ICCS404: Project 2 – AR Cannon Game

Due: December 20, 2024, 23:59

Required Features (80%)

1. Cannon Ball Implementation (10%)

- Create a sphere entity for the cannon ball
- Visibility requirements:
    - Hidden during countdown
    - Visible only during firing animation
    - Hidden after animation completes
- Position at active cannon's barrel tip when firing starts

2. Firing Mechanism (30%)

- Strict sequence when countdown reaches zero:
    1. Make ball visible at firing cannon's barrel tip
    2. Fire the cannon ball with straight or projectile motion
    3. Wait for complete ball animation and collision check
    4. Show hit feedback if collision detected (1 second)
    5. Hide ball and clean up
    6. Only then start the next turn's countdown
- **Important**: Next countdown cannot start until firing sequence completes

3. Hit Feedback System (20%)

- Add red plane under each textured ground:
    - 10% larger than existing ground plane
    - Position slightly below ground plane
    - Initially hidden
- When hit:
    - Show red plane for exactly 1 second

4. Projectile Motion (20%)

- Implement parabolic motion (no straight lines)
- Ball must follow realistic (physically-based) arc trajectory

Extra Features (Pick any combination totaling 20%)

Option 1: Health System (10%)

- Each cannon starts with 3 health points
- Deduct 1 health point per hit
- Display current health above cannon:
    - Choose one style: "♥♥♥", "■■■", "●●●", "3", or "Health: 3"
- When one cannon reaches 0 health:
    - Show "You Won!" above winning cannon
    - Show "You Lost!" above defeated cannon
    - Messages replace health display

Option 2: Random Cannon Angles (20%)

- Use separate cannon_barrel.glb and cannon_mount.glb
- At each turn start:
    - Random vertical angle (20° to 70° from horizontal)
    - Only barrel rotates, mount stays fixed
    - Display actual angle as whole number (e.g., "45°")
    - Angle text above cannon, facing camera

Option 3: Sound Effects (10%)

- Cannon firing sound
- Countdown warning sound
- Background music
- Proper sound management (load/play/stop)

Option 4: Creative Enhancement (10%)

- Must consult with instructor (pisut@cmkl.ac.th) for approval before implementation
- Only one feature allowed
- Your own gameplay enhancement
- Must maintain performance
- Document implementation clearly in README.md

Valid 20% Combinations Examples

- Random Cannon Angles (20%)
- Health System + Sound Effects (20%)
- Health System + Approved Creative Enhancement (20%)
- Sound Effects + Approved Creative Enhancement (20%)

Submission Requirements

- You may hand-in your project individually or as a pair
- Submit as firstname.zip or firstname1_firstname2.zip containing:

```
├── index.html
├── assets/        # All assets in one folder
│   ├── cannon.glb
│   ├── textures
│   ├── sounds
│   └── [other assets]
├── components/     # Each file contains ONE component
│   ├── countdown-manager.js
│   ├── cannon-ball.js
│   ├── hit-feedback.js
│   └── [other components]
└── README.md
```

- Put your name(s) in the README.md and all your code files