Creative Coding Assignment: "Algorithmic Art & Interactive Worlds"

Assignment: Group Project (4 members)
Submission: Source Code, Executable, 1-2 Page Project Documentation, and 3-Minute Demo Video
Submission Deadline: 26th of November 2025
Evaluation (Presentation): 27th of November 2025

Objective: Demonstrate mastery of fundamental graphics algorithms by creating an interactive experience that creatively integrates rasterization techniques with 3D environments.

---

1. The Creative Challenge

You are digital creators tasked with building an interactive application that seamlessly blends **procedural 2D artwork** with **immersive 3D environments**. Your project must creatively incorporate all four required technical concepts, but how you integrate them is entirely up to your imagination.

The Only Rule: Your project must meaningfully use:

- Basic OpenGL lines
- Bresenham's line algorithm
- Midpoint circle algorithm
- 3D model with texture mapping

Everything else is your canvas.

Example 1 – "Interactive 3D City Designer"

Concept:
Students develop a **cityscape generator** that takes user input to design and visualize a custom futuristic city.

User Inputs:

- Number of buildings and layout size
- Road network pattern (grid, radial, random)
- Skyline pattern type (low-rise, mid-rise, skyscraper mix)
- Texture themes (user-chosen building facades and road textures)
- Optional input for park radius or fountain size (for **Midpoint Circle Algorithm**)

How concepts integrate:

- **Basic OpenGL Lines:** Draw roads, boundaries, and guides for building placement.
- **Bresenham's Line Algorithm:** Used for generating clean, pixel-perfect 2D road layouts or connecting road lines on the ground grid.
- **Midpoint Circle Algorithm:** Used to draw circular parks, fountains, or roundabouts.
- **3D Model with Texture Mapping:** Buildings are rendered as 3D blocks with user-selected facade textures.

Immersive Element:
After layout generation, the user can **go through their own city in 3D**, experiencing how their 2D layout translates into a fully rendered environment.

Example 2 – "Interactive Office Room Designer"

Concept:
Students create a **virtual office room** where the user can design the layout, furniture, and decoration interactively.

Inputs:

- Room dimensions (width, length, height)
- Number and size of desks, chairs, and tables (**Midpoint Circle Algorithm** for circular table tops, circular decorations)
- Position of walls, windows, and doors (**Bresenham's Line Algorithm** for straight edges and partitions)
- Texture selection for walls, floors, and furniture (**3D model texture mapping**)
- Optional: Ceiling lights or plant decorations

How concepts integrate:

- **Basic OpenGL Lines:** Draw the room boundaries, window frames, and furniture outlines in 2D.
- **Bresenham's Line Algorithm:** Precisely render walls, door frames, and window edges.
- **Midpoint Circle Algorithm:** Draw circular tables, rugs, or ceiling lights.
- **3D Model with Texture Mapping:** Render furniture (desks, chairs, lamps) and walls/floor in 3D with user-selected textures.

Immersive Element:
Once the user configures the office, they can **walk through the room in 3D** or rotate the camera to inspect their furniture placement, creating a fully interactive experience.

Assessment Part 1 Rubric (50% of the total assignment)

Technical Implementation (40%)

- Correct implementation of all algorithms
- Stable and performant code
- Effective integration of 2D and 3D elements

Creativity & Design (30%)

- Novel and engaging concept
- Elegant integration of required elements
- Polished user experience

Presentation (20%)

- Clear explanation of concepts
- Engaging demonstration
- Professional documentation

Code Quality (10%)

- Readable, well-commented code
- Good project organization
- Appropriate use of OpenGL features

Part 2: "Interactive Storyworlds & Living Environments"

Submission: Complete project files, build executable, design document, and 5-minute showcase video
Objective: Create an immersive, interactive virtual world that tells a story through environmental storytelling, dynamic systems, and user agency.

The Creative Challenge: Build a World with a Soul
Create an interactive VR environment that feels **alive, meaningful, and responsive**. Your world should have:

- **A story** told through the environment itself
- **Dynamic systems** that respond to player presence
- **Emotional atmosphere** through lighting, sound, and animation
- **Purposeful interactions** that reveal narrative and character

1. Scene Composition and Model Representation
Begin by using 3D modelling software (e.g., Blender, Maya, or 3ds Max) to create a basic 3D scene that establishes an atmosphere and narrative. You may include royalty-free 3D models along with your own models.

- **Required Geometry:** The scene must include core static geometry, such as a **House** and terrain, represented by **Polygonal Models**.
- **Animated Objects:** Include a primary **animated object** (e.g., a large swinging beacon, a complex machine, etc.) that demonstrates **Low-Level Animation Techniques** or **Rigid-Body Animation**.
- **Dynamic Objects:** Add characters and at least one other dynamic object that is a non-rigid element (e.g., a flag, a liquid spill, or smoke). This object should be modelled using principles of a **Physically-Based Deformable Model** or a **Particle System**.
- **Appearance:** Apply appropriate materials and **Parametric Texture Mapping** to all key assets.
- **Documentation Focus:** The design document must discuss the chosen **Model Forms** and detail the **Data Structures for Polygonal Models** used in your final scene.

