SOLO DEVELOPED PROJECT
IN UNITY

2024.11
/
2024.12

GOODA!M

2D ACTION PHYSICS GAME

PROJECT. 4



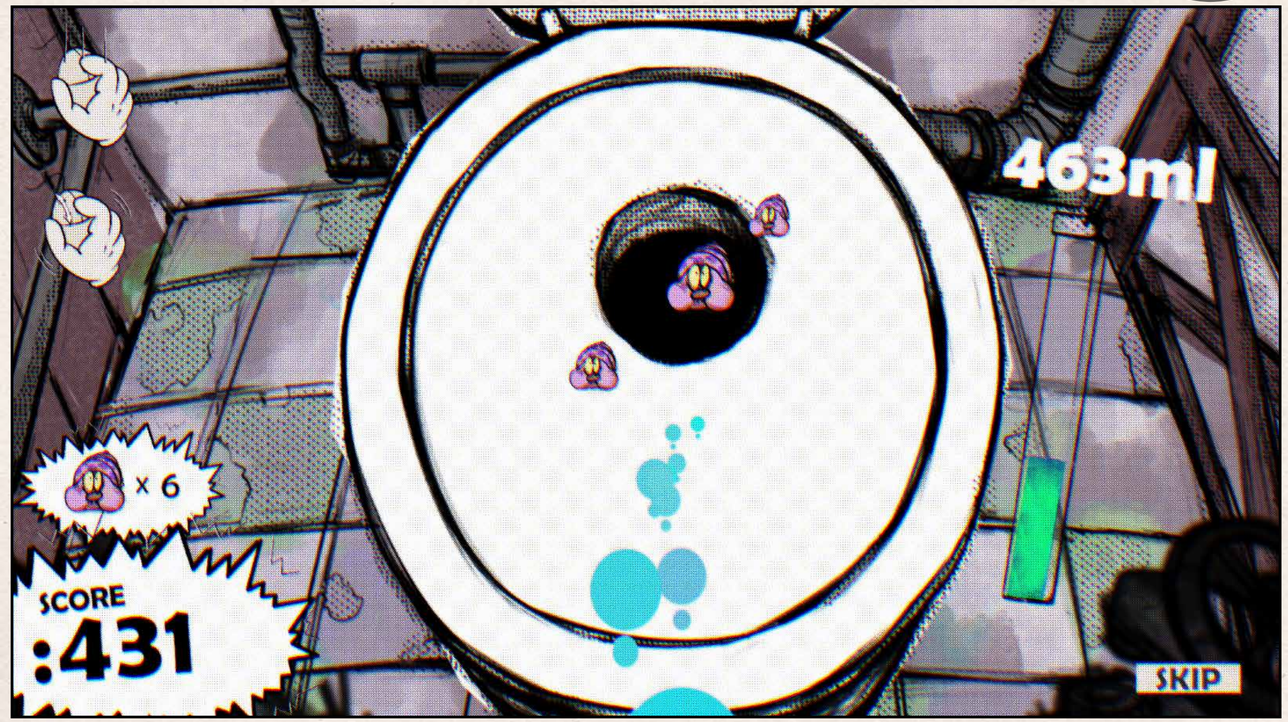
Unity itch.io

GAMELINK: https://xiaoxie33.itch.io/goodaim

Game Preview

PART. 1

Game Screenshots



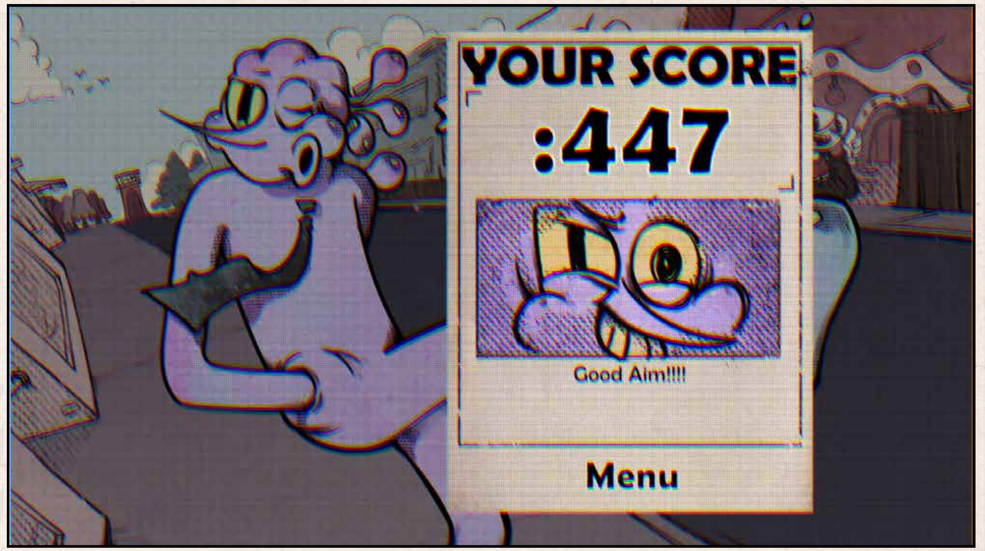
Level Scale

- 9 Levels
- 60 Second
- 1 Player



Ending Screen

Just like most arcade games, players receive a score at the end of the game, and the game will give an evaluation based on that final score!



Different Level

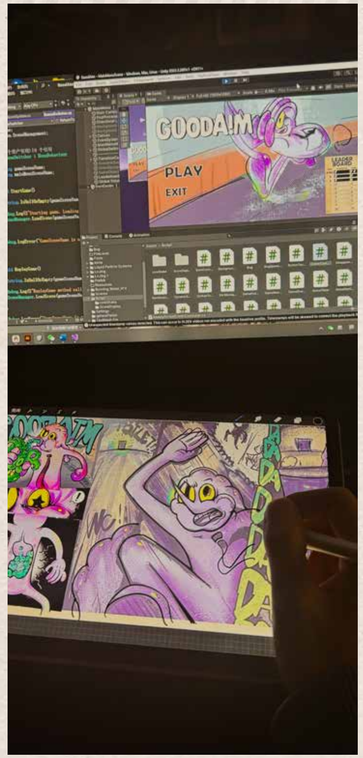
To add more variety to the game and explore additional functions for the "water gun," I introduced a fire-themed level. Unlike the bathroom stage, this level requires players to frequently use the water gun's "spray" feature to extinguish large areas of flames.



CONCEPT TO COMPLETION

Development Process

PART. 2



STAGE 1

On-Paper Sketch for Project Planning

