CSCI 3260 Principles of Computer Graphics

Assignment 2: Fury Road

Due Time: 5:00pm, Nov. 9 (Mon), 2015

Late penalty: 10% per day.

Fail the course if you copy

I. Introduction

Have you watched the movie ‘Mad Max: Fury Road’? There are lots of cool vehicles in it. In this assignment, you will create one of them by yourself…with openGL…

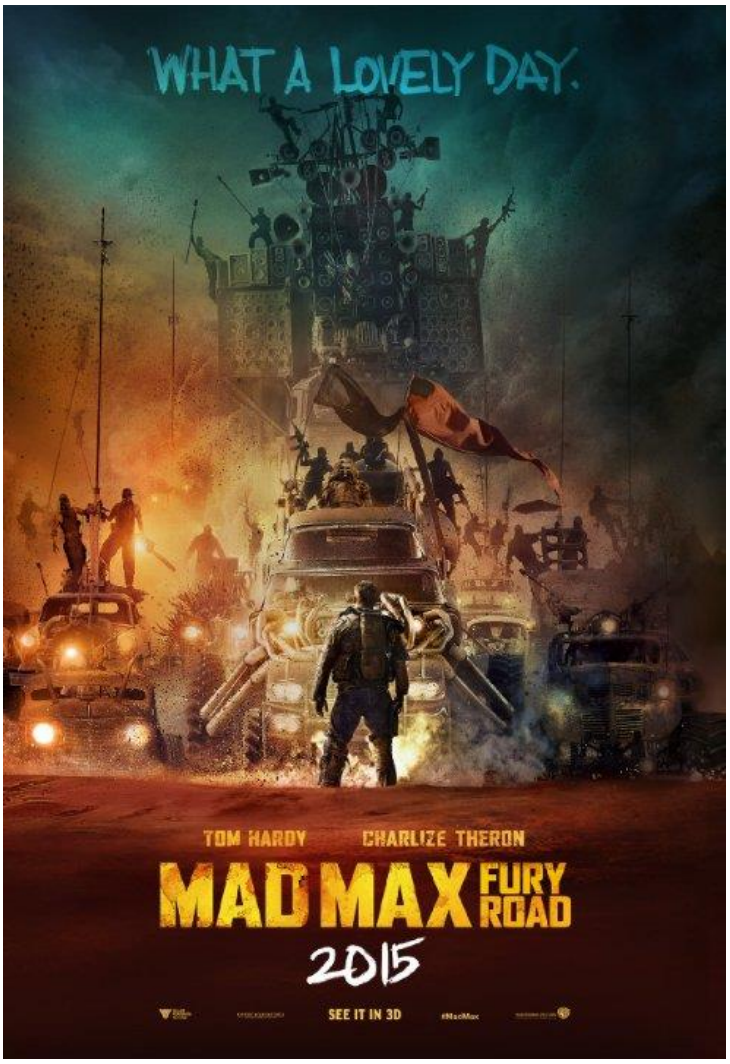


Fig. 1 Mad Max: fury road

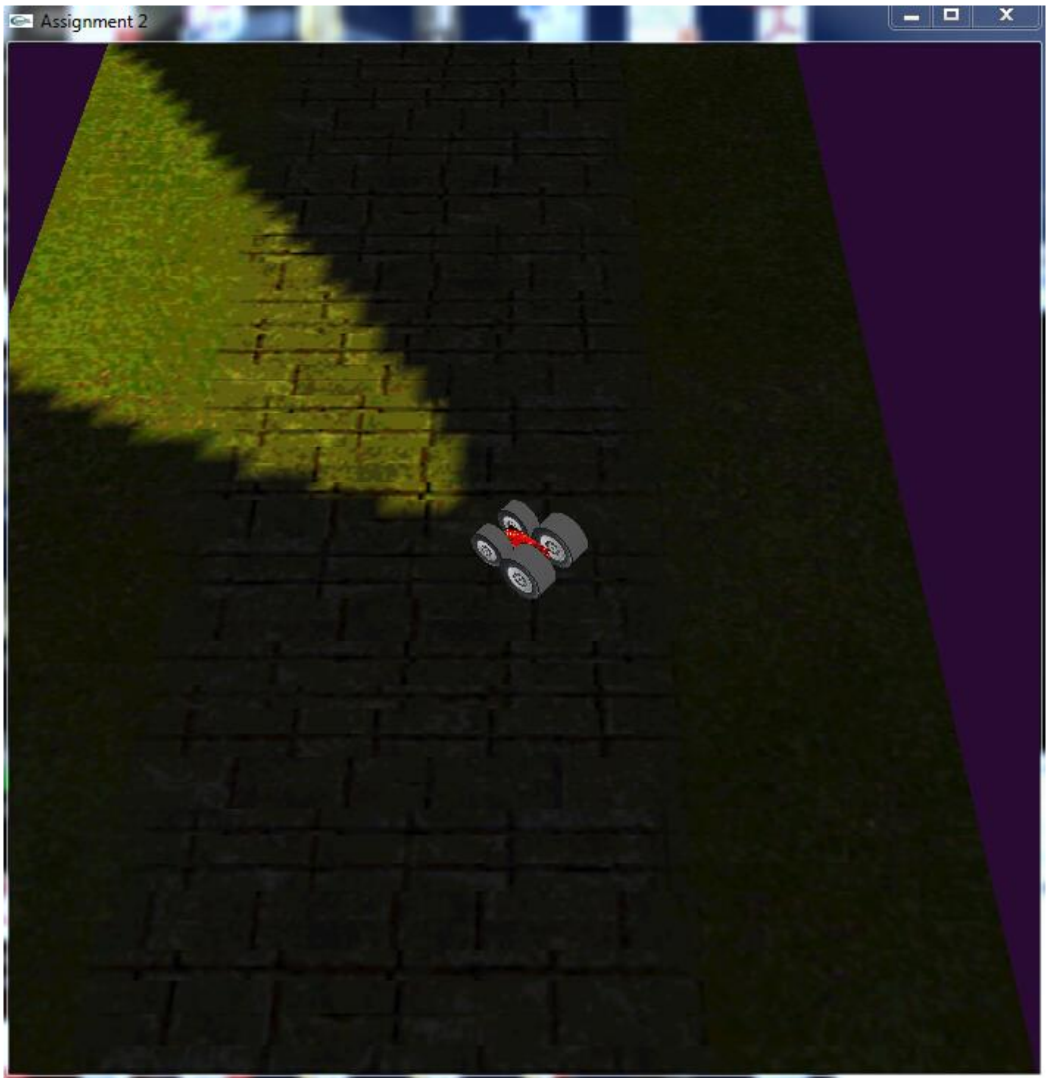
Through designing the vehicle, you will learn how to use view/model transformation to create

