

3D Graphics and Gamification – PROG49635

Exercise 2

Due Date:

See SLATE

Project Type:

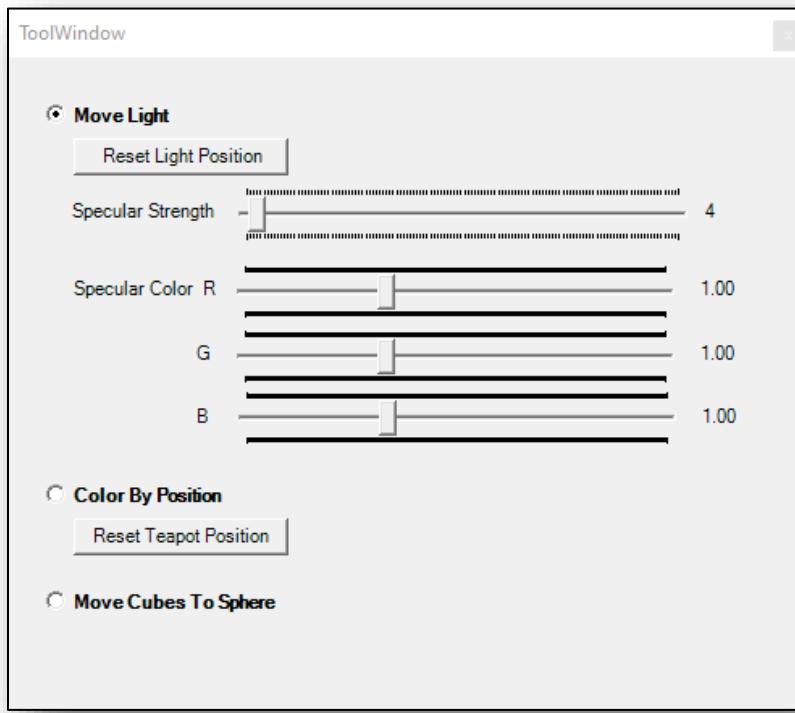
Individual

Submission:

- Your submission will be completed by uploading your entire C++ solution, as a ZIP file, to the Exercise 2 drop box on SLATE, before the due date/time.