During the project's conceptual stage, I sketched out the game flow on draft paper and explored various ways to showcase its features. In particular, I provided a detailed breakdown of the water gun's functions—such as concentrated and spray modes, random trajectories, and so on—and also laid out the framework for the game's scoring system.

STAGE 2

Developing an Playable Game Prototype

During the early development phase in Unity, I used a rough sketch as the background to test and refine the water gun's trajectory and the game's scoring system. After determining the overall features and scope of the project, I began creating the art assets.

STAGE 3

Initiate the Creation of Art Drafts

I took inspiration for the overall style of the game from one of my favorite games, "Cuphead," incorporating a 1960s American cartoon and comic aesthetic.

STAGE 4

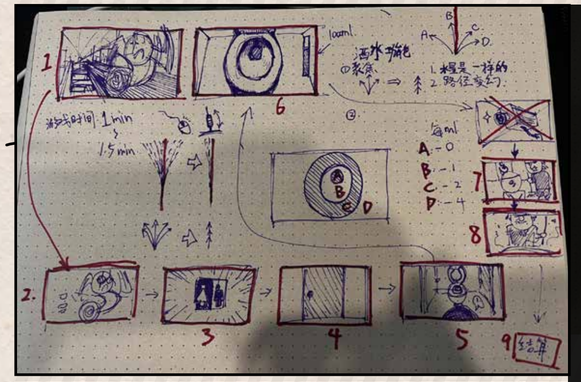
User Interface Design

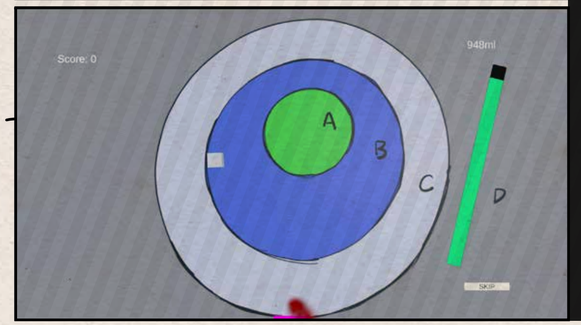
After determining the overall features and scope of the game, I began creating the art assets. I took inspiration from one of my favorite games, "Cuphead," incorporating a 1960s American cartoon and comic-style aesthetic.

