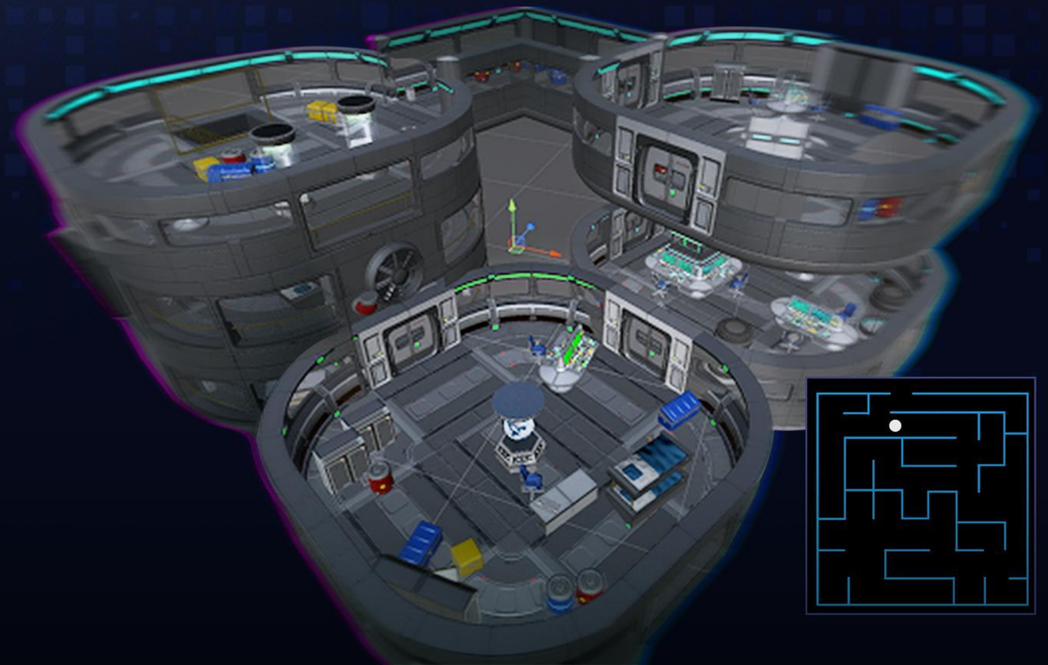
ECHO OF CONSCIOUSNESS

UNITY CREATIVE GAME PROJECT

PERSPECTIVE: 3D Spatial Exploration + 2D Electronic Interface Puzzle



SUMMARY

This project is a narrative-driven puzzle game exploring digital consciousness, brain–computer interfaces, and human identity.

Players take on the role of a test subject who survives a failed human experiment and becomes trapped in an abandoned laboratory. During the experiment, the protagonist gains the unexpected ability to upload their consciousness into electronic systems and manipulate digital door locks.

The game is structured around two interconnected worlds:

- A 3D spatial maze representing the physical laboratory.
- A 2D door-lock maze representing digital systems accessed through consciousness upload.

By constantly switching between these two states, players attempt to escape the facility while confronting fundamental questions about existence and identity.

GAMEPLAY LINK:

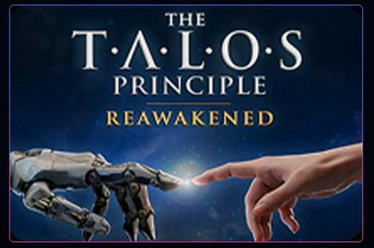
INSPIRATION

This project is inspired by a long-standing philosophical question:

> If a person could be perfectly copied, destroyed, and reconstructed elsewhere through teleportation, would the new individual retain the same consciousness?

GAME REFERENCE

SOMA



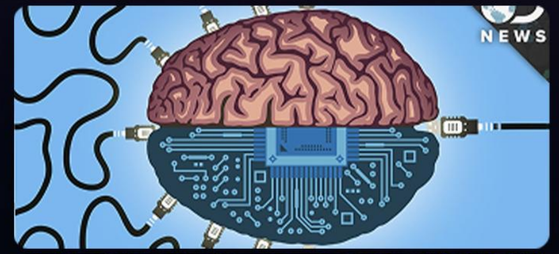
THE TALOS PRINCIPLE

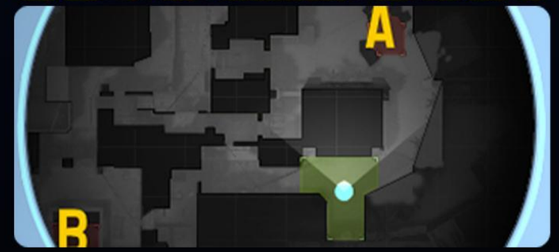
CONCEPT

During early research, I focused on the following areas:

- Science fiction and philosophical discussions on consciousness copying and digital immortality.
- Games that use dual perspectives or parallel worlds to express fragmented identity.
- Interactive designs that treat electronic interfaces as navigable spaces rather than mere UI elements.