3D scene. To make your vehicle more cool and realistic, you will learn to use texture mapping and lighting as well. Last but not the least, we will learn to use keyboard input and window event handling to control your car!

II. Implementation Details

Task 1: Modeling and Transformation (10%)

To begin with, you need to set up a perspective view of the scene for your vehicle. The scene should at least consist of your vehicle and the ground. The ground should be large enough (i.e. 8 times larger than the vehicle) to make your vehicle move around on it.



You have to use openGL transformation commands to build the scene including glTranslate, glRotate, glPushMatrix, glPopMatrix.

Task 2: Texture Mapping and Lighting (30%)

The second task for you is to design your car and make it cool and realistic. The first thing you need is to use texture mapping skills in openGL to map textures onto the corresponding surface. You need to map at least 2 different textures on the vehicle (e.g., the body and wheels) and at least 1 on the ground.

Two extra files (bitmap.c & bitmap.h) are provided to you for reading the bitmap images and converting the image data to raw RGB bytes. You can use function TextureLoadBitmap() in bitmap.c to read the bitmap images.

You should adjust the orientation of the texture map and make sure all the textures are mapped with the corresponding coordinates. Some textures files (in bitmap format) are provided (wheel.bmp, car.bmp and ground.bmp), which are located in the directory of sample program. You are encouraged to use your own textures, or download from internet.

You are also required to introduce at least two light sources into your scene. One of them is for the environment light. You could create a point light source to simulate the environment light. As long as the result is obvious, you can put the light source anywhere you like. The other one is the car light for your vehicle which can move with your vehicle.

You have to use the functions in openGL including glTexImage2D(), glLight(), glMaterial(), glColorMaterial(), etc.

Task 3: Interactive Control of Animation (40%)

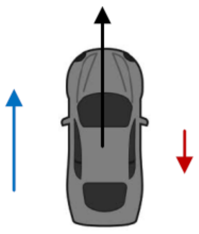
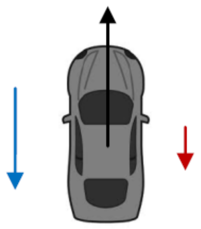
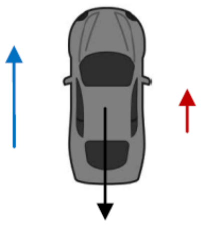
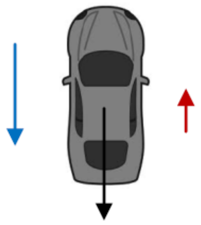
You should be able to control your car on the ground with the following controllable animation:

(a) Car Control

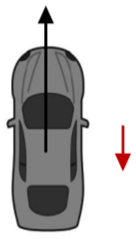
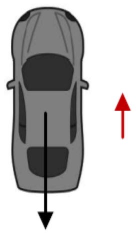
Press "↑↓←→" to start and contrl the motion of the car.

(1) Press "↑" and "↓" to move forward and backward

In this task, you should define a unit vector to represent the direction of the vehicle. Whenever the "↑" or "↓" key is pressed, the vehicle should accelerate or decelerate with constant acceleration along this direction vector. There should always be a frictional acceleration when the car is moving and this acceleration is always in the opposite direction as the moving direction. A list of possible situations is shown in the table below to help you understand the situation:

List of Situations			
	Car is static (velocity = 0). No key is pressed.	→ Moving Direction → Applied Acceleration → Frictional Acceleration	
	Car is moving forward. "↑" key is pressed.		Car is moving forward. "↓" key is pressed.
	Car is moving backward. "↑" key is pressed.		Car is moving backward. " ↓" key is pressed.