STAGE 5

Finish Building the Project in Unity

I completed the construction of each scene in Unity and optimized the display of visual effects.



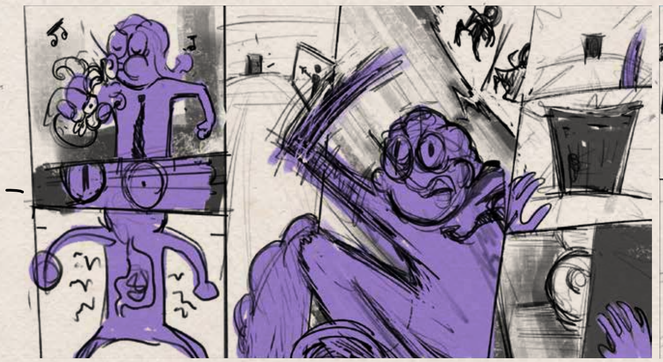


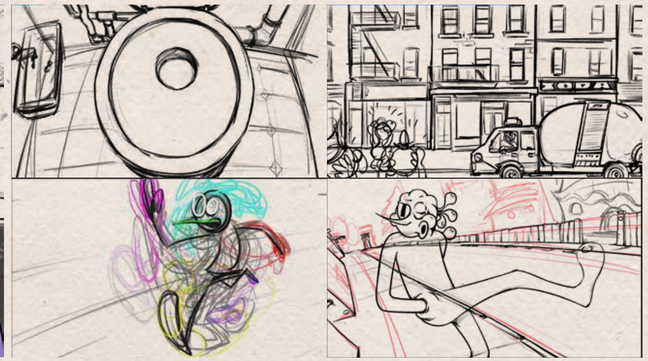
Gameplay Adjustments

In the initial design of the scoring system, I intended to center it around the water gun's "random trajectory" mechanic. Players would need to adjust the firing angle so that the droplets landed at the center of the toilet to earn points. As shown on the left image, each droplet hitting point A would grant +1 point, point B would grant 0, point C would subtract 1 point, and point D would subtract 2 points.

However, during testing, I found that this style of gameplay was extremely dull and overly reliant on randomness. Most players felt frustrated by their inability to control the droplet trajectories, rather than feeling compelled to play another round.

As a result, I reduced the amount of randomness in the trajectory and added " " to require players to make more strategic decisions—whether to boost their score multiplier or accumulate raw points. The remaining element of randomness in the water gun still provides uncertainty in each round, ensuring players stay alert and engaged.

