

# Sustainable Game Environment Development

## A Resource-Efficient Approach

This project focuses on developing a sustainable 3D game environment inspired by leading stylized artists. By incorporating resource-efficient techniques and aligning with Sustainable Development Goal 12, it achieves a balance between visual appeal and responsible production.

### RESEARCH & INSPIRATION

Initial research focused on exploring stylized environment artists including Alex Treviño's storytelling, and Polygon Runway's stylisation methods

Used Miro boards to gather references and develop concepts, enabling rapid iteration and clear design direction

Implemented rapid prototyping methodology using primitives in Blender for quick composition testing

Balance between visual quality and resource efficiency  
Clear artistic direction combining low-poly and clay aesthetics  
Created efficient workflow from concept to implementation



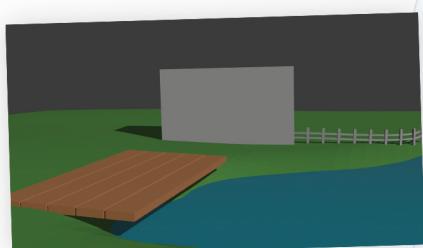
INSPIRATION > BLOCKOUT > FINAL

### DEVELOPMENT & DESIGN

Used Blender's grease pencil feature and primitives to sketch and block out a starting point of the scene. This uses a small amount of resources and is not wasteful in time or energy.

The artstyle was chosen as it is not only efficient and not resource-intensive, but it was suitable for the time-frame and expectation of the project.

Environment exported for use in Unity as a game world. Shaders had to be rebuilt and remade.



### SUSTAINABLE DEVELOPMENT GOAL 12

Sustainable Development Goal 12 promotes responsible consumption and production practices. This project does this through:

Efficient use of polygon density, texture size, and procedurally generated shaders to limit wasted resources and energy consumption.

Reduced processing power needed, less memory usage and smaller filesizes  
- Reduced Digital Consumption  
- Efficient asset creation pipeline, optimised workflow  
- Responsible Production Methods.



### SHADING & LIGHTING

Lighting was handled with a HDRI – requiring less computation power for realistic lighting.

All procedural materials were baked and then recreated in Unity using URP, which is efficient while being able to support high-fidelity shaders.

In Blender, all materials were developed procedurally using the Node Wrangler add-on. This allowed for dynamic, high-quality shaders with efficient resource usage.

The water shader was made entirely using URP shader graph tools.

