A blue and black logo

AI-generated content may be incorrect.

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Course | **COMPUTER GRAPHICS FUNDAMENTALS** | Course Code | **CGD6214** |
| Lecturer | **ASSOC. PROF. DR. JUNAIDI BIN ABDULLAH**  **MS. SYUHADA BINTI ABDUL RAHMAN** | Trimester | **2520** |
| Programme | **BCS** | Due Date | **WEEK 6**  **(20 September 2025, Saturday 11:59 PM)** |
| Assessment | **ASSIGNMENT (INDIVIDUAL)** | Weightage | **30%** |
| Course Outcome to achieve:  CLO2 – Illustrate the basic concepts of 2D and 3D graphics | | | |

**INSTRUCTION:**

**ASSIGNMENT: 3D Solar System with Basic Lighting**

**Assignment Overview**

Create a 3D solar system simulation that demonstrates fundamental computer graphics concepts, including 3D transformations, hierarchical modelling, viewing systems, and basic lighting and shading techniques. Your simulation should show planetary movements with realistic lighting from the sun.

**Learning Objectives**

By completing this assignment, students will:

* Apply 3D transformation matrices for object positioning and animation
* Implement hierarchical modelling for planetary relationships
* Demonstrate understanding of coordinate systems and viewing transformations
* Apply basic lighting models (ambient, diffuse, specular)
* Implement basic shading techniques for a realistic appearance

**PART A: TECHNICAL REQUIREMENTS**

**Core Requirements**

**1. Solar System Components**

* **Sun:** Central star with appropriate scaling and basic emission properties
* **Planets:** Minimum 5 planets (Mercury, Venus, Earth, Mars, Jupiter)
* **Moons:** At least 2 moons (Earth's Moon, one of Jupiter's moons)
* **2D shape -** circle
* **~~Basic Geometry:~~** ~~Use primitive shapes (spheres, cylinders) to create celestial bodies~~

**2. Hierarchical Modelling and Transformations**

* **Transformation Hierarchy:**
  + Sun as root node in world coordinates
  + Planets as children of the Sun with orbital transformations
  + Moons as children of their planets with local orbital transformations
* **Mathematical Implementation:**
  + Use transformation matrices for all positioning
  + Implement proper translation, rotation, and scaling
  + Demonstrate concatenation of transformations
* **Animation System:**
  + Smooth orbital revolution using trigonometric functions
  + Individual planetary rotation on its own axis
  + Time-based animation with consistent frame rates

**3. Viewing and Camera System**

* **Multiple Camera Modes:**
  + Free-look camera with mouse/keyboard control
  + Fixed orbital camera following the solar system
  + Close-up planetary view mode
* **Projection Systems:**
  + Implement perspective projection with adjustable field of view
  + Demonstrate understanding of viewing transformation pipeline
* **Navigation Controls:**
  + Smooth camera movement and rotation
  + Zoom in/out functionality

**4. Basic Lighting and Shading**

* **Phong Lighting Model Implementation:**
  + Ambient lighting component for basic visibility
  + Diffuse lighting from the sun (directional light source)
  + Specular reflection for shiny planetary surfaces
* **Light Source:**
  + Sun as primary directional light source
  + Proper light direction calculations for all planets
* **Material Properties:**
  + Different material coefficients for different planets
  + Demonstrate varying shininess and surface properties

**Advanced Features (OPTIONAL)**

**5. Enhanced Shading Techniques**

* **Multiple Shading Methods:**
  + Compare flat shading vs smooth shading
  + Implement Gouraud or Phong shading
* **Normal Vector Calculations:**
  + Proper vertex normals for smooth spherical surfaces
  + Demonstrate understanding of surface normal computation

**6. Basic Texturing**

* **Planet Surface Textures:**
  + Apply simple textures to at least 3 planets
  + Proper texture coordinate mapping for spherical objects
  + Basic texture filtering (nearest neighbour or linear)
* **Texture Loading:**
  + Load textures from image files
  + Apply textures correctly with lighting calculations

**7. Enhanced Animation**

* **Realistic Orbital Mechanics:**
  + Different orbital speeds for different planets
  + Elliptical orbits instead of perfect circles
* **Time Control System:**
  + Speed up/slow down animation
  + Pause and resume functionality