These studies led me to transform electronic systems from narrative devices into rule-based, explorable game space.





GAME DESIGN

THE CORE GAMEPLAY IS BUILT AROUND A RECURRING LOOP:

Explore the physical space

Encounter obstruction

Enter the digital system

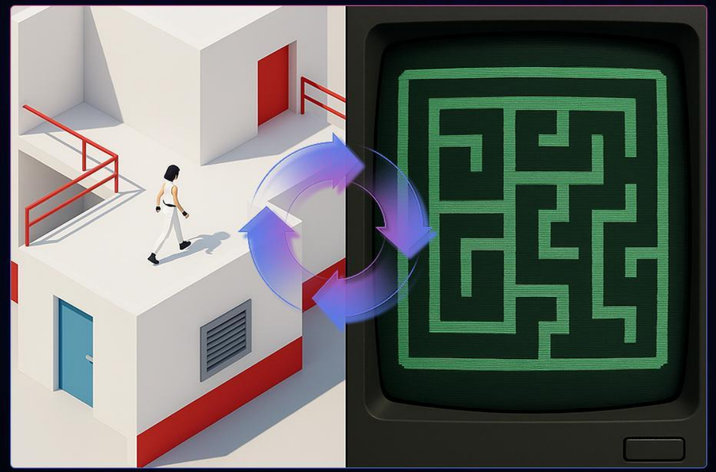
Unlock progression

Return to the physical world

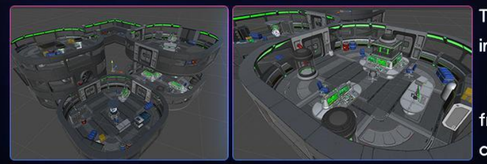
Overall Gameplay Structure

Players navigate the laboratory in a 3D physical form, but progression is repeatedly blocked by locked system nodes.

To advance, they must relinquish their physical body and upload their consciousness into digital systems to solve puzzles.



3D Spatial Maze · Physical World



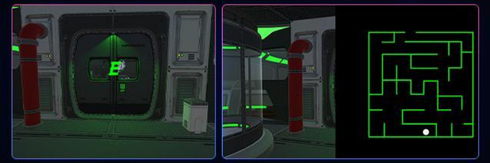
The physical layer takes place in an abandoned laboratory structured as an interconnected maze.

Here, player actions are constrained by spatial and bodily limitations, with frequent barriers such as locked doors. Mistakes can lead to irreversible consequences, emphasizing fragility, uncertainty, and limited control.

2D Door-Lock Maze · Digital World

When encountering locked nodes, players can upload their consciousness into system interfaces, shifting to a 2D abstract maze.

In contrast to the physical world, rules here are clear, feedback is immediate, and there is no bodily risk. Solving these puzzles unlocks corresponding paths in reality.

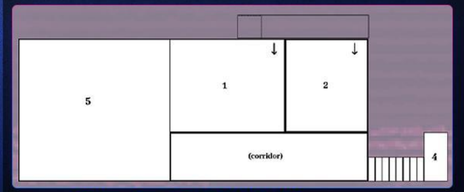


Design Intent

The digital world is intentionally designed to feel safer, more efficient, and more controllable, while the physical world remains unstable and restrictive.

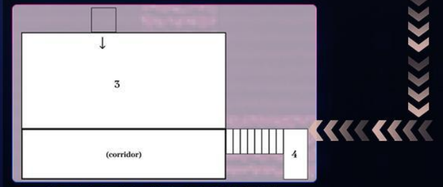
Over time, players are guided to realize that entering the digital space becomes the most rational choice for progression and survival.

LEVEL DESIGN

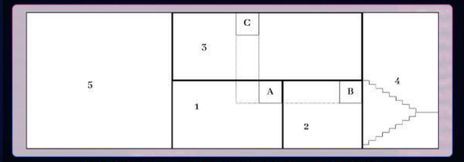


MAP 1F (TOP VIEW)

staircase



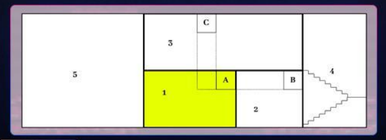
MAP 2F (TOP VIEW)



MAP (HORIZONTAL VIEW)

A, B, C: vents dashed lines: ventilation duct

start → 1 → A → B → 2 → B → C → 3 → 4 → 1 → 5 → end



ROOM 1:

The protagonist wakes up here.