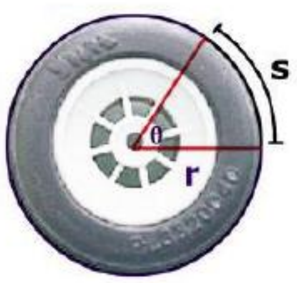
When no key is pressed, the following situation should be handled as well.

	Car is moving forward. No key is pressed.		Car is moving backward. No key is pressed.

You need to use the Newton's second law of motion to make your vehicle's movement more realistic with

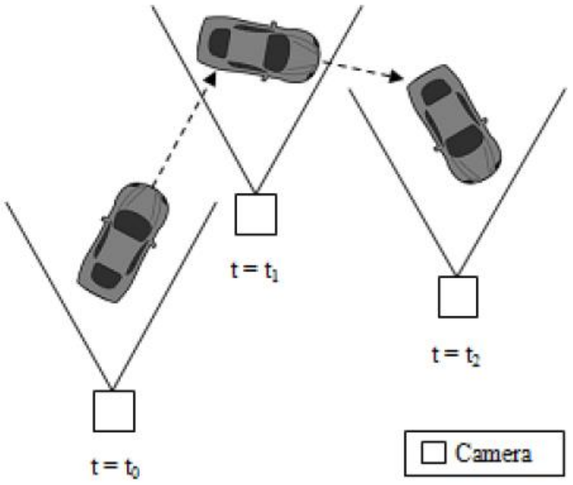
$$v = u + at$$
$$s = ut + \frac{1}{2}at^2$$

The wheels should rotate when car moves with the formula as follows



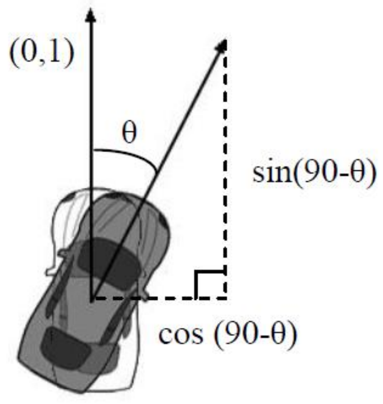
$$\theta = \frac{s}{r}$$

Here, s is the distance travelled by the vehicle, r is the radius of the wheel and θ is the rotational angle. And the camera should follow your vehicle as well.



(2) Press " ←" and " →" to rotate left and right respectively

This should be done by rotating both the car and the direction vector. The relationship between the rotational angle and the direction vector is shown in the following figure:



(b) Reset

Press 'r' to reset the position and orientation of the vehicle. Don't forget to move your camera with the vehicle.

(c) Remarks

Don't let your car get out of the ground. And you may use the following functions the finish the above tasks: keyboard(), keyspecial(), keyspecialup(), idle(), display().

Bonas Task: Enhance visual effect of your animation (20%)

OpenGL provides many functions for your program to create various visual effects. You can study them and introduce them in the viewer. Here are some suggested improvements:

- Complex vehicle and scene with texture
- Additional view of the scene
- Changing lighting

- Smoke when car moves (particle system)
- Interesting scenes
- More movements for the vehicles
- Other effects

III. Grading Scheme

Your assignment will be graded by the following marking scheme:

Basic (80%)	
● Modeling and Transformation of car and ground	10%
● Texture mapping of car (2 textures)	10%
● Texture mapping of ground (1 texture)	5%
● Environment Lighting	10%
● Car Light	3%
● Keyboard Control (a1)	15%
● Keyboard Control (a2)	12%
● Camera Moving	10%
● Reset and remarks	5%
Additional tasks (up to 20%)	
● Complex vehicle and scene with texture	5%
● Additional view of the scene	5%
● Changing lighting	5%
● Smoke when car moves (particle system)	10%
● Interesting extra scenes	5%
● More movements for the vehicles	5%
● Other effects	5%
Total	100%

Note: no grade will be given if the program is incomplete.

IV. Guidelines to submit programming assignments

1) You are suggested to write your programs on Windows, since there will be enough technical support. If you developed the program in other platforms, make sure your program can be compiled and executed on Windows as the program will only be tested on this platform.

2) Modify the provided *submit.cpp*, and provide all your code in this file. No more additional .cpp or .h files are allowed. Type your full name and student ID in *submit.cpp*. ***Missing such***

***essential information will lead to mark deduction*.**

3) Zip the source code file (i.e. submit.cpp), the texture files (i.e. texture.bmp), the executable file (i.e., submit.exe), and the readme file (i.e., readme.txt) in a .zip or .rar file. The total size should not exceed 3MB. Name it with your own username (e.g. wkchan.zip)
4) Submit your assignment in the following way: *Please send your file to csci3260@cse.cuhk.edu.hk* with title "sid Assignment 2" (Note that, don't use Gmail to send the email. Use your cu_link email address)
5) In case of multiple submissions, only the latest one will be considered.
6) ***Fail the course if you copy.***