Portfolio

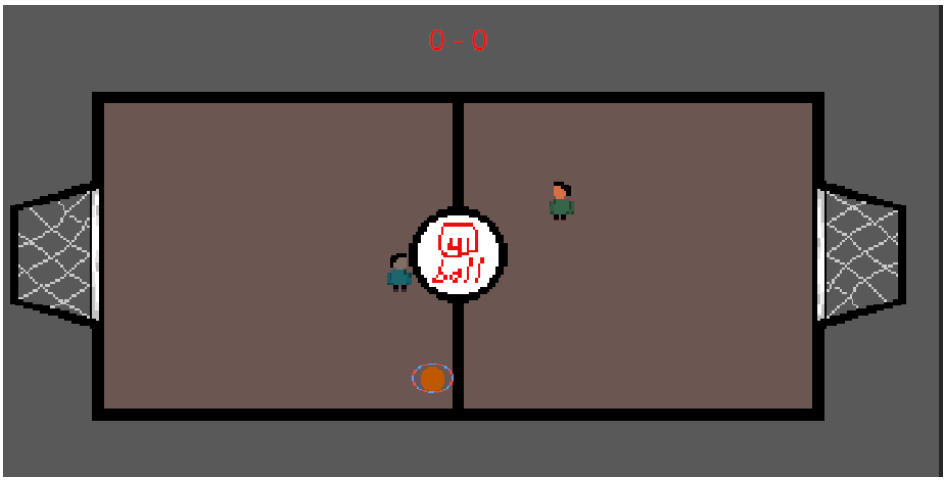
Project1 Punchball2d:

Game Introduction

Punchball 2D is a simple two-player competitive game where the goal is to score by hitting the ball into the opponent's goal. Additionally, players can also target each other to prevent the opponent from scoring.

Concept Design

All the game's artwork was designed and created by me. The game field combines elements of a football pitch and a basketball court, with my custom game logo integrated in the center, forming the main scene of *Punchball 2D.*



Gameplay Design

The inspiration for this game came from an experience I had while playing The *Escapists 2* with friends. In that game, which focuses on escaping from prison, my friends and I ignored the main objective and instead had fun attacking each other using the game's friendly fire system. This "off-task" gameplay was incredibly enjoyable, so I wanted to create an indie game that incorporates player-versus-player combat without making it the primary goal.

Thus, *Punchball* was born—a game where the objective is to score goals while allowing players to attack each other. Players can punch the ball or their opponent to score points, and the first to reach 13 points wins the game.

Technical Details

The gameplay is simple, and all scripts in the game were written by me independently. Below is a partial showcase of the code.

Future Plans

1.Random Events: I plan to add more random events, such as the ball changing size,

slitting into multiple balls, or altering its weight after each goal.

2.Item System: I intend to introduce randomly spawning one-time-use items, such as throwing knives that instantly eliminate opponents, super-powered gloves, or ranged weapons like fist guns. These items will give players additional objectives beyond chasing the ball.

3.Optimization of Ball and Player Mechanics: For example, adding invincibility frames when players spawn, a respawn timer, or randomizing the ball's starting position within the center area. These changes will prevent the gameplay from becoming repetitive after a few rounds and extend the game's longevity.

Link: https://flamberge-backtrace.itch.io/punchball

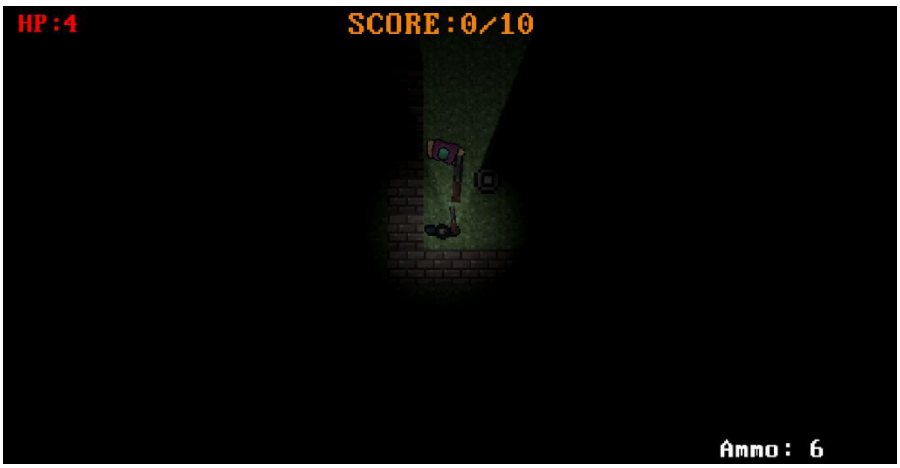
Project2 ShotgunMan:

Game Introduction

Shotgunman is a top-down collection-based game where players must collect enough points to advance to the next level while defending themselves with a shotgun.