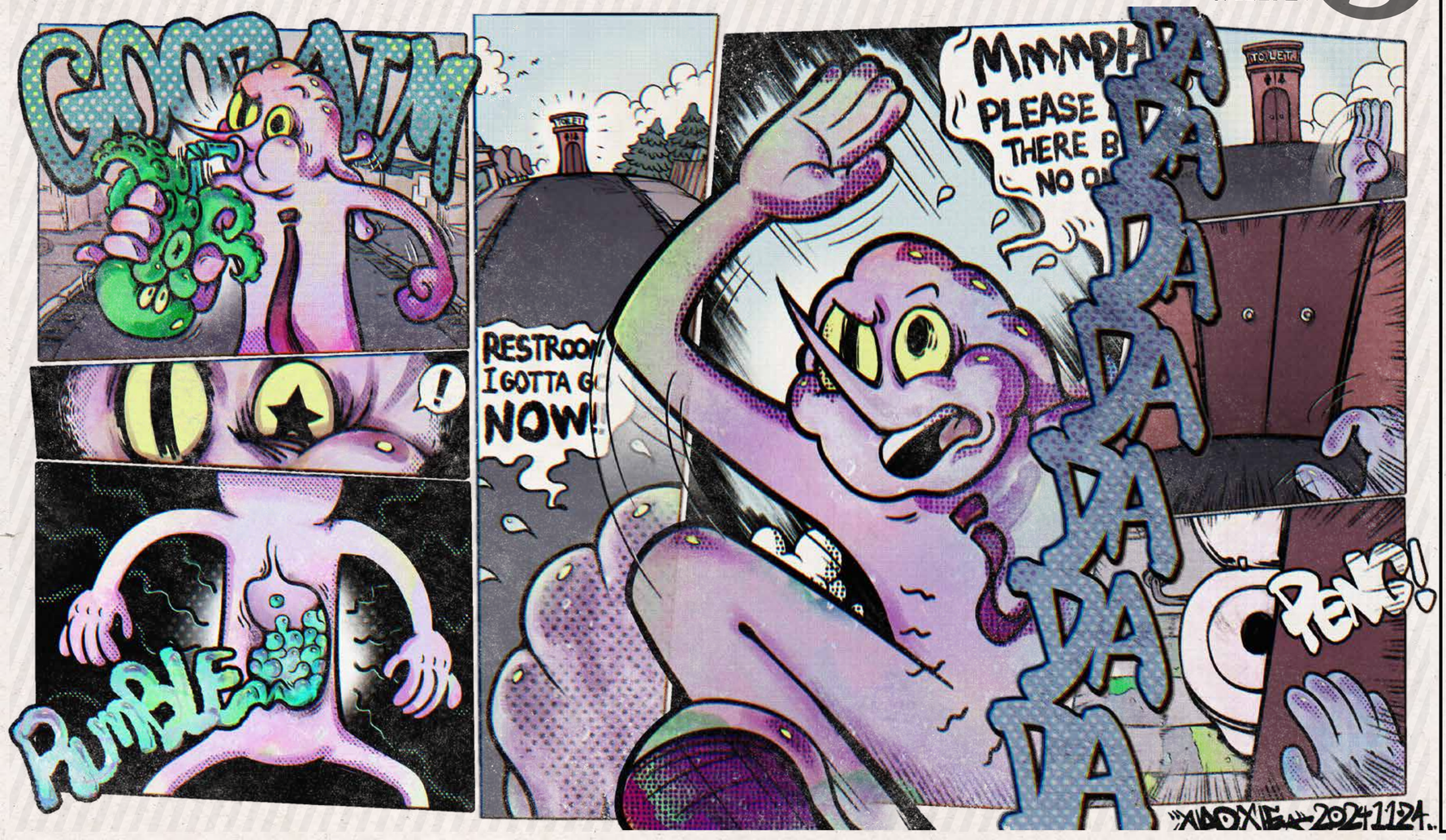






Visual Art

PART. 3





Regarding the game's worldview, I placed the story in a world very similar to ours, where creatures also need to eat and excrete, just like humans. Nevertheless, I wanted the game to have a lighthearted, fun style, so I decided to adopt a 1960s American comic aesthetic.

Worldview and Character

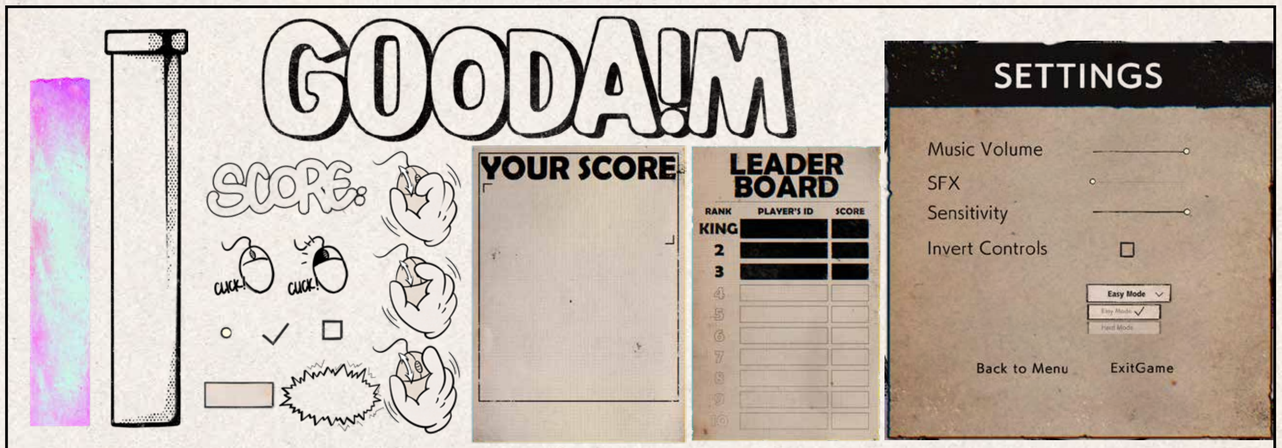
Regarding the character "Aimos," I wanted a main protagonist with a distinctive personality and a bizarre yet memorable design—someone who could carry the entire game's art style and make a strong impression on players. That led me to create "Aimos," who has one large eye and one small eye with different expressions, a long nose, and "water droplet dreadlocks" that were quite painstaking to animate. I really love this character, as he is the soul of the entire game.