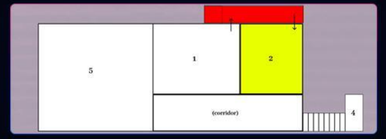
There are shrinking devices that can shrink the main character.

Gravity device, adjust the direction of the scene's gravity to the right.

There is a vent A.

The original direction of gravity.

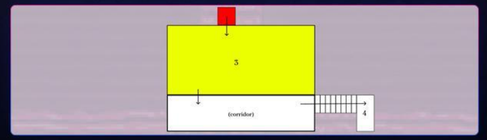
The direction of gravity after using the device.



After the direction of gravity changes, you can step on the wall, and the shrinking protagonist can get into the vent duct and jump down to reach the vent in room 2 (B).

ROOM 2:

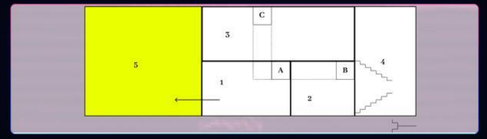
- There is a gravity device.
- Gravity is still to the right after the player comes out of the vent.
- First, bring gravity back to its original state.
- Then turn gravity in direction and jump into the B vent again.



In the ventilation duct, walk from vent B to vent C and climb out to reach room 3.

ROOM 3:

- There is a time device that takes you back to this room ten years ago, and you can bring back an item.
- The protagonist activates the time device and goes back ten years ago to get the ID card and bring it back.
- ID card can open all doors (permissions).
- Room 1 is a factory on the left side through the window on the left, and there is an exit to the outside (roller shutter door, closed).
- After opening the door, go to the corridor and go downstairs.

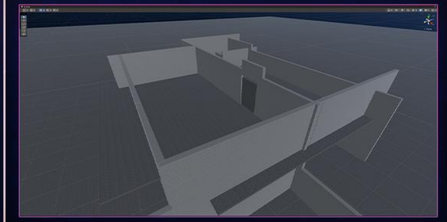


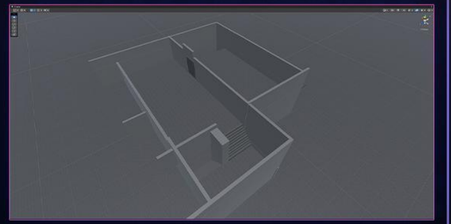
Go back to room 1, open the door on the left, and enter room 5 (factory).

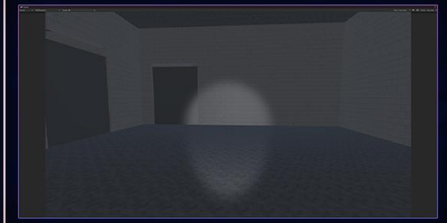
ROOM 5:

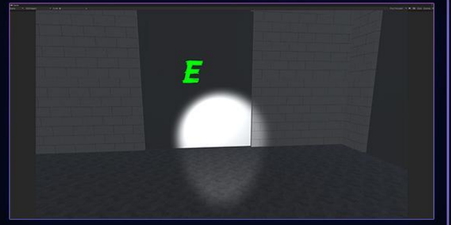
- Roller shutter door failure requires parts.
- Fix the part after locating it.
- There are routers that can connect computer faces to the Internet, connect consciousness in the lab network to the Internet (and so on).
- There is not enough electricity, either open the door or connect to the Internet.
- Players make choices.