Concept Design

The top-down perspective and shooting mechanics are inspired by *Hotline Miami*, while the lighting design draws from the dark and oppressive atmosphere of *Darkwood.*



Gameplay Design

The gameplay is inspired by the many browser games I played as a child. When I was in elementary school, I was captivated by these simple score-based games, often getting scolded by my parents for playing too much. In *Shotgunman*, I aim to recreate that simple joy of scoring points and progressing through levels.

The enemy chase mechanics are similar to Pac-Man, where enemies pursue the player through a maze. However, players can fight back using a shotgun, though ammunition must be found scattered throughout the map. The enemy AI not only chases and attacks but also engages in tactical maneuvers. I plan to add more enemy behavior patterns in the future.

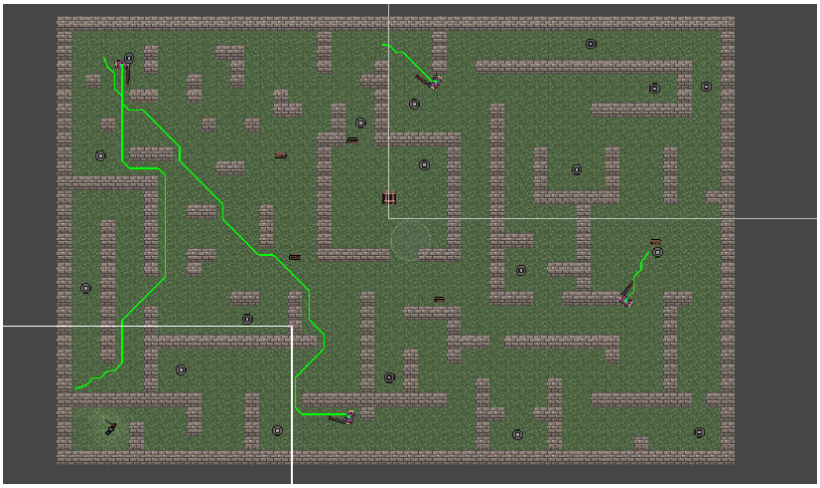
Technical Details

Except for the A* pathfinding algorithm used for enemy movement, which relies on a pre-made Unity plugin, all other scripts in the game were written by me independently. In this project, both the player and enemies are implemented using finite state machines, and health logic is calculated through a universal Character component I created.

```
using UnityEngine;

3 个引用
public class EnemyPatrolState : EnemyState
{
    1 个引用
    public EnemyPatrolState(Enemy _enemy, EnemyStateMachine _stateMachine, string _animBoolName) : base(_enemy, _stateMachine, _animBoolName)
    {
    }

    9 个引用
    public override void Enter()
    {
        base.Enter();
        Debug.Log("Enemy is in patrol state");
    }

    8 个引用
    public override void Update()
    {
        base.Update();
        enemy.FacePathPoint();
        enemy.MoveToPathPoint();
        enemy.changeRandomPatrolPoint();
        enemy.CalculateDistanceToPatrolPoint();
        enemy.currentTarget = enemy.patrolPointList[enemy.currentPatrolIndex];

        if (enemy.distanceToPlayer < enemy.chaseDeterminRange)
        {
            stateMachien.ChangeState(enemy.chaseState);
        }
    }

    8 个引用
    public override void Exit()
    {
        base.Exit();
    }

}
```

Enemy Patrol State Code

Enemy chasing and pathfinding are implemented using the A* algorithm, which automatically selects the shortest route to the target.



Enemy A* Pathfinding to Patrol Point

Future Plans

1.Gameplay Expansion: I plan to transform *Shotgunman* into an infinite-level game similar to Gold Miner. Players will calculate the points needed for each level and decide how to allocate them—whether to upgrade abilities, purchase items, or save points to reduce the pressure of collecting points in the next level. These roguelike elements will significantly enhance the game's fun, and I believe this will be a highly engaging design.

2.More Enemy Types: I intend to add more enemy varieties, such as low-health high-speed enemies, enemies that affect the player's lighting, and others.

3.Player Progression System: Within the current game framework, player attributes and shotgun properties can be easily modified through scripts. I plan to build a player progression system on this foundation, including stat upgrades after each level, and the acquisition of active and passive items.

Link: https://flamberge-backtrace.itch.io/shotgunman

Demo Introduce Video: https://youtu.be/TThoAd3HPrw

P.S. In the link provided, you can download the playable game prototype and the Unity project files.