PROJECT.4 GOODAIM

UI

PART. 4

User Interface Design







Background

In order to reinforce the character's image, I drew Aimos whistling on the end screen once the game is over, making for a comical scene.





Homage



Block Fire

For the fire scenario, I chose an old neighborhood. In this scene, the flames shrink with each shot, and there's even a trembling crowd of onlookers in the lower-left corner.





Animation Frame

I created an animation of Aimos sprinting wildly toward the toilet for the main menu background.

Main Tools Used

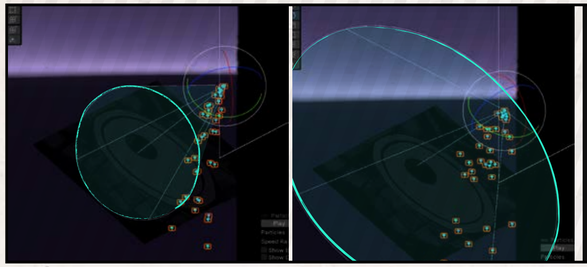
Ps Ai

PROJECT.4 GOODAIM

UNITYINPLEMENT

WaterJet

PART. 5



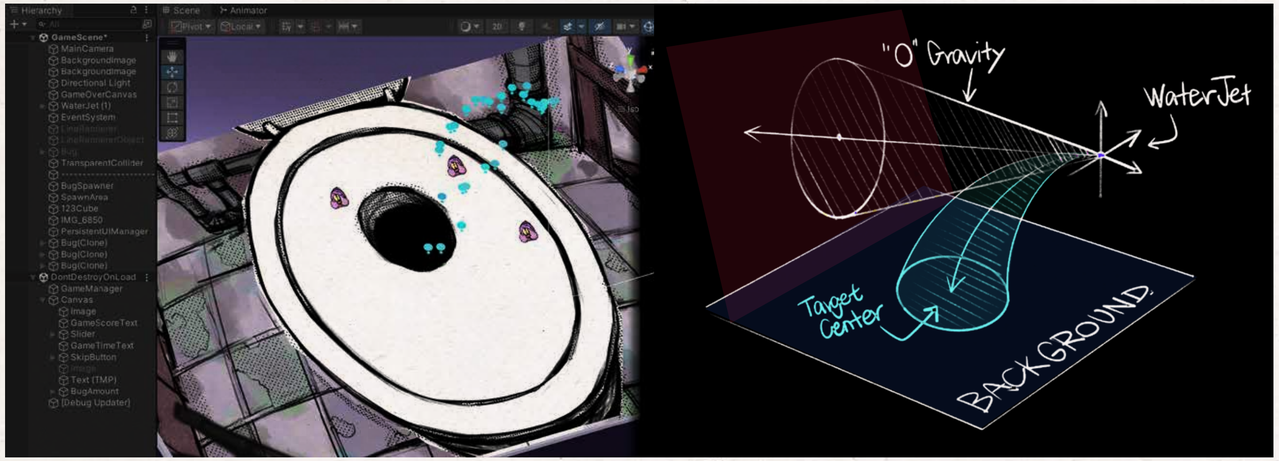
```
void ControlForce()
{
    var shapeModule = waterJetParticleSystem.shape;

    float scrollInput = Input.GetAxis("Mouse ScrollWheel");
    if (scrollInput != 0f)
    {
        currentForce = Mathf.Clamp(currentForce + scrollInput
        shapeModule.angle = Mathf.Clamp(shapeModule.angle
    }

    var mainModule = waterJetParticleSystem.main;
    mainModule.startSpeed = currentForce;
}

public float angleMin = 25f;
public float angleMax = 80f;
public float angleSpeed = 5f;
```

