Criteria	Weight	D (0 - 49)	C (50 - 64)	B (65 - 79)	A (80 - 90)	A+ (90 - 100)
Lighting	45	Insufficient or no attempt at rendering a 3D model with Blinn-Phong lighting using the programmable pipeline in OpenGL. OR > As submitted, the release executable does not run or the source fails to build in either debug or release.	> As submitted, both the release executable and source (debug and release) build and run without errors. The application may crash but the goals of the submission are clearly visible before the crash. > At least two 3D models exist in the scene with different distances to the lights, appropriate textures, correct normals, and use a Blinn-Phong shading technique. > At least one point light with attenuation lights up the scene affecting the 3D model(s). > Point lights are represented as untextured (pure color) 3D objects that match the color of the light showing the lights position in the scene. > Point lights affecting the scene can be toggled on/off with a keypress - Key '1'.	As C grade, plus the following additions: > As submitted, both the release executable and source (debug and release) build and run without errors or crashing. > Two or more point lights with attenuation light up the scene affecting the 3D models with varying degree. > A light manager class manages the light properties, passing the light properties into the shader via uniforms, structures, and arrays (where appropriate).	As B grade, plus the following additions: > 10 or more 3D models exist in the scene with textures, correct normals, and use a Blinn-Phong shader. > A single directional light is applied to the scene. > Directional light affecting the scene can be toggled on/off with a keypress - Key '2'.	As A grade, plus the following additions: > A single spot light using the cameras position and direction is applied to the scene and updates on camera movement. > Spot light affecting the scene can be toggled on/off with a keypress - Key '3'.
Cube maps + Reflections	25	Insufficient or no attempt at rendering a Skybox using the programmable pipeline in OpenGL. OR > As submitted, the release executable does not run or the source fails to build in either debug or release.	> As submitted, both the release executable and source (debug and release) build and run without errors. The application may crash but the goals of the submission are clearly visible before the crash. > A Skybox class is created and used to construct an 3D cube mesh and cube map texture. > Rendering the skybox to the scene shows all 6 faces appropriately textured and seamlessly lining up. > The shaders used for the skybox use the perspective division to set the z-component to a value of 1.0f. > The depth function for rendering the skybox is changed to GL_LEQUAL allowing the skybox to be visible.	As C grade, plus the following additions: > As submitted, both the release executable and source (debug and release) build and run without errors or crashing. > At least one 3D object in the scene has a reflection technique correctly applied, reflecting the skybox. > Moving the camera or object updates the reflection appropriately based on the new view angle.	As B grade, plus the following additions: > The reflection technique uses a reflection map texture. The reflection map must allow for a noticeable full/partial reflection on part of the texture while also having zero reflection on a different part of the texture.	As A grade, plus the following additions: > The reflection technique also has Blinn-Phong lighting added and is affected by all lighting within the scene.
Free Camera + Input	20	Insufficient or no attempt at implementing a floating camera that can rotate using the programmable pipeline in OpenGL. OR > As submitted, the release executable does not run or the source fails to build in either debug or release.	> As submitted, both the release executable and source (debug and release) build and run without errors. The application may crash but the goals of the submission are clearly visible before the crash. > A camera using a perspective projection matrix is used to view the scene. > The camera can correctly move forwards and backwards along the cameras forward (look) direction axis using keyboard input (W and S). > The camera can correctly strafe left and right along the cameras right direction axis using keyboard input (A and D). > The camera can correctly move up and down along the global up direction axis using keyboard input (Q and E).	As C grade, plus the following additions: > As submitted, both the release executable and source (debug and release) build and run without errors or crashing. > The camera can correctly rotate on at least one standard axis (pitch and/or yaw) by using the raw mouse motion input to update the look direction. > Correct Multi-Sampling Anti-Aliasing (MSAA) is enabled. > Correct back face culling is enabled. > Toggle wireframe on/off using a key press.	As B grade, plus the following additions: > The camera can correctly rotate on both standard axes (pitch and yaw) simultaneously by using the raw mouse motion input to update the look direction.	As A grade, plus the following additions: > The projection Field of View (FOV) can be changed using the mouse scroll wheel to zoom in and out.
Programming Practices	10	> Insufficient or no attempt at meaningful and consistent programming practices such as commenting, readability, naming conventions, code structure, etc. OR > As submitted, the release executable does not run or the source fails to build in either debug or release.	> As submitted, both the release executable and source (debug and release) build and run without errors. > Submission has the correct file/folder structure although some extra intermediate files may have been included. > Some use of naming conventions and code formatting. > Some commenting is present to clarify the purpose of key areas of code. > A ReadMe.txt file is included stating the functionality, controls, and any additional triggers or needed information for the project to showcase all included features.	As C grade, plus the following additions: > Good use of naming conventions, code formatting, and accessors to increase readability. > Comments are used to clarify the purpose and use of data and functions demonstrating an understanding of the key areas of related code. > Some function headers are present. > Some additional classes and functions have been used where appropriate to refine the codebase.	As B grade, plus the following additions: > No intermediate and/or unnecessary files have been included. > No warnings are generated during building that originates from student project files. > Very good use and consistency of naming conventions, code formatting, and accessors to increase readability across all files. > Function headers are consistent across most files and functions. > Classes and functions are appropriately used to create systems and demonstrate an understanding of modular code.	As A grade, plus the following additions: > Function headers are consistent and present across all files and functions. > Classes and functions are appropriately used to create systems and demonstrate a higher understanding of modular code and proper C++ OOP concepts.