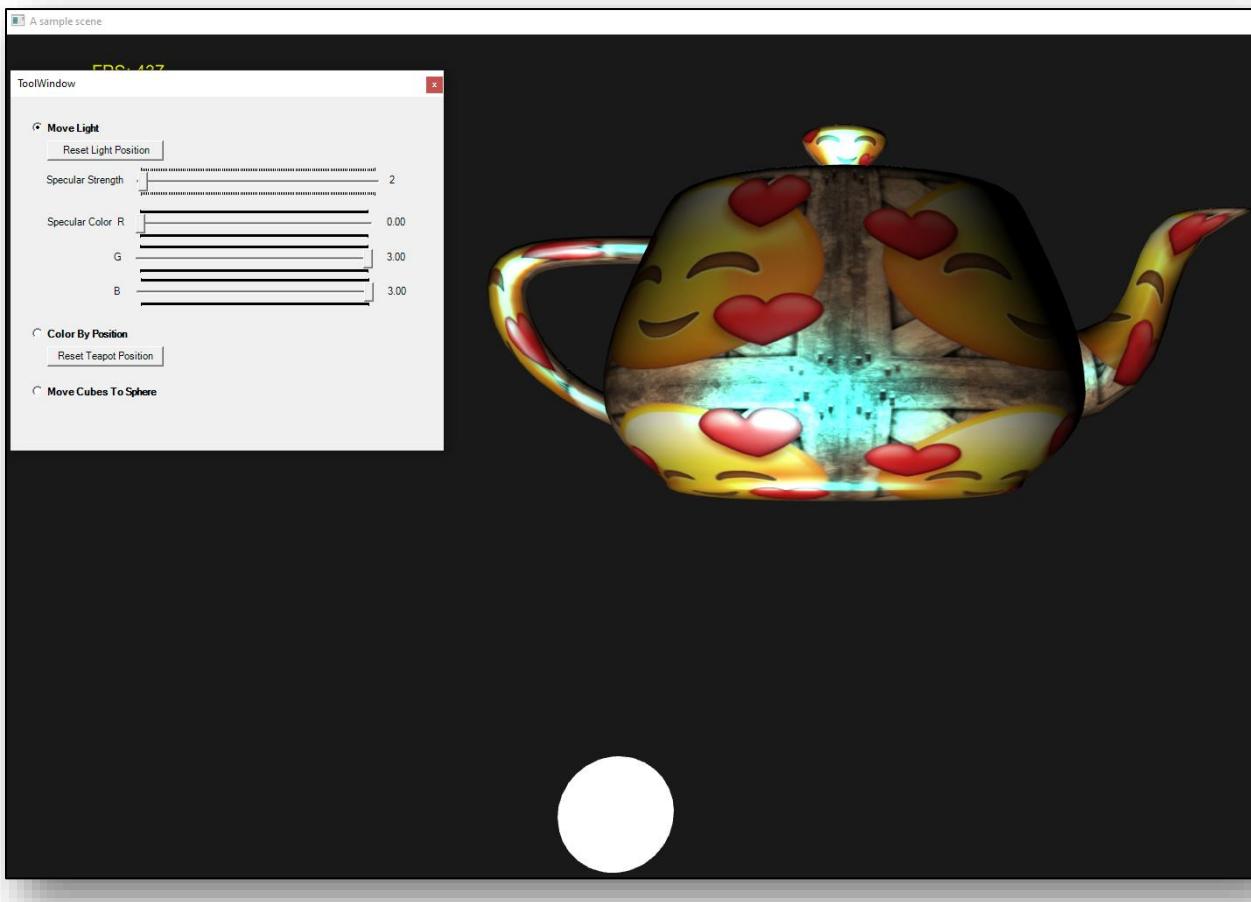
Assignment details and tasks

- Create a new C++ OpenGL application, called MultiRenders. Enable C++/CLI.
- Use 3DS Max (or another 3D app) to create the required 3D models for this assignment. Export your models as OBJS and import it along with required materials into your assignment application.
- All rendered models for this assignment must be constantly rotated around its X axis.
- Add a topmost tool window to the application that contains 3 radio buttons, 2 buttons and 4 trackbars.



