

Interactive Storyworlds & Living Environments - Project Documentation

Project: "The Festival of Eternal Light: A Japanese Village Journey"

1. Executive Summary

"The Festival of Eternal Light" is an immersive VR environment where players restore a sacred blade through environmental puzzle-solving to revive a Japanese village festival. The experience demonstrates mastery of computer graphics through dynamic lighting systems, physics-based animations, particle effects, and responsive environmental storytelling.

2. Scene Composition and Model Representation

2.1 Polygonal Models (Static Geometry)

Core Structures: Traditional Japanese buildings (House1, Box001: 2,500-4,000 vertices), shrine complex, torii gates, wooden bridge. **Terrain:** Heightmap-based mesh with stone pathways, cherry blossom trees (3,200 vertices). **Props:** Stone lanterns (Ishidōrō), sacred pedestals, environmental decorations.

2.2 Animated Objects (Rigid-Body & Low-Level Animation)

Primary - Rotating Statue: Continuous Y-axis rotation (30°/sec, user-controllable 0-120°). Implementation: Transform.Rotate(Vector3.up, speed * Time.deltaTime) with state-based activation. Interaction: E key within 5m radius. Audio: Looping stone grinding sound (spatial).

Secondary - Flags (2 nobori banners): Procedural wind simulation combining rotation (pitch/yaw/roll) and position offset. Parameters: wind strength (0.5-3.0), wave height (0.3-0.5), gust intervals (2-5 sec).

2.3 Dynamic Objects (Particle Systems & Deformable Models)

Book Rise Animation: Physics-based lerp interpolation (Vector3.Lerp(current, target, Time.deltaTime * speed)). Displacement: 0.5 units vertical, speed: 2 units/sec. Demonstrates deformable object principles through smooth position transformation.

2.4 Parametric Texture Mapping

URP PBR Materials: Base Map (albedo), Metallic/Smoothness, Normal Map (tangent-space), Occlusion, Emission (dynamic control). **Key Assets:** Lanterns (T_PaperPlates_D.png, cylindrical UV), Windows (T_Windows_D.png + emission map, planar projection), Wood (T_Woodplanks03_D.jpg), Torii Gate (ToriGate_Texture.jpg).

Dynamic Emission:

```
csharp
```

```
material.EnableKeyword("_EMISSION");
```

```
material.SetColor("_EmissionColor", glowColor * intensity);
```

```
material.globalIlluminationFlags = MaterialGlobalIlluminationFlags.RealtimeEmissive;
```

3. Real-Time VR Implementation

3.1 Rendering & Viewing Algorithms

Navigation (Affine Transformations): WASD translation via `CharacterController.Move(direction * speed * Time.deltaTime)` (5 units/sec base, 7.5 sprint). Mouse rotation: Yaw (Y-axis unlimited), Pitch (X-axis clamped -90°/+90°). Matrix pipeline: World→View→Projection (60° FOV, near 0.3, far 1000).

Culling & Optimization:

- **Frustum Culling:** Automatic BVH (40% draw call reduction)
- **Occlusion Culling:** Baked visibility data (Smallest Occluder: 5 units, 50% outdoor reduction, 35% GPU savings)
- **LOD System:** 3 levels (0-50%, 50-20%, 20-5% screen height), billboard imposters
- **URP Features:** SRP Batchers, GPU Instancing, Dynamic Batching (<300 verts)
- **Result:** 120→55 draw calls, 8ms→5ms frame time

3.2 Illumination & Shading Control

Multi-Source Lighting (136 total):

- **Directional:** Sun/moon simulation (intensity 0.8, cascade shadow maps: 4 levels, 2048² resolution, 150-unit distance)
- **Point Lights:** 72 paper lanterns (yellow), 24 ground lanterns (warm orange), 18 streetlights (white), 14 windows (warm white), 8 interactive objects
- **Shadows:** Cascade Shadow Maps with soft PCF filtering, selective casting (structures cast, props receive only, distant disabled)
- **Reflections:** 6 baked/real-time probes (512²/256² resolution), mixed priority system
- **Glass Refractions:** Alpha blending (0.3-0.5), normal map UV distortion

