Anvar Sultanov
2330129@champlaincollege.qc.ca
2330129

Unity Project Checkpoint 01: Ricochet

Project description

Concept

Ricochet is a dynamic 3D first-person shooter game where the core gameplay revolves around ricocheting bullets. Players cannot directly aim at enemies. Instead, bullets bounce off walls, forcing players to anticipate trajectories and have to come up with creative ways to turn the environment to their advantage.

Mechanic

All bullets in the game must ricochet off a wall first and then hit the enemy to kill them. Both players and enemies have infinite ammo but have a cooldown to prevent bullet spamming. The enemies can be killed with a single bullet anywhere on their body. However, 2 enemy bullets are required to kill the player.

Environment

The game happens on a randomly generated and closed arena. The arena will have multiple randomly positioned obstacles like walls and pillars.

Enemies

The enemies are bots that shoot bullets in random directions. The enemies move randomly all over the arena and they spawn in waves, to increase the pressure on the player.

Objective

Survive as many enemy waves as possible, achieving high scores based on precision and survival time.

Anvar Sultanov
2330129@champlaincollege.qc.ca
2330129

Implementation plan

Week 1: Project Setup & Player Creation (CHECKPOINT 2)

- Create a new Unity project
- Set up a basic 3D arena with a flat floor and 4 walls)
- Add a player character (for simplicity I will use a simple capsule for now)
- Implementing player movement (WASD keys)
- Add a first-person camera

Week 2: Shooting & Ricochet Mechanics

- Add a shooting mechanic (press a button to fire bullets)
- Make bullets ricochet off walls
- Add a cooldown to prevent bullet spamming.
- Add visual feedback for bullets (a trail or glow).

Week 3: Enemy AI & Basic Combat

- Create a simple enemy prefab (I will use a capsule for now)
- Implement random enemy movement within the arena
- Add enemy shooting mechanics (bullets also ricochet)
- Make enemies die when hit by a ricocheted bullet

Week 4: Procedural Arena Generation

- Design a system to randomly place walls and obstacles in the arena
- Ensure the arena is always enclosed and playable
- Add variety to the arena layout (like different wall configurations, pillars)

Week 5: Enemy Waves & Spawning

- Create a wave system that spawns enemies in increasing numbers
- Add a timer or condition to trigger the next wave
- Test wave progression and difficulty scaling
- Add a simple UI to display the current wave number.

Anvar Sultanov
2330129@champlaincollege.qc.ca
2330129

Week 6: Health, Damage System & Visuals

- Implement a health system for the player (2-hit deaths)
- Add visual feedback for player damage (screen effects or sound)
- Add lighting effects (e.g., neon glow, dynamic lighting).
- Add particle effects for bullets, explosions, and ricochets.

Week 7: Sound & Music

- Add sound effects for shooting, ricochets, and enemy deaths.
- Add background music that fits the game's tone
- Add a mute button or volume controls in the UI.

Week 8: Final Testing & Polish

- Playtest the game extensively to find bugs or balance issues.
- Adjust difficulty, wave progression, and enemy behavior as needed.
- Add a main menu and pause menu with basic options (e.g., restart, quit).
- Build and export the game