Hierarchical 3D Character **Animation**

WebGL-based Frog Jump Game with Physics Simulation

Team name: Three Little Froggies

Member:

이준서

Ensio

Project Structure & Modules

System Architecture and Component Overview

📂 File Structure

⚙️ app.js
Main application logic & initialization

🖌️ renderer.js
WebGL rendering & shader management

👤 character.js
Hierarchical model & terrain rendering

🎮 controller.js
Physics simulation & input processing

</> index.html
Canvas setup & shader source code

⭐ Key Features

🔲 Hierarchical Modeling

🏃 Physics-based Animation

🎨 Lighting & Texturing

➤ Interactive Controls

Collaboration

이준서

Animation

Rendering

Texture Mapping

Game Component

Ensio

Character modeling

Frog Model Hierarchy

Torso — Translate(x, y, z) rotateY rotateX

Left Upper Leg — rotateX
Right Upper Leg — rotateX
Left Upper Arm
Right Upper Arm
Head

Left Lower Leg — rotateX
Right Lower Leg — rotateX
Left Lower Arm
Right Lower Arm
Left Eye
Right Eye

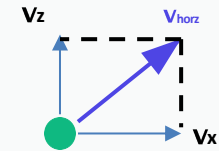Left Foot — rotateX
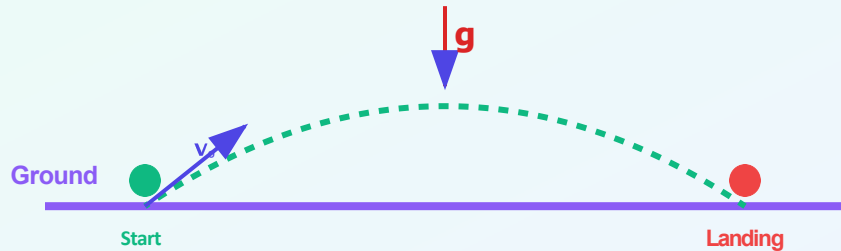Right Foot — rotateX

=11 DoF

Physics-Based Animation
Realistic Jump Motion with Gravity & Velocity

⚛ Physics Principles

📈 Parabolic Jump Trajectory

🧮 Motion Equations

$$V_{horz} = V_0 \times \cos(\theta x)$$
Horizontal Speed

$$y = v_0 \sin(\theta x) \times t - \tfrac{1}{2}gt^2$$
Vertical Position

$$z = V_{horz} \times \cos(\theta y) \times t$$
z Position

$$x = V_{horz} \times \sin(\theta y) \times t$$
x Position

Shader & Texture
GPU Programming & Visual Effects

Shader Pipeline

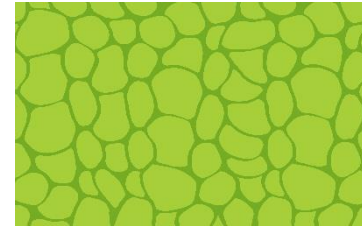| 3D Model	→	Vertex Shader	→	Fragment Shader	→	Final Render

Vertex Shader

• Position transformation
• Normal vector calculation
• Texture coordinate passing

Fragment Shader

• Lighting calculations
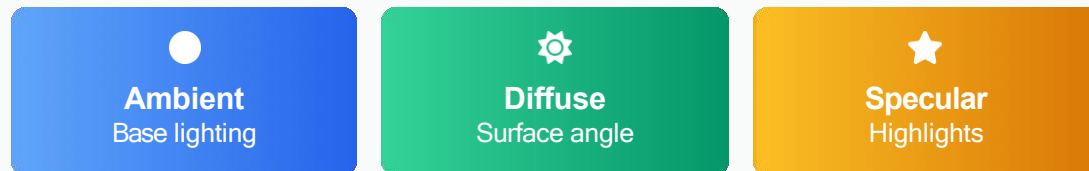• Texture sampling
• Final pixel color

Texture Mapping

Frog Texture

+

Ground Texture

Phong Lighting Model

Ambient
Base lighting

Diffuse
Surface angle

Specular
Highlights

Interaction & Control

User Input **&** Camera Management

🎮 Control Schemes

⌨️ Keyboard Controls

W	**A**	**S**
Forward Angle	Left Turn	Backward Angle

D	**SPACE**
Right Turn	Jump

🖱️ Mouse Controls

Drag to Rotate	Scroll to Zoom
Camera angle control	Distance adjustment

⚙️ Game State Management

✅ Landing Detection	⚠️ Game Over
Reset physics state when character lands	Handle fall scenarios and restart

⭐ Key Features

⚡ Real-time response	✋ Intuitive controls
👁️ Dynamic camera	🔄 State synchronization