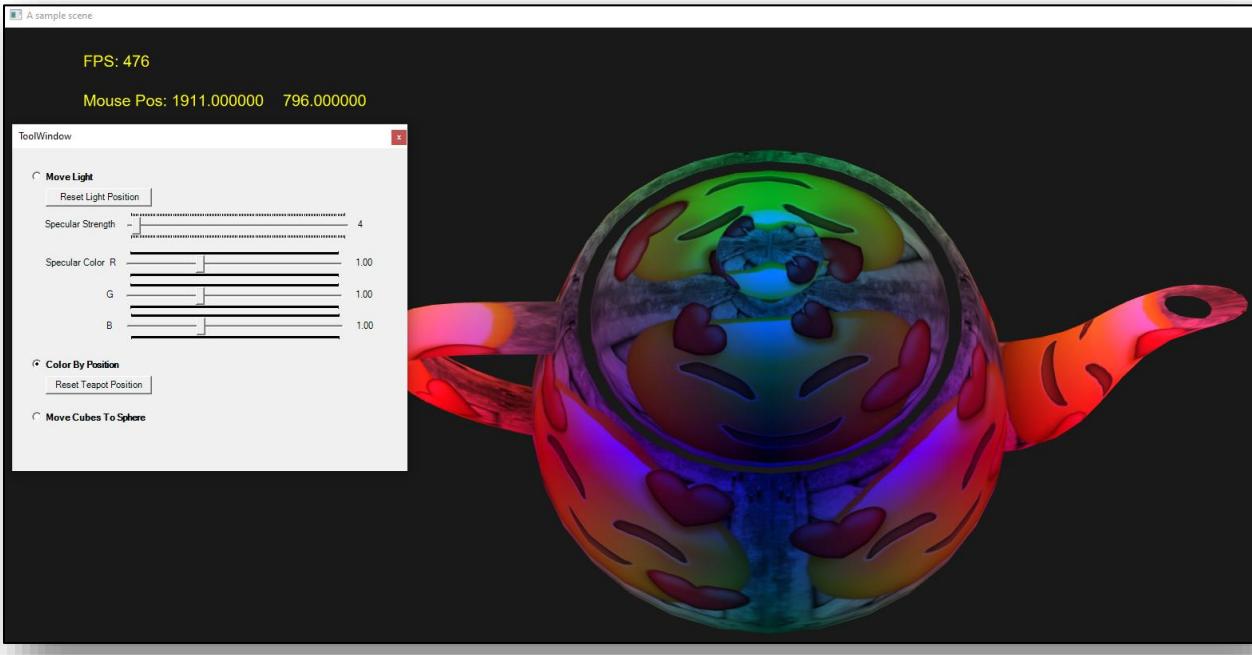
- The “Move Light” option should be selected by default. When selected, the application should:
 - Render a sphere using a shader that only colors the sphere white (no texture). This sphere represents your light for the scene. Initial position: (0.0f, 0.0f, 0.1f)
 - Render a textured teapot (all textures available on SLATE). Initial position (0.0f, 0.0f, 0.0f). The material used to render the teapot must only include:

- *Teapot.jpg (Lambertian)*; and
 - *TeapotSpecular.jpg (specular)*
- The “Reset Light Position” button should reset the light sphere to its initial position.
- The “Specular Strength” trackbar should adjust the teapot material’s specular strength component. The trackbar’s values should be:
 - *Minimum: 1*
 - *Maximum: 128*
 - *Value: 4*
- The “Specular Color” trackbars should adjust the R, G and B specular color components of the light’s specular color component. These trackbars have the following values:
 - *Minimum: 0*
 - *Maximum: 300*
 - *Value: 100*
- The specular color trackbars have labels showing its current value. The labels must convert the trackbar values to values between 0.00 and 3.00. These are the actual values that will be sent to your shader.
- When you click the left mouse button in the render window, determine which quadrant of the screen was clicked (top-left, top-right, bottom-left, bottom-right). Move the light in the direction (quadrant clicked - scene centre) of the clicked quadrant. The further into the quadrant you click, the faster the light should move.

