Global Game Jam 2025 | Theme: “Bubble” | Engine: Unity

- Type: 48-hour Game Jam project
- Team: Zelin Wang, popilopi, Shutong Zhang, ClaireZ
- My Role: Programmer & Project Management
- Tools: Unity, Arduino (humidity sensor)

🔗 **Gameplay Video**

Overview

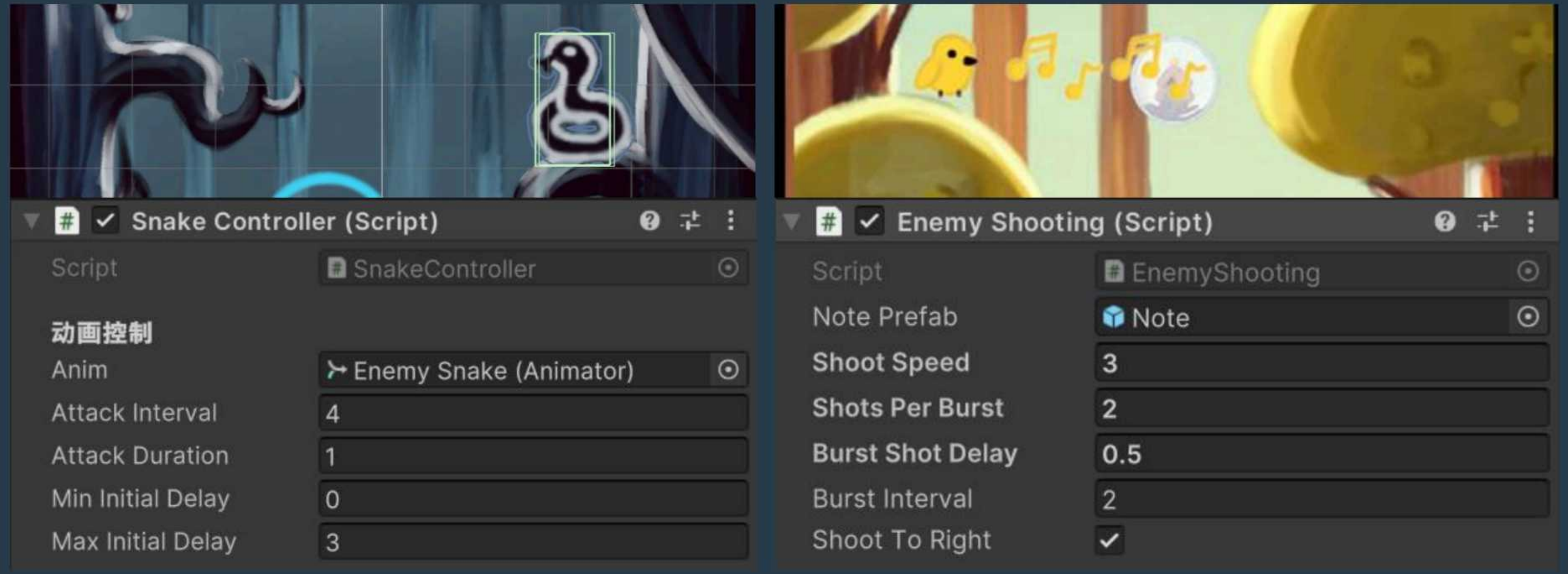
Bupuda is a 2D game made during Global Game Jam 2025 with the theme “Bubble.”

Players blow into a humidity sensor to create wind, which pushes a small bubble carrying a lost princess through different obstacles and helps her return to her home in the sky.

The game mixes the soft and fragile feeling of bubbles with simple gameplay inspired by Flappy Bird. By using real breath as input, it creates a immersive and unique experience.

My Contribution

- Implemented core gameplay systems, including player movement, enemy attack behavior, and camera parallax scrolling to create a sense of depth.



☑ Snake Controller (Script)

Script	SnakeController
动画控制	
Anim	Enemy Snake (Animator)
Attack Interval	4
Attack Duration	1
Min Initial Delay	0
Max Initial Delay	3