**8. Visual Enhancements**

* **Basic Visual Effects:**
  + Simple orbit trail rendering
  + Day/night terminator line on planets
  + Basic atmospheric glow effect using colour gradients
* **Improved Geometry:**
  + Higher resolution spheres for smoother appearance
  + Simple ring system for Saturn using flat geometry

**Technical Implementation**

**9. Code Quality and Structure**

* **Modular Design:** Separate classes for celestial bodies, camera, and lighting
* **Mathematical Accuracy:** Proper use of transformation matrices and vector math
* **Shader Implementation:** Basic vertex and fragment shaders for lighting calculations
* **Performance:** Smooth animation without significant frame drops

**PART B: DELIVERABLES**

**1. Complete Source Code Package**

StudentID\_Assignment/

├── src/ # C++ source files

├── include/ # Header files

├── shaders/ # GLSL vertex and fragment shaders

├── assets/ # Textures (if using advanced features)

├── libs/ # OpenGL libraries (GLFW, GLAD, etc.)

├── build/ # Compiled executable

└── README.txt # Compilation and running instructions

**2. Technical Report in PDF (3-4 pages)**

Must include:

* **Mathematical Foundation:** Explanation of transformation matrices used
* **Hierarchical Structure:** Diagram showing your object hierarchy
* **Lighting Implementation:** Description of lighting model and calculations
* **Algorithm Overview:** Key algorithms for animation/navigation
* **Feature Documentation:** Screenshots showing all implemented features
* **Challenges and Solutions:** Technical problems encountered and solutions

**3. Video Demonstration (2-3 minutes)**

Must show:

* **Core Functionality:** All basic requirements are working
* **Camera Movement:** Different viewing modes and smooth navigation
* **Lighting Effects:** Demonstration of lighting from different angles
* **Advanced Features:** Any additional features implemented
* **Smooth Performance:** Consistent frame rate during demonstration

**PART C: ASSESSMENT RUBRIC**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Criteria** | **Excellent (4)** | **Good (3)** | **Satisfactory (2)** | **Needs Improvement (1)** | **Weight** |
| **Geometric Modelling & Components** | All required components with excellent detail and accuracy | Most components are well-implemented with good detail | Basic components present and functional | Some components are missing or poorly implemented | 20% |
| **Hierarchical Modelling & Transformations** | Sophisticated hierarchy with perfect matrix transformations | Good hierarchy with mostly correct transformations | Basic hierarchy working with minor issues | Poor hierarchy or incorrect transformations | 30% |
| **Camera/Viewing System** | Excellent navigation with multiple smooth viewing modes | Good camera system with most features working | Basic camera functionality implemented | Limited or poorly functioning camera | 20% |
| **Lighting & Shading** | Complete Phong model with multiple lights and materials | Good lighting with most components implemented | Basic lighting with some components working | Poor lighting implementation or missing features | 15% |
| **~~Advanced Features~~** | ~~3+ advanced features excellently implemented~~ | ~~2 advanced features well implemented~~ | ~~1-2 basic advanced features~~ | ~~Advanced features are missing or poorly done~~ | ~~20%~~ |
| **Technical Implementation** | Excellent code structure, shaders, and performance | Good technical quality with minor issues | Adequate implementation with basic shaders | Poor code quality or technical problems | 15% |

**PART D: SUBMISSION GUIDELINES**

**Technical Requirements**

* **OpenGL Version:** 3.3 or higher
* **Performance:** Maintain at least 30 FPS during normal operation
* **Dependencies:** Include all required libraries or provide setup instructions

**Submission Details**

* **Format:** Single ZIP file named StudentID\_CGD6214\_Assignment.zip
* **Deadline:** 20 September 2025, Saturday 11:59 PM
* **Platform:** Ebwise

*Please do not submit any reports in hard copy or via email.*

**Late Submission Policy**

* **0-24 hours:** -10% penalty
* **24-48 hours:** -20% penalty
* **48-72 hours:** -30% penalty
* **Beyond 72 hours:** Not accepted

**END OF QUESTION**