User-Controllable Illumination (In-World Sliders):

1. **Lambert Model (Diffuse):** Adjusts `_DiffuseIntensity` (0.0-2.0) affecting `NdotL = saturate(dot(normal, lightDir))`. Low values = moody, high = bright saturation.
2. **Phong Model (Specular):** Controls `_Smoothness` (0.3-1.0) and `_SpecularLevel` (0.0-1.0). Modifies highlight intensity via `pow(RdotV, shininess) * specularLevel`. Effect: Adjusts surface glossiness for metallic/glass materials.
3. **Ambient Light:** Global `_AmbientLevel` (0.0-1.0) adjusts ambient occlusion contribution and base scene brightness without direct sources.
4. **Atmosphere & Color:** `_SkyboxTint` modifies fog color, skybox gradient, and overall mood (blue=calm, red=tense). Uses tone mapping for HDR color management.

Implementation:

csharp

```
public void OnSpecularSliderChanged(float value) {  
    RenderSettings.reflectionIntensity = value;  
  
    foreach (Material mat in materials)  
        mat.SetFloat("_Smoothness", Mathf.Lerp(0.3f, 1.0f, value));  
}
```

}

3.3 Dynamic Physics & System Control

Physics Parameter Control:

- **Statue Rotation Speed:** Slider (0-120°/sec) directly modifies rotationSpeed in Transform.Rotate() calculation

Physics-Based Interactions: Blade pickup with parenting (transform.SetParent(handPosition)), gravity toggle, collision-based object manipulation.

Data Visualization (Scalar/Vector):

- **UI Panels:** Real-time bar graphs (light intensity 0-10 scale), vector arrows (wind direction/magnitude), proximity meters (distance to objectives)
- **Debug Gizmos:** Yellow wireframe spheres (interaction radii), cyan arrows (wind vectors), color-coded states (green=active, yellow=interactive)

4. Quest Narrative & Gameplay Flow

Checkpoint Sequence:

1. **Bridge Start:** Player spawns, streetlights OFF, all decorative lights OFF
2. **Shrine Discovery:** Message "Find the sword" (3m proximity to empty blade pedestal)
3. **Book Clue (InteractiveBook.cs):** E press → book rises, clue: "Look under the tree"
4. **Lantern Offering (InteractiveLantern.cs):** Under cherry tree, E press → lantern lights (warm orange), clue: "Go find the statue"
5. **Statue Activation (InteractiveStatue.cs):** E press → continuous rotation begins (30°/sec), looping stone grinding audio, clue: "Look inside the shrine"
6. **Blade Retrieval (BladePickup.cs):** Shrine interior, 5m reveal (yellow glow), 3m pickup ("Press E"), blade parents to hand
7. **Blade Return (ReturnPoint.cs):** Village center shrine, 3m proximity + E key → blade placement, celebration audio

Festival Activation: Sequential lantern illumination (0.1-sec intervals, random order), 72 paper lanterns cascade ON (yellow glow + emission), ground lanterns + house windows activate simultaneously (warm white), celebration light pulses 5x (0.4-sec cycle), message: "Thank you! The festival can begin!" (5-sec display).

Dynamic Response Systems: Pulsing lights (pre-interaction), physics-based wind (continuous), spatial audio layers (ambient + triggered), color-coded UI messages (context-sensitive prompts).

5. Technical Implementation Summary

Core Scripts: ReturnPoint.cs (lighting orchestration, 136 lights), InteractiveLantern.cs (checkpoint + particles), InteractiveStatue.cs (rigid-body rotation), InteractiveBook.cs (rise animation + glow), BladePickup.cs (reveal/pickup logic), FlagWind.cs (physics simulation), PlayerMovement.cs (CharacterController navigation).

Asset Pipeline: Blender models (.fbx export), 18 compressed textures (DXT5/BC7), 5 spatial audio clips (.wav 16-bit), Unity prefab system for modular scene assembly.