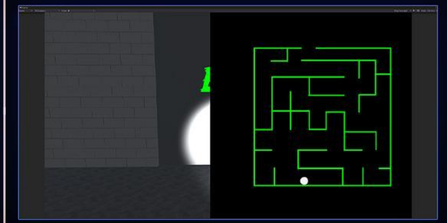
PROTOTYPE DESIGN

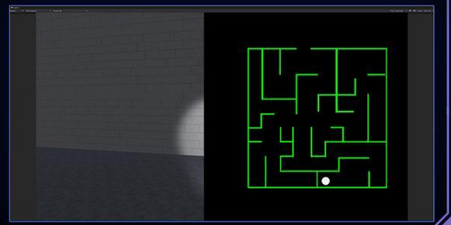






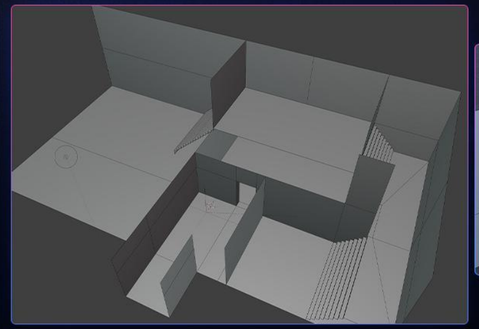




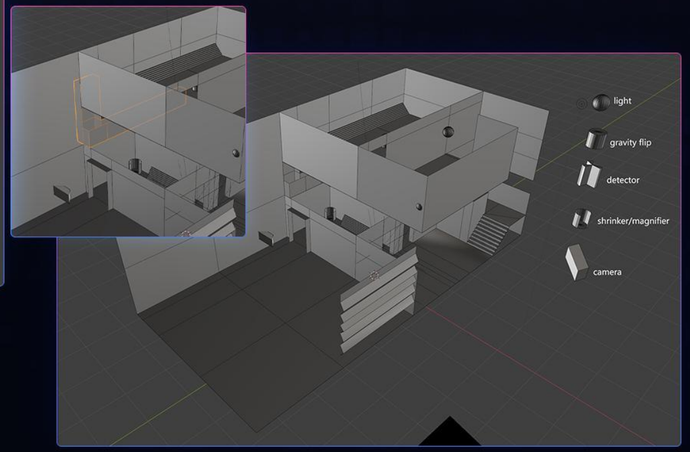


DEVELOPMENT

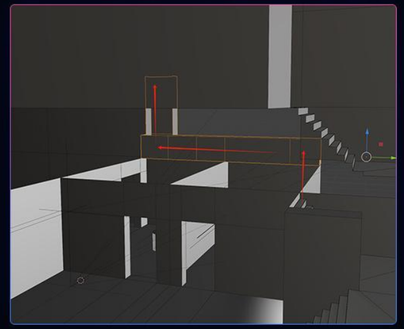
3D Spatial Maze



In the first version of the level design draft, I found that the top-down floor plan couldn't effectively represent the spatial relationships of the two-story building. As a result, I encountered many spatial conflicts when building the first 3D level whitebox based on this floor plan.



Later on, I directly used 3D software to further redesign and optimize the layout based on the initial whitebox.



another way to the second floor through the ventilation duct

Core code sample

```
void Move()
{
    Vector2 input = new Vector2(
        Input.GetAxisRaw("Horizontal"),
        Input.GetAxisRaw("Vertical")
    );

    if (input.sqrMagnitude < 0.01) return;

    input.Normalize();
    Vector2 delta = input * speed * Time.deltaTime;

    TryMove(new Vector2(delta.x, 0));
    TryMove(new Vector2(0, delta.y));
}

void TryMove(Vector2 delta)
{
    Vector2 nextPos = rect.anchoredPosition + delta;
    if (!IsBlocked(nextPos))
        rect.anchoredPosition = nextPos;
}
```

UI MAZE CORE LOGIC

```
void Update()
{
    if (!playerBody.GetComponent<PlayerMovement>().can3DMove) return;

    float mouseX = Input.GetAxis("Mouse X") * mouseSensitivity * Time.deltaTime;
    float mouseY = Input.GetAxis("Mouse Y") * mouseSensitivity * Time.deltaTime;

    xRotation -= mouseY;
    xRotation = Mathf.Clamp(xRotation, minY, maxY);

    transform.localRotation = Quaternion.Euler(xRotation, 0f, 0f);
    playerBody.Rotate(Vector3.up * mouseX);
}
```

MOUSE LOOK CAMERA CONTROL

```
public void OpenDoor()
{
    StartCoroutine(DelayOpenDoor());
}

IEnumerator DelayOpenDoor()
{
    yield return new WaitForSeconds(0.2f);
    door1.OpenDoor();
    door2.OpenDoor();
}

private void Update()
{
    if (isArea && Input.GetKeyDown(KeyCode.E))
    {
        player.GetComponent<PlayerMovement>().can3DMove = false;
        StartCoroutine(SlideIn());
    }
```