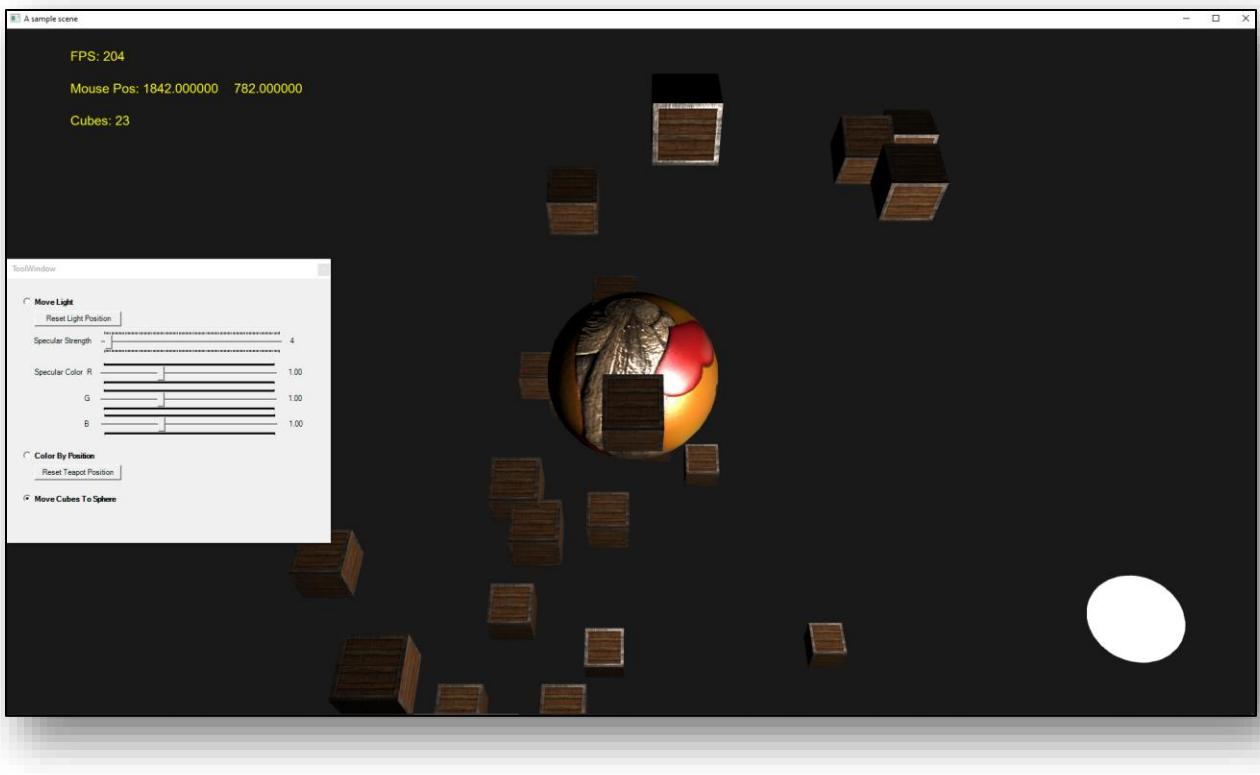


- When selected, the “Color By Position” option should:
 - Render the same teapot as was used in the “Move Light” option
 - The color of each pixel should be determined by its world position, in the pixel shader:

- As the teapot moves towards a positive or negative world X position, the teapot should be tinted progressively more with a red color
- As the teapot moves towards a positive or negative world Y position, the teapot should be tinted progressively more with a green color
- As the teapot moves towards a positive or negative world Z position, the teapot should be tinted progressively more with a blue color
- Move the teapot around with the mouse. Move the teapot using the same method as was used to move the light.
- No advanced lighting (specular, Lambertian, normal) should be implemented for the shader that renders the teapot for the “**Color By Position**” option
- The following screenshot shows how the teapot is textured and colored. It is green towards the top and bottom, blue in the middle and red towards the left and right. As you move the teapot around, the color will change depending on the direction moved.



- When selected, the “**Move Cubes To Sphere**” option should:
 - Render a textured sphere (all textures available on SLATE). Initial position (0.0f, 0.0f, 0.0f). The material used to render the sphere must only include:
 - *Teapot.jpg (Lambertian)* and
 - *TeapotSpecular.jpg (specular)*
 - When the user presses the left mouse button, spawn a cube at a random (X, Y, Z) position around the sphere. The material used to render the cube must only include:
 - *MetalFrameWood.jpg (Lambertian)*
 - *MetalFrameSpecular.jpg (specular)*
 - There is no limit as to the number of cubes a user can spawn
 - The cubes are not instance-rendered, but rendered individually
 - Each tick, each spawned cube should move closer to the sphere, until it touches the centre of the sphere, then that cube should disappear (removed from the list of cubes and destroyed)
 - Print the number of currently spawned cubes in the top-left corner



- See slate for a sample gameplay video
- See next page for Submission Guidelines and Grading Rubric

Grading Rubric

#	Criteria	Marks
1	Completed all the assignment tasks. Clear and concise code will receive higher marks	20
Total		20