**Final Assignment  
Cube Dodge 3D**

Rishi Gandhi

Student ID: 202014908

Department of Computer Science, Faculty of Science

Memorial University of Newfoundland

COMP 4302: 3D Computer Graphics

Oscar Meruvia-Pastor

March 27, 2024

**Option Chosen:**

I decided to implement a 3D version of a basic arcade game with retro-style elements, incorporating personalized features specific to the members of the group.

**Description of Implementation Features and Characteristics:**

1. Three.js Integration: Utilizes the Three.js library for 3D object creation, rendering, and scene management.
2. Player Cube: A controllable cube serves as the player character, with movement along the x and z axes and a distinctive texture.
3. Environmental Interaction: Player cube interacts with ground and enemy cubes, adjusting velocity and applying gravity upon collision.
4. Orbit Controls: Allows easy manipulation of the camera through mouse input for orbiting and zooming.
5. Particle Effects: Particles orbit around the player cube, enhancing visual appeal.
6. Dynamic Enemy Spawning: Enemies spawn dynamically and move towards the player cube.
7. Lighting: Scene includes directional and ambient lighting, with adjustable properties such as visibility, position, color, and intensity.
8. User Interaction: Keyboard events control player movement and trigger actions like toggling light visibility, changing light position, and camera movement.
9. GUI Controls: Utilizes dat.gui library for a graphical interface to adjust light and camera settings.
10. Modular Design: Organized into classes and functions for improved readability and maintainability.
11. Animation Loop: animate() function oversees continuous scene updates and user input.

**User Manual:**

Interacting with the Program:

1. Movement Controls: Use `W`, `A`, `S`, `D` keys to move the player cube.
2. Toggle Light Visibility: Press `I` key to toggle directional light visibility.
3. Adjust Light Properties:
   1. Toggle light visibility using GUI toggle.
   2. Change light color randomly by pressing `C` key.
   3. Adjust light color using GUI sliders.
   4. Move light position with `Z`, `X`, `V`, `B`, `N`, `M` keys or GUI sliders.
4. Camera Navigation:
   1. Use OrbitControls for mouse-based movement.
   2. Keyboard inputs (`F`, `G`, `O`, `J`, `K`, `L`) move the camera in various directions:
   3. `F`: Move left
   4. `G`: Move right
   5. `O`: Move up
   6. `J`: Move down
   7. `K`: Move forward
   8. `L`: Move backward.
5. Texture Usage: Textures are applied to player cube, enemies, particles, and ground.
6. Particle System: Particles move in circular motion around the player cube.
7. Collision Detection: Game stops if player cube collides with an enemy.

**Video Submission:**

The YouTube video demonstrating the implementation can be accessed here.