# CSE 20212: Fundamentals of Computing II
# Group Project Report 3

Yuxuan Chen, Shuyang Li

Yuxuan Chen

**Total time spent: ~19 hours**

- Thursday, March 13, 2014 10:00am - 13:00pm
  - Finished Blender tutorials up to first half of Lesson 5: Materials and Textures on gryllus.net
- Sunday, March 16, 2014 10:00am - 5:00pm
  - Model a motorcycle wheel with Blender
- Thursday, March 20, 2014 7:00pm - 8:00pm
  - Finished Lesson 5: Materials and Textures on gryllus.net
- Saturday, March 22, 2014 2:00pm - 5:00pm
  - Fixed the project so that it does not crash on my computer when it is run
- Sunday, March 23, 2014 3:00pm - 8:00pm
  - Tried to render a simple cube with OpenGL but still failed
  - Started modeling a simple racing track with Blender

Shuyang Li

**Total time spent: ~11.5 hours**

- Mar. 7 6:30p-7:30p
  - try to translate camera with mouse and keyboard events
- Mar. 9 1p-2p
  - successfully moved and translated camera; backface culling
- Mar.12 11a-11:30a
  - found obj file loader and tested (on GitHub, FreeBSD license)
  - didn't write our own object loader because it's a relatively large amount of work (parsing vertex, uv, normal, faces, etc.), and we don't really want to reinvent the wheel
- Mar.13 3p-5p
  - writing OO-wrapper for tiny_obj_loader
  - developed basic game structure and protocol for drawable objects
- Mar.16 10p-11p
  - attempt to render a wheel object that Ethan sent my way
  - not successful: program doesn't draw anything
- Mar.18 6:30p-7:30p

    - debugging program: now throws bad access exception on glDrawElements call
- Mar.19 4p-5p 8p-9p
    - still debugging! asked Jeremy and he doesn't have any clue either
        - buffers are correctly bound, calls follow the correct routine
        - will look into states at drawing time
- Mar.22 2p-5p
    - fixed bad access exception; it's a drawing state issue
    - added glm matrices
    - started using vertex array objects
        - more issue: doesn't draw properly; currently experiencing white screen

Comment: Looks like we're kind of far from finish, but once we fix this drawing issue, we're good to go. All drawing routines are sorted using VAOs, and can be easily separated into corresponding classes; and game logic will take little time to write (speed and turning are easy; collision could be easily hacked). OpenGL is super hard to debug, which brings so much hassle: I've been staring at thirty lines of code for a week trying to fix this drawing issue.