2. Real-Time VR Implementation

Import the 3D scene into a rendering/game engine (e.g., Unity, Unreal Engine) to develop a real-time VR environment.

Rendering and Viewing Algorithms :

- **Navigation:** Implement user interactions to explore and navigate the virtual environment, applying **Affine Transformations** (rotation, translation) for movement and viewing.
- **Scene Environment:** Implement a Skybox, clouds, and ambient sound.
- **Optimization:** To improve performance, you must employ at least one **Culling and Hidden Surface Elimination Algorithm**, such as **Frustum Culling** or **Occlusion Culling**.

Illumination and Shading Control :

- **Lighting Systems:** Implement multiple, distinct light sources. The environment must support and demonstrate **Shadows** (using either **Shadow Volumes** or **Shadow Maps**), reflections, and refractions.

- **User-Controllable Illumination:** The user must be provided with interactive **sliders** or in-world controls to dynamically adjust the light's appearance, demonstrating knowledge of:

  - **Illumination Models:** Controls must adjust the contribution of the **Lambert Illumination Model** (Diffuse component) and the **Phong Illumination Model** (Specular component, i.e., the **specular level**).

  - **Ambient Light:** A control must adjust the overall scene **ambient level** (e.g., controlling the contribution of **Ambient Occlusion**).

  - **Atmosphere & Color:** A control must adjust the overall color/mood of the scene, referencing concepts of **Tone Mapping** and **Color Models**.

Dynamic Physics and System Control:

- **Physics System Control:** The user must be able to change the speed/behavior of a dynamic object (e.g., the animated object, or an element of the **Particle System**) via an interactive in-world slider or menu. **This input must directly modify a physics parameter.**

- **Physics-Based Interactions:** Implement at least one instance of a **Physics-based interaction** (e.g., simple destruction, object manipulation, or object flow).

- **Visualization Input:** You must incorporate a system that visualizes data (e.g., the change in the dynamic object's speed, or a weather state) using principles of **Scalar Data Visualization** or **Vector Data Visualization**.

Example Scene: "Echoes of the Lighthouse"

Concept:
You awaken on a stormy island with an abandoned **lighthouse** that reacts to your presence. As you explore, the world tells the story of the lost keeper through shifting weather, light, and sound. Every change you make alters the environment — the world feels alive and emotionally responsive.

Scene Composition & Models

- **Static Geometry:** Polygonal terrain, rocks, and the lighthouse (core structure).

- **Animated Object:** Rotating beacon lamp (rigid-body or low-level animation).

- **Dynamic Objects:** A torn flag (cloth simulation) and smoke or mist (particle system).

- **Appearance:** Parametric texture mapping on metal, stone, and glass; materials shift subtly with lighting and atmosphere.

- **Documentation Focus:** Discuss polygonal model data structures and texture parameter logic.

Real-Time VR Environment

- **Navigation:** Player moves and rotates using affine transformations (VR controller or keyboard).

- **Environment:** Skybox with day–night lighting, ambient sounds of wind and sea.

- **Optimization:** Frustum and occlusion culling for performance.

Lighting & Shading Controls (User Inputs)
Players interact with **in-world sliders or VR panels** to:

- Adjust **diffuse (Lambert)** and **specular (Phong)** lighting strength.
- Control **ambient light** and overall scene brightness.
- Change **sky and fog color**, altering emotional tone (e.g., calm blue → tense red).

Dynamic Physics & System Control
User inputs directly modify scene behavior:

- **Beacon speed slider:** Controls light rotation (physics motor).
- **Wind strength:** Affects flag and particle motion.
- **Storm trigger:** Amplifies waves, wind, and lighting effects.
- **Destructible object:** Simple physics-based collision (e.g., dropped lantern).
- **Data visualization:** Floating panel shows live parameters like light intensity and wind speed.

Immersive Flow

1. The player arrives to calm light and gentle waves.
2. Adjusting light sliders changes the room's emotion and reveals clues about the story.
3. Increasing wind and storm intensity alters physics, sounds, and visibility.
4. The environment becomes a living narrative — the lighthouse "responds" as if alive.

Presentation Showcase Requirements

Live Demo

- Establish the world and its rules
- Demonstrate key interactions and systems
- Showcase a memorable "wow moment"

Technical Explanation

- Explain one innovative technical implementation
- Discuss VR-specific design challenges
- Show how animation and lighting serve the narrative

Design Rationale

- Explain your environmental storytelling approach
- Discuss what makes your world feel "alive"
- Share your favorite creative decision

Assessment Criteria (50% of the total assignment)

Technical Excellence (35%)

- Quality of 3D modeling and texturing
- Smoothness and believability of animations
- Effective use of lighting and atmospheric effects
- VR performance and comfort

Creative Vision (30%)

- Originality and coherence of concept
- Effectiveness of environmental storytelling
- Emotional impact and atmosphere
- Attention to detail and world consistency

Interactivity & Immersion (25%)

- Intuitive interactions
- Responsive environmental feedback
- Sense of presence and agency
- Polished user experience

Documentation & Presentation (10%)

- Clear explanation of technical and creative choices
- Professional demonstration skills
- Quality of supporting materials