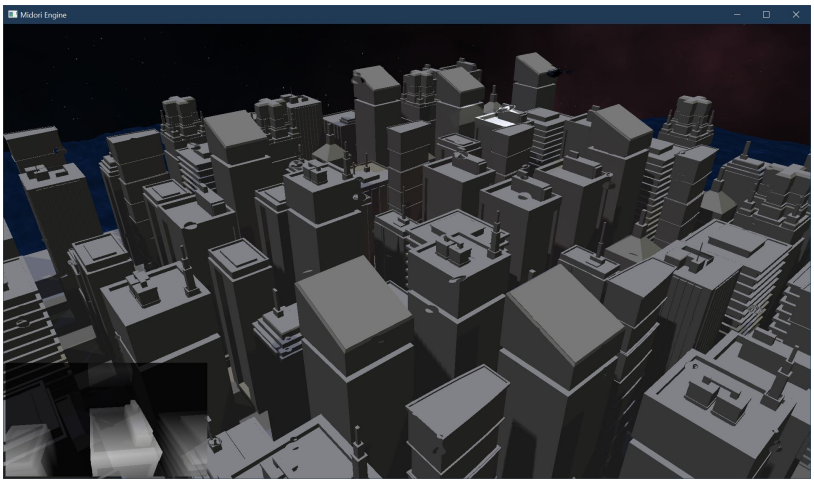

Figure 1: Bird's eye view of the generated city with helicopter perspective minimap.

2 Police cars with alternating point lights

Between the buildings, police cars can be seen driving randomly in the streets. Each police car has a blue point light and a red point light. These lights alternate at a constant rate as the cars drive, illuminating the road and nearby buildings.

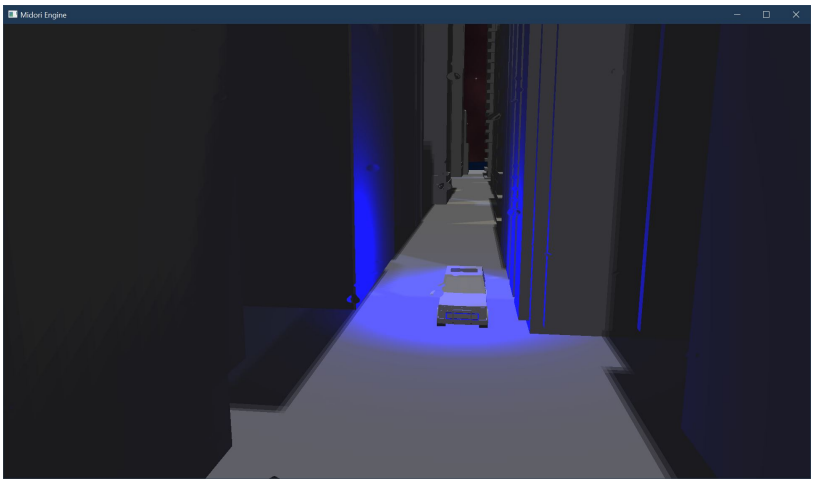

Figure 2: Perspective from between the buildings with a police car driving towards the camera.

3 Spotlight shadow map

The spotlight is cast from the front of the helicopter and pans left to right from it's position. When geometry is hit by the spotlight beam, it is illuminated, however a real-time shadow is cast behind the object.

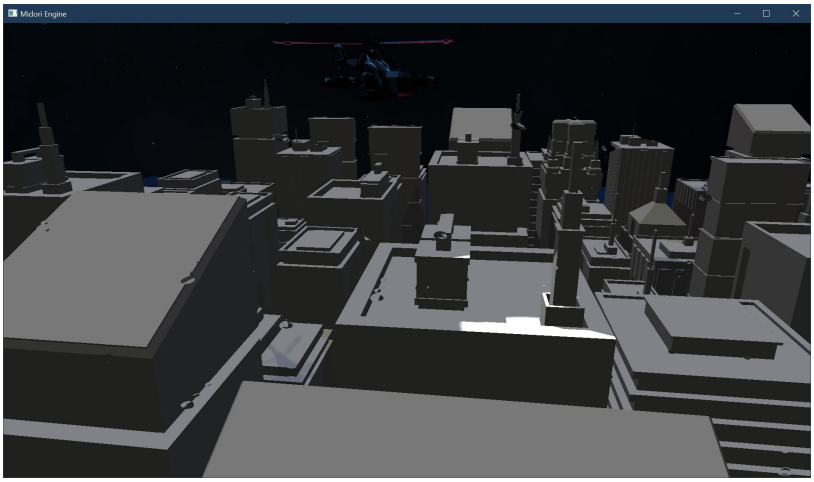

Figure 3: Helicopter casting a spotlight on uneven terrain causing the light to not be rendered on faces not seen from it's perspective.