I used the mouse wheel to control both the power of the water gun and the spray's scatter/focus functionality. Scrolling the mouse wheel up increases the water pressure, concentrating the stream into a narrow line; scrolling down decreases the power and broadens the spray's range.




```
1 个引用
void ControlDirection()
{
    Vector3 adjustedInitialDirection = Quaternion.Euler(initialShootAngle, 0, 0) * initialShootDirection;

    Vector3 mousePosition = Input.mousePosition;
    Vector3 screenCenter = new Vector3(Screen.width / 2, Screen.height / 2, 0);
    Vector3 offsetFromCenter = mousePosition - screenCenter;

    float horizontalOffset = offsetFromCenter.x * sensitivity / Screen.width;
    float verticalOffset = offsetFromCenter.y * verticalSensitivity / Screen.height;

    horizontalOffset = Mathf.Clamp(horizontalOffset, -maxRadius, maxRadius);
    verticalOffset = Mathf.Clamp(verticalOffset, -maxRadiusver, maxRadiusver);

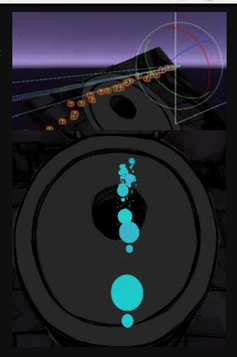
    float targetYRotation = initialShootAngle + horizontalOffset * yAxisRotationLimit;
    float targetXRotation = verticalOffset * -xAxisRotationLimit;

    Quaternion targetRotation = Quaternion.Euler(targetXRotation, targetYRotation, 0);

    float noiseX = (Mathf.PerlinNoise(Time.time * noiseFrequency, 0) - 0.5f) * 2f * noiseAmplitude;
    float noiseY = (Mathf.PerlinNoise(0, Time.time * noiseFrequency) - 0.5f) * 2f * noiseAmplitude;

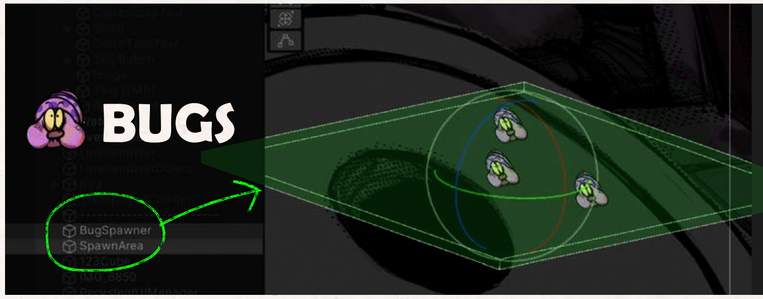
    Quaternion noiseRotation = Quaternion.Euler(noiseX, noiseY, 0);
    Quaternion finalRotation = targetRotation * noiseRotation;

    waterJetParticleSystem.transform.rotation = Quaternion.Slerp(waterJetParticleSystem.transform.rotation, finalRotation, smoothSpeed * Time.deltaTime);
}
```



particle control

Initially, I tried computing the spray's landing point on the background proportionally based on the cursor's position on the screen. However, I ran into a problem where the gravity effect looked extremely odd—it was as if the water droplets behaved like a half-dead fly, spinning around until they finally landed on the target. I then switched to launching the droplets with a particle emitter, which greatly improved the visuals. I still reused the old logic for the water gun's rotation feature, but I imposed limits on the rotation angle so players wouldn't end up spraying large amounts of water outside the visible area.



Small bugs spawn randomly within a green zone in the game. Eliminating a bug doubles your points for hitting the toilet's center! The more bugs you take out, the higher the multiplier goes. As a result, during the short 60-second play session, you have to decide how much time to spend eliminating bugs versus how much time to devote to spraying the toilet center!

PROJECT.4 GOODAIM

GAMETEXTING

PART.

After extensive testing, the game's scoring system underwent numerous iterations. Many thanks to all the friends who tested it and offered their feedback!



GAMELINK: https://xiaoxie33.itch.io/goodaim