A5 Project Proposal

Surveying Neo Tokyo

Zhi Kai Lu
20666916
zklu

Final Project:

Purpose :

To create an interactive game where the user controls a helicopter surveying the chaotic dystopian city.

Statement :

This project aims to tell the story of a Neo-Tokyo from sci-fi film "Blade Runner", where the player explores the city from a newscopter equipped with searchlights and audio enhancing technology. Audio and interesting buildings are key components of this project, so a large chunk of time will be dedicated to sampling and/or creating these assets. Further, this game serves to demonstrate core computer graphics concepts covered in the course, alongside some more complex algorithms from academic papers.

The course notes briefly covered 'lighter' lighting techniques and possible optimizations; I'm interested in further exploring lighting algorithms to implement a robust technique that's suitable for quickly rendering with multiple lights for a moving camera, as is often the case in videogames. I also want to test the limits of the graphics lab's computers: at what point are there too many meshes to render on the fly? What optimizations can I do then? How much do these optimizations improve the speed of rendering? Similarly, I love RPGs with story-telling through game mechanics and details, and creating a complex environment with audio would be my homage to such games. To me, the most challenging part of the project will be juggling the big list of objectives, prioritizing progress on certain features so that other features can be sensibly built on top of other features.

Technical Outline :

Buildings will be created using the L-system concept from the course notes. Neo-Tokyo has a lot of skyscrapers so adequate width, height, and various roofs will be defined for the grammar. Texture map will be added to them, without bump mapping for difficulty and efficiency reasons - and also because the scene will take place in night with many buildings obscuring other ones.

Besides buildings, the models (people, cars, traffic lights, billboards, streetlights, etc.) will be hand created and defined as lua objects with a hierarchy for efficient transformation, varied characteristics and multiple instances. Some models (cars, lights) will change state (move, broadcast different sounds, show different colours) over time this AI will be handled with polymorphism in openGL's appLogic. For example, people will enter buildings and have different audio when they are in the building, cars will honk or move to an alley when the player shines light on them for too long.

Beyond simply meshes, each object in my scene goes through its own set of logic per render, producing some animation. Hopefully this doesn't result in any latency as the player navigates the scene. While I'm currently unsure of how many models the graphics computers can handle, I am prepared for reducing lag with a few strategies: clipping and hidden surface removal, idle state for when object is sufficiently far away, and a simplified representation of an object when it is somewhat far away. The specific distances will have to be tested with trial and error to avoid objects jumping out.

There will be many billboards, storefront signs, and car headlights as sources of light in the scene. I'd need a robust algorithm to compute lighting in this scenario. I looked at deferred shading based on the professor's recommendations, but there were many papers that simply namedrop deferred shading. I plan on following Policarpo's paper on the subject matter, and expanding out based on the difficulty and time constraints.

Infrared mode will be achieved by disabling hidden surface removal, rendering non-person objects by setting alpha values according to distances and the amount of objects the eye has already seen through. Logically, this sounds like a form of non-recursive ray tracing, which would be too computationally demanding alongside the proposed shading method and the sheer number of polygons. As such, we can approximate by using the information from the first pass of deferred shading, treating the camera as a light source that contributes alpha value.

The helicopter will be operated by the standard WASD input for movement, while the mouse will control the searchlights and audio enhancer from the helicopter. The helicopter will be responsive while reasonably simulating the swaying and inertia that a helicopter should have. I plan to do this through generating small amounts of random noise for its velocity, and tracking how long a WASD key was pressed before being released.

Because of the popularity of Blade Runner, there exists lots of fanart and audio samples that I can legally use. I also plan to record some of my own samples with a friend's USB mic, and edit the audio through Ableton. Adding these audio files to the program requires an audio library like FMOD or DirectSound. I'll play around with both to see which one uses less memory.

From the sound rendering paper, I will use the basic sound attenuation presented in the preface of chapter 5: 'Effects of distance and direction'. It assumes no object is between the transmitter and receiver. If time permits, I will try to implement a more complex algorithm is calculated during the steps of tracing rays (I'm not using ray tracer to shade but perhaps there's some workarounds to get the information needed for calculations). I will also implement doppler's shifts by keeping track of the player's velocity and altering the frequency of each sound file after attenuation, hopefully there's a library to do that easily.

Bibliography :

Müller, Pascal, et al. "Procedural modeling of buildings." Acm Transactions On Graphics (Tog). Vol. 25. No. 3. ACM, 2006.

Mentions many enhancements that can be added to a grammar for building generation. The sections regarding occlusion optimization and snapping are great for skyscrapers.

Policarpo, Fabio, Francisco Fonseca, and CheckMate Games. "Deferred shading tutorial." Pontifical Catholic University of Rio de Janeiro 31 (2005): 32.

Includes great visual examples and some code about how to transform a basic lighting model to deferred shading.

Takala, Tapio, and James Hahn. "Sound rendering." ACM SIGGRAPH Computer Graphics 26.2 (1992): 211-220.

Thoroughly explores accurate sound algorithms within the context of a computer graphics pipeline.

Objectives:

Full UserID: zklu **Student ID:20666916**

1: Buildings generated from L-Systems look cohesive and different from each other.

2: Scene layout is defined with cars, people, billboards, traffic lights, and parameteres to allow L-building generation.

3: Textures are appropriately mapped onto buildings and billboards.

4: Objects' AI interact with other objects and react to searchlights shining on them.

5: Use deferred shading to illuminate objects' surfaces with numerous (10-20) light sources, and a moving light source (helicopter's searchlight).

6: User can enter an Infrared mode to see people through structures depending on how many structures between camera and person.

7: Reflection implemented in appropriate materials.

8: User interface for controlling Helicopter: steering with keyboard (includes rotation around arbitrary axis), searchlight with mouse.

9: Appropriate sound file is triggered when sufficiently close to respective object.

10: Sound volume calculations: doppler shift and sound attenuation.