4 Helicopter terrain mapping

The Helicopter has a reflective appearance by reflecting the sky in a mirror-like way towards the camera. The stars in the sky texture can be seen on the texture of the helicopter.

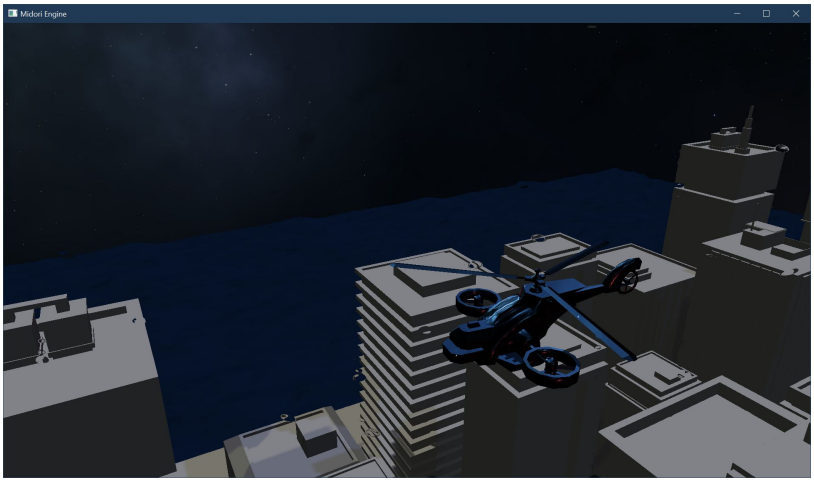

Figure 4: Helicopter surface using the camera's position to reflect appropriate samples from the sky.

5 The water around the city

The water around the city, is arranged in square tiles that are tessellated into many triangles. Each of these triangles is then displaced using a displacement map, and new normal values are calculated from a normal map. The water moves over time in a way where the seams between the tiles cannot be seen.

The water tiles that meet with the ground of the city, are tessellated less on the edge for smoother transition.

While the water has a blue colour, it still reflects the sky onto it's colour. directional lighting calculations are also performed.

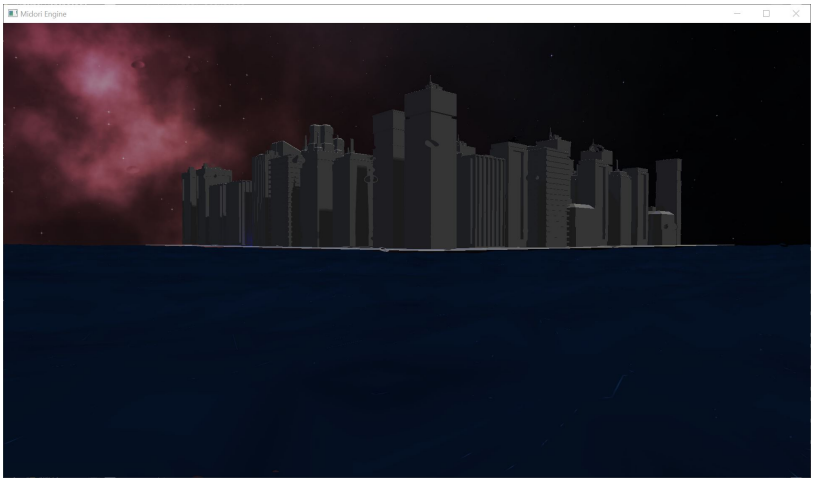

Figure 5: Close-up of the water tiles that surround the city.

6 Keybinds and links

6.1 Keyboard controls

Key	Action
w	Move the camera towards it's facing direction.
a	Move the camera left in relation to the facing direction
s	Move the camera backwards in relation to the facing direction
d	Move the camera right in relation to the facing direction
Shift	Move the camera downwards in relation to the facing direction
Space	Move the camera upwards in relation to the facing direction
Escape	Toggle between mouse cursor being released and controlling the camera facing direction.
m	Toggle between showing the depth information from the the helicopter spotlight on the screen.

6.2 Links

- YouTube Showcase
- GitHub Project Repo
- Game engine used