

DOKUMENTASI PROYEK OOP

EchoShift

A. Ringkasan Proyek

EchoShift adalah sebuah game action dungeon crawler 2D di mana pemain menavigasi melalui dungeon yang dibuat secara prosedural, melawan musuh, mengumpulkan rampasan (loot), dan maju melalui beberapa level. Game ini dibangun menggunakan Unity Engine dan dilengkapi dengan sistem UI yang lengkap, termasuk menu utama, fungsionalitas jeda (pause), dan layar game over.

B. Fitur Utama

Gameplay Inti

- **Procedural Dungeon Generation:** Pembuatan level dinamis menggunakan algoritma corridor-first dan room-first untuk menciptakan layout yang unik setiap kali bermain.
- **Combat System:** Pertarungan real-time dengan mekanisme senjata dan efek knockback.
- **Enemy AI:** Musuh cerdas dengan perilaku deteksi, pencarian jalur (pathfinding), dan pertarungan.
- **Sistem Loot:** Item yang dapat dikumpulkan dan properti yang dapat dihancurkan di dalam lingkungan game.
- **Sistem Health:** Manajemen kesehatan untuk pemain dan musuh.

Fitur Teknis

- **Mobile Support:** Touch controls dan input joystick virtual untuk perangkat mobile.
- **Manajemen Audio:** Sistem sound effects dan background music.
- **Run Tracking:** Player progress tracking dan persistensi data.
- **Integrasi API:** Konektivitas backend untuk data pemain dan otentikasi.
- **Dukungan Multi-Platform:** Mendukung build untuk PC & Android.

Sistem Antarmuka (UI)

- Menu utama dengan fungsionalitas login/register.
- HUD dalam game dengan tampilan health dan score.
- Sistem Pause menu.
- Layar Game Over dengan opsi untuk memulai kembali.
- Mobile-friendly controls.

C. Panduan Memulai

Prasyarat

- Unity 2022.3.49f1 atau versi lebih baru.
- Visual Studio atau VS Code untuk pengembangan.
- Android SDK (untuk build Android).

Instalasi

1. Clone repository:

```
git clone <repository-url>
cd EchoShift
```

2. Buka proyek di Unity:

- Luncurkan Unity Hub.
- Klik "Open" dan pilih folder Game.
- Tunggu hingga Unity mengimpor semua aset.

3. Buka scene Menu Utama:

- Buka Assets/Scenes/Main Menu.unity.
- Klik dua kali untuk memuat scene.

Menjalankan Game

- Di editor Unity, tekan tombol **Play**.
- Gunakan menu utama untuk menavigasi opsi permainan.
- Pilih level untuk mulai bermain.

D. Kontrol Permainan

Aksi	PC Controls	Mobile Controls
Movement	Tombol WASD or Arrow Keys	Joystick Virtual (sisi kiri)
Attack	Tombol Kiri Mouse	Tombol Attack(sisi kanan)
Navigasi Menu	Klik Mouse	Input Sentuh

E. Sistem Permainan

Generasi Dungeon

Game ini menggunakan beberapa algoritma untuk menciptakan layout dungeon yang bervariasi, antara lain:

- **Random Walk:** Menciptakan struktur seperti gua yang organik.
- **Corridor First:** Menghasilkan koridor terlebih dahulu, lalu menempatkan ruangan.
- **Rooms First:** Menempatkan ruangan terlebih dahulu, lalu menghubungkannya dengan koridor.

AI Musuh

Musuh memiliki perilaku AI, yang meliputi:

- **Deteksi Target:** Deteksi berdasarkan garis pandang dan jarak.
- **Pathfinding:** Navigasi di sekitar rintangan.
- **Combat Behaviors:** Pola serangan berdasarkan kecepatan dan ukuran efek knockback.
- **Steering Behaviors:** Gerakan yang mulus dan penghindaran tabrakan.

Sistem Pemain

- **Movement:** Kontroler karakter yang mulus dengan gerakan physics-based.
- **Combat:** Penanganan senjata dengan efek knockback.
- **Health:** Damage system dengan umpan balik visual.
- **Animasi:** Animasi karakter saat bergerak dan menyerang

F. Struktur Proyek

Proyek diorganisir dengan struktur folder yang jelas untuk memisahkan skrip, aset visual, audio, dan prefab.

```
Game/
├── Assets/
│   ├── Scripts/           # All game scripts
│   │   ├── Player/       # Player-related scripts
│   │   ├── Enemy/        # Enemy AI and behaviors
│   │   ├── DungeonGenerator/ # Procedural generation
│   │   ├── UI/           # User interface
│   │   ├── Game/         # Game management
│   │   ├── API/          # Backend integration
│   │   └── Components/   # Reusable components
│   ├── Scenes/           # Game scenes
│   ├── Prefabs/          # Reusable game objects
│   ├── Sprites/          # 2D artwork
│   └── Audio/            # Sound effects and music
├── Library/              # Unity cache files
├── ProjectSettings/      # Unity project settings
└── Packages/             # Package dependencies
```

Script-script Utama

- **Movement.cs:** Mengontrol gerakan pemain.
- **Player.cs:** Komponen utama pemain.
- **Enemy.cs:** Kelas dasar untuk semua musuh.
- **LevelManager.cs:** Mengelola progresi level.
- **AudioManager.cs:** Mengelola semua audio dalam game.

G. Panduan Build untuk Platform

Build Android

1. Buka **Build Settings** (File > Build Settings).
2. Pilih platform **Android**.
3. Konfigurasi **Player Settings**:
 - Atur **Package Name** (contoh: com.example.echoshift).
 - Konfigurasi signing keystore.
 - Atur **Minimum API Level** ke 22 atau lebih tinggi.
4. Klik "**Build**" untuk membuat file APK.

Build PC

1. Buka **Build Settings**.
2. Pastikan **PC, Mac & Linux Standalone** terpilih.
3. Pilih arsitektur target (disarankan x86_64).
4. Klik "**Build**" untuk membuat file executable.

H. Integrasi API

Game ini memiliki konektivitas backend untuk fitur-fitur berikut:

- **Otentikasi Pengguna**: Sistem login dan registrasi.
- **Data Pemain**: Pelacakan kemajuan dan statistik.
- **Run Records**: Riwayat lengkap semua run pemain dengan sistem score dan peringkat

Backend dibangun menggunakan Java Spring Boot dan menyediakan endpoint RESTful untuk mengelola data pemain.

I. Database Schema

Game ini menggunakan database PostgreSQL dengan tabel-tabel berikut:

- **players**: User accounts dengan username dan password.
- **player_sessions**: Active user sessions dengan authentication tokens.
- **runs**: Rekaman game run masing-masing pemain dengan score, time, and level data.

J. API Endpoints

Authentication

