Stone Game

The aim for the stone cage game is to attempt to make a very short AAA style game that has good plot, graphics, game mechanics and visuals design while making it creative and unique. To try the best of what i can make give a short time frame of 5 days. In the following I will cover the thought behind the game design then the implementation of game mechanics.

Constrain

Limited time and experience are two things that I will consider before planning my project. To make sure I can safely produce a product that meets my ambitious goal, I made the following choices and compromise.

Art style: I noticed a lot of independent game design have inconsistent art style and caused the game to look cheap I believe its the following reason:

-Out of place assets: because many model assets are used from different model collections, oftentimes the theme and quality are out of place and don't serve the story and are visually out of place.

-Model quality: some assets used have different quality, some are pixelated and some have high definition.

-Poor model placement: model placement that partly sinks or rises above the surface next to makes the game look cheap.

Experience: I had little knowledge and knowledge with unreal engine and game development in general. It's best if I can utilize what I have learned from my previous game.

Resources: Sound editing, and voice acting

Time: 5 days

Personal preference: I don't want world building through text and would like to avoid subtitles in-game text. Because it's really easy to break immersion and feels like a cheap way to explain the game to the player.

Game design:

So for my game, I want to have a consistent art style that could avoid or solve the problem with out of place assets, model quality and poor model placement. Story that makes sense without voice acting and text. While being a very short AAA style game that has good plot, graphics, game mechanics and visuals design, level design, while making it creative and unique.

So I chose stone character and world setting because it has the following characteristic:

Aboudent asset, model, texture of different stones that I can use that works from different sources but within a consistent art style, the model edges mix together easily and the texture gives great detail for the map while hard to distinguish its quality.

Story:

The game is an expiration journey, the players will discover who they are, where they are and what the journey is about. And its soul/ body switching mechanic. The character is in a giant space lab container with a broken wall in front and multiple other giant containers floating above. The character will find out about the trapped giant and its goal for freeing itself and the stone giant from the container.

The main game mechanic that player can utilise is the soul switching mechanic, which the player will utilize it to navigate through map and free itself

Visual

The goal is to have a memorable visual and character design. To do that I will need repetition and simplicity. All the characters have the same model that is a retexture default game engine character model with slight modification of the added yellow core, This saves time to construct a level and the repetition can leave an impression on the play of the character while being a light source, part of map design and serves the plot and the game respawn mechanic. With a simple mixture of two elements, stone body and yellow core.

The giant is also just a modified version of the main character with a larger yellow core that is proportionally larger than the default character. This again serves as a repetition of the character design with simple visuals. Player can easily identify its element and silhouette and remember it design.

Map building(visual):

Nanite is used for detailed 3d surfaces. The intersection between water and stone has a wet transition in the final scene and beginning scene with the player standing next to water.

Stone models are mixed to create a random and natural landscape.

The core of all the characters are used as light sources to illuminate the map and serve as guides to suggest possible navigation for the player. While highlighting its importance of soul(core) switching.

Player respawn

The goal is to simulate soul switching though camera movement and visual indication of the core switched from one playable character to another. Multiple things happen at the same time to achieve this result. I will first describe events that happened then how they are achieved.

As this mechanic is initiated, the current stone character that the player is controlling is destroyed into pieces and player loses control over the current character as the core leaves the current body and

floats towards the closest new stone character. At the same time the camera is launched backward moving farther away from whichever the player is currently facing to simulate the soul leaving the body while locked on to the floating core. As the core gets closer to the new body the camera also moves towards the new body and eventually smoothly reaches the new character's default position to suggest the soul has moved to a new character. The player can now control and play as the new character and the respawn process is complete.

Technical detail

Orb Movement to the New Actor

- Floating Orb (Core):
    - Actor Blueprint with a simple Movement Component
    - Approach:
        1. Interp between the two locations (Old Character → Get New Closest Character).
        2. Use a Timeline to smoothly move the orb.
- Destruction of Old Character:
    - Call DestroyActor on Stone Character A once the orb (core) is launched.
    - Alternatively, use a Fractured Mesh to create a stone-breaking effect before destruction.

Camera Switch & Invisible "Soul" Actor

- Invisible Model:
    - Create an Invisible Actor to attach the camera on.
    - Give it Physics Simulation.
    - The camera is attached to this actor via Set View Target with Blend.
- Camera Motion:
    - Phase 1 (Backward Launch):
        - Use a Homing Missile-style approach so the camera quickly moves away from Stone Character A.
        - The camera remains locked to the orb.
    - Phase 2 (Shift to New Body):
        - The homing destination changes to Stone Character B, causing the camera to move forward and center on the new character.
        - Once close enough, switch the camera to the new character's standard Player Camera.

Homing Missile Logic for Camera Movement

- Homing Component:
    - Unreal has a built-in ProjectileMovementComponent that supports homing to a target.
    - Implementation:
        1. Add a ProjectileMovementComponent to the invisible soul actor.

2. Enable Homing and set the Homing Target to the new character's location.
3. For the second half, dynamically change the homing target to the new camera position or the new character's camera boom.

- Smooth Transition:
    - Use a Timeline that gradually eases the camera from the backward flight to the forward flight, so it looks natural and cinematic.

Switching Control to the New Character

- Possess New Pawn:
    - In Player Controller, call Possess on Stone Character B.
    - This is triggered once the orb reaches a threshold distance to the new character.
- Camera Switching:
    - After possession, Set View Target with Blend to the new character's camera.
    - Set a Blend Time so the transition is smooth.

AI Follow Player

- Implementation:
    - Within the AI logic, whenever a player switch occurs:
        1. On Possess event → Update AI's target to new player.
        2. AI continues to follow or move toward the new player character.

Trigger Box

1. Place a Box Trigger in the level.
2. On Overlap events with the old character, trigger arm movement for giant stone character

Other less complex feature implemented:
Animation for character, moving platform, glowing core, water, natural lighting.