HORIZONTAL DOOR

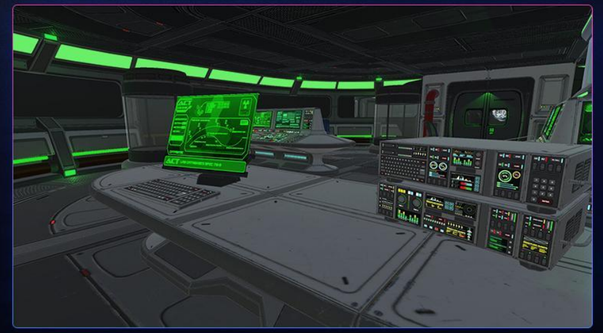
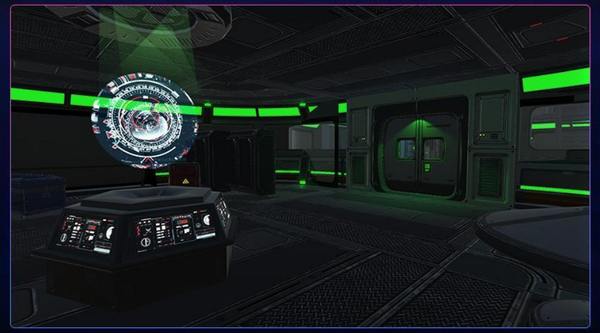
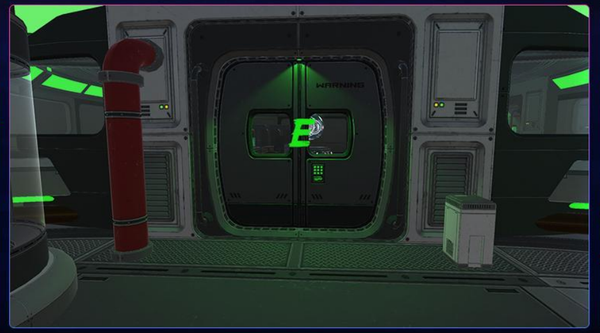
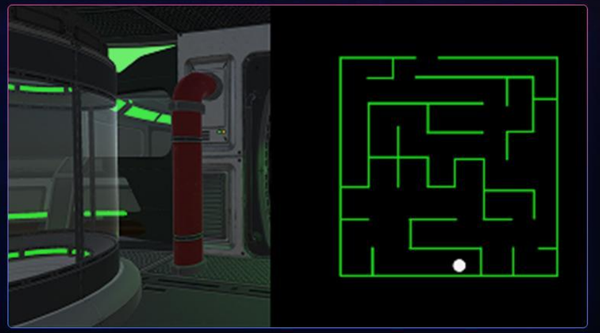
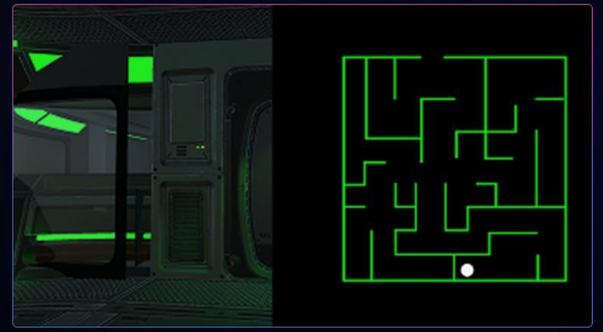
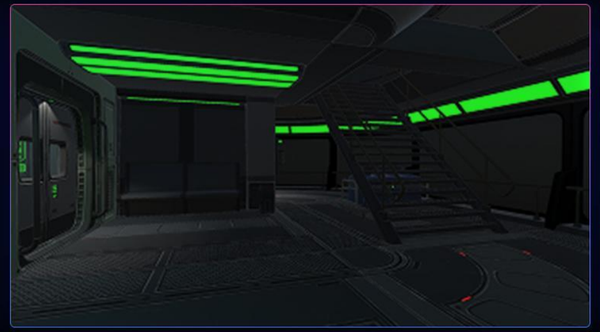
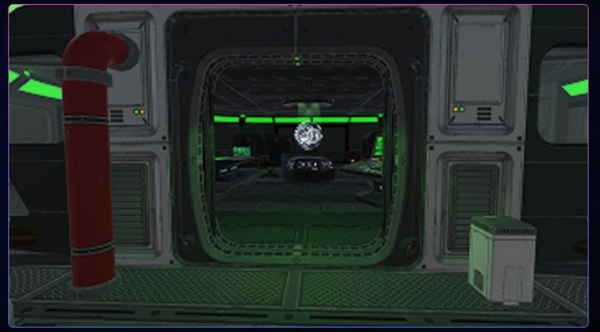
```
void Update()
{
    if (!can3DMove) return;

    float h = Input.GetAxis("Horizontal");
    float v = Input.GetAxis("Vertical");

    Vector3 move = transform.right * h + transform.forward * v;
    controller.Move(move * moveSpeed * Time.deltaTime);

    if (controller.isGrounded && velocity.y < 0)
        velocity.y = -2f;

    velocity.y += gravity * Time.deltaTime;
    controller.Move(velocity * Time.deltaTime);
}
```

3D PLAYER MOVEMENT CONTROL

GAME PLAY