MSIM 441/541 & ECE 406/506

Computer Graphics & Visualization

**Programming Assignment Three:**

**Transportation Simulation**

Thomas J Laverghetta

Tlave002@odu.edu

757-620-7607

Virtual  
MSIM411

12/12/20

# Introduction

The transportation simulation project is designed to allow players/users to drive a simulated car through a virtual world that includes roadways with markings and an intersection with a traffic light and surveillance camera on each corner. This document will go over the design, implementation, and results for this simulation so to illustrate use and how to build upon what has been done.

The report will have three sections: program design, implementation and results, and conclusion. The program design will discuss the Doxygen documentation generated for this simulation. Implementation and result is the main section of this document, and will discuss the simulation’s design, implementation, and results from implementation. Lastly, conclusion will discuss accomplishments, learning outcomes, difficulties, and future improvements.

# Program Design

The design documentation for the programming assignment was made using Doxygen. Doxygen is a tool for generating documentation from annotated C++ sources (classes, enums, variables, functions, etc.), and programming languages (C, Objective-C, Java, etc.). To make documentation using your source code with Doxygen you comment your source code with specified markers and description of the code. The markers give specific meaning to the comments (such as identifying the code is a class, enum, variable, function, etc.). The markers and other documentation on Doxygen can be found on their website (<https://www.doxygen.nl/manual/docblocks.html>). Using these markers, I commented my code and ran Doxygen wizard tool. This generated HTML design documentation (PA\_3\_Laverghetta\_Thomas.chm). The HTML design documentation illustrates the functions and variables used. The functions and variables will all have short briefings explaining their use within the program and any corresponding dependencies. Doxygen will also illustrate how the functions are connected and where dependencies are.

# Tasks

## Customized Car

For this task, I replaced the previously used Honda S2000 OBJ model (used in PA2) with taxi OBJ model. The taxi OBJ along with respective MTL and texture files were provided by Dr. Shen; although, the texture images provided were in JPG format, and object parser only accepts PPM format. Therefore, I had to reformat files.

To reformat JPG to PPM I used GIMP. In GIMP, I would open the JPG file, vertically flip the image (needed because JPG files store the pixels by starting from bottom-left corner then proceeding in row major form, while PPM files start from top-left corner then proceed in row major form) and used GIMP’s export capability to export modified JPG to PPM in ASCII format.

I was also tasked with creating a custom license plate for the taxi with the following text: “Thomas MSIM441”. To do this, I modified the previous license plate by replacing its text (ANT 1CS) with required text using GIMP.

## Billboards

# Conclusion and Discussion

## Accomplishments

I was able to create a basic driving simulator using OpenGL’s primitive functionality and Dr. Shen’s ObjModel API.

## Learning Outcomes

I learned how to use different coordinate systems (world coordinates vs car’s coordinates), using OpenGL’s cameras and viewport functions, and transforms.

## Difficulties

I had no major difficulties with the simulation; the simulation was straight-forward to construct. Dr. Shen’s lectures and slides helped with the ease of implementation as it did not take me long to find what I was looking for when stuck.

## Future Improvements

In the future, I would suggest adding more objects to the environment such as trees and/or buildings. Currently, the simulation seems bland. Also, I would consider going through and optimizing the program. Currently, the program is very slow. For example, the timed update function which is set to 20ms takes more than a second on average to run.