

Animating a Bicycle - Rendering

CS475M: Computer Graphics - Assignment 2, Part 2

Due Date: 15/10/2016

1 The Assignment

This assignment is aimed at producing an animated (very) short film by the end of the course. There are many steps to making an animation that involve a combination of aesthetic and technical skills. In this course we will learn many of these basic technical aspects. The aesthetic quality of the final result is, however, dependent on how much thought/effort you put into it.

This is the second part of this assignment and it deals with rendering the bicycle that you modelled in the previous part with textures and under various kinds of illumination.

TO DO:

1. Create a room. The room should at least have four walls, a floor and a roof. You are free to create more complex indoor environments, as long as you satisfy the basic requirements for a room. Put the bicycle in this room. Consider the size difference between a bicycle and room, and decide the scale for your room.
2. Texture the floor of the room so that it appears to be a wooden floor.
3. Put a picture frame on one of the walls and texture map a poster or painting on it.
4. Add a texture to the frame of your bicycle.
5. Model a light fixture - this can be a lamp, a bulb, and tubelight - anything you want.
6. Add a headlight to your bicycle.
7. Use the L key to toggle the wall light.
8. Use the H key to toggle the head light. The head light should be a spotlight, that when turned on, should form a round spotlight on the

surface in front of the bike. Remember that this spot will only be visible if you have enough vertices on the surface (as Gouraud shading will compute illumination on the vertices). Another option is to have less number of vertices, but write a per pixel shader for the spotlight. You can choose either one of these options.

9. The effect of lighting must be visible with all textures applied.
10. The left and right arrow keys should turn the handle bar and front wheel of the bicycle left or right.
11. The front arrow key rotates the back wheel and pedals, and makes the bicycle move forward. Compensate for a turned handle, and figure out how to compute the the correct translation for the bicycle given the perimeters of both the wheels, and the angle at which the handle is turned. Each press of the front arrow key should move the bicycle a little and stop - do not make the bicycle move continuously.
12. Define 3 cameras in your scene. Camera 1 moves with the bicycle and is situated at roughly a riders eye level on the bike, it should look to the front and slightly down (so that the handle bars of the bicycle are visible). Camera 2 also moves with the camera but is situated behind and on top of the bicycle, and can see the entire bicycle. Camera 3 is fixed on the wall of the room, near the ceiling and looks down in a fixed direction.
13. The keys 1,2 and 3 should be used to toggle between the three cameras.

DO NOT:

1. Compile and produce an *a.out*. Learn how to use a Makefile, as used in assignment 1.
2. Write code for non-inlined functions in header files.
3. Code that does not compile will not be evaluated.

MARKING:

- Modelling the room properly (minimum 4 walls, 1 floor and 1 ceiling): 50 marks
- Wooden texture on the floor: 10 marks
- Picture frame on the wall with texture of painting or poster: 20 marks
- Texture on cycle frame: 10 marks
- Light fixture model on wall: 10 marks

- Headlight model on bicycle: 10 marks
- Lights toggle keyboard control: 20 marks
- Arrow keys control to move and turn bicycle: 30 marks
- Textures working with lighting in the scene: 10 marks
- Three cameras with their corresponding keys: 30 marks
- Report: 10 marks
- Total: 210 marks
- Deduction - I am expecting everybody to write properly formatted, indented and structured code from now on. Untidy code will be penalized.
- Late submission will follow a policy of graceful degradation with a 25% penalty for each day's delay (i.e., you get zero marks if the assignment is more than three days late after the due date.)

TO SUBMIT:

1. A Tar-Gzipped archive of the complete source code (and only source code). It should compile using the given Makefile on any Ubuntu system.
2. A link to a html report page on the assignment that should contain some details about what you implemented and images of some the results that you generated. Put the link in a README file in the archive you submit.
3. The submission will be through the submission portal.