Final Program **Check Point 1 - Due November 6** **Check Point 2 - Due November 20** **Final Due Date - Due December 2**

(Total: 300 pts)

For your final programming assignment, I want to encourage you to have fun as you learn more about graphics. Your program will therefore create an original scene in Minecraft fashion. It is simple but can be used to demonstrate different aspects of graphics like texture mapping, lighting, visible surface detection, surface rendering etc. We will be unable to cover some of the more advanced concepts in this course but learning to implement them using OpenGL through our LWJG Library is very simple. Hence, this project forces you to explore features of the LWJGL without really worrying much about the underlying theoretical principles, especially if you are keen to learn and do a good job in this project. I am encouraging you to be original in your thinking and in your code. Do not look for third party code, no matter how cool or attractive it makes your project. However, if you use someone else's code for a part of your program, give appropriate credit to that source. Remember, a project that just takes third party code from various sources and joins them together to show something will not receive any points vs one which is simple and inelegant but is all yours from ground up.

The following is a list of requirements your program should meet by each checkpoint:

Check Point 1 Requirements:
> As always the Java style sheet should be followed with code separated into separate classes as needed. You should have a window created that is 640x480 and centered on the screen. Your program should be able to display a cube (which is at least 2 in width) in 3D space with each face colored differently. You should be able to manipulate the camera with the mouse to give a first person appearance and be able to navigate the environment using the input.Keyboard class with either the w,a,s,d keys or the arrow keys to move around as well as the space bar to move up and the left shift button to go down. Finally, your program should also use the escape key quit your application.

Check Point 2 Requirements:
> Your program should still be able to do all from the above checkpoint. In addition to the above requirements your program should now be able to draw multiple cubes using our chunks method (creating a world at least 30 cubes x 30 cubes large), with each cube textured and then randomly placed using the simplex noise classes provided (Your terrain should be randomly placed each time you run the program but still appear to smoothly rise and fall as opposed to sudden mountains and valleys appearing). Finally, your program should have a minimum of 6 cube types defined with a different texture for each one as follows: Grass, sand, water, dirt, stone, and bedrock.

Final Check Point Requirements:
> Your program should still be able to do all from the above checkpoints. In addition to the above requirements your program should now be able to correctly place only grass, sand, or water at the topmost level of terrain, dirt, or stone at levels below the top, and bedrock at the very bottom of the generated terrain. A light source should be created that will leave half the world brightly lit and the other half dimly illuminated. Finally, frustum culling should be applied so that non-visible surfaces are not rendered.

If you are motivated enough, you may also add extra functionalities (such as making sure the user does not go past the edge of your created "universe", collision detection, gravity, and "face picking") to the core program requirements we will be looking for. These added functions should be clearly stated in detail in your comments, including how they work (Press F1 to change everything from the current texture to one that makes it look like an alien world for example).

Your completed programs are due at the beginning of the 10th week. I will hand out the criteria for grading the projects on the day of each presentation, every group will get 5 minutes to display their program and be graded by their fellow classmates as well as myself. You may not grade your own presentation/program. Everybody is required to be present on the second day of presentations, otherwise the student missing will result in his/her group being disqualified from the contest and being awarded a zero for that project.

Note: Please do not turn in anything that does not use Java with our LWJGL library. If your project is not working and has bugs etc., you will not receive any credit for it.

What to turn in:
- Soft copy of the program (using Blackboard)