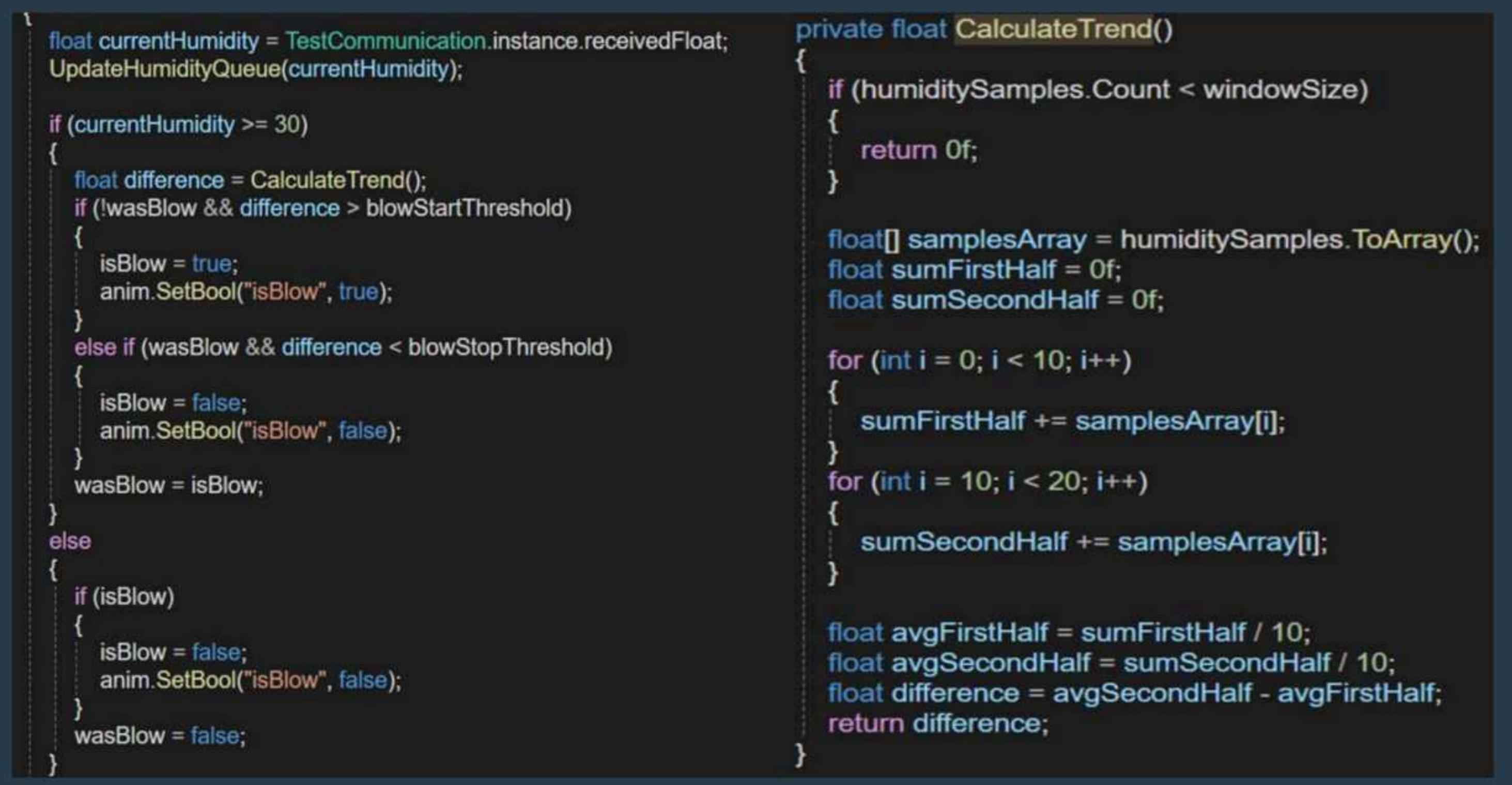
☑ Enemy Shooting (Script)

Script	EnemyShooting
Note Prefab	Note
Shoot Speed	3
Shots Per Burst	2
Burst Shot Delay	0.5
Burst Interval	2
Shoot To Right	✓

Designer-friendly adjustable parameters for flexible enemy behavior

- Developed the breath-based input system, allowing players to control wind by blowing into a humidity sensor connected to Arduino.

Technical Detail



```
float currentHumidity = TestCommunication.instance.receivedFloat;
UpdateHumidityQueue(currentHumidity);

if (currentHumidity >= 30)
{
    float difference = CalculateTrend();
    if (!wasBlow && difference > blowStartThreshold)
    {
        isBlow = true;
        anim.SetBool("isBlow", true);
    }
    else if (wasBlow && difference < blowStopThreshold)
    {
        isBlow = false;
        anim.SetBool("isBlow", false);
    }
    wasBlow = isBlow;
}
else
{
    if (isBlow)
    {
        isBlow = false;
        anim.SetBool("isBlow", false);
    }
    wasBlow = false;
}
```

```
private float CalculateTrend()
{
    if (humiditySamples.Count < windowSize)
    {
        return 0f;
    }

    float[] samplesArray = humiditySamples.ToArray();
    float sumFirstHalf = 0f;
    float sumSecondHalf = 0f;

    for (int i = 0; i < 10; i++)
    {
        sumFirstHalf += samplesArray[i];
    }
    for (int i = 10; i < 20; i++)
    {
        sumSecondHalf += samplesArray[i];
    }

    float avgFirstHalf = sumFirstHalf / 10;
    float avgSecondHalf = sumSecondHalf / 10;
    float difference = avgSecondHalf - avgFirstHalf;
    return difference;
}
```

Designed a sampling algorithm to detect blowing by collecting and averaging 30 frames of humidity data, then calculating differences between time windows to determine state changes, making the detection more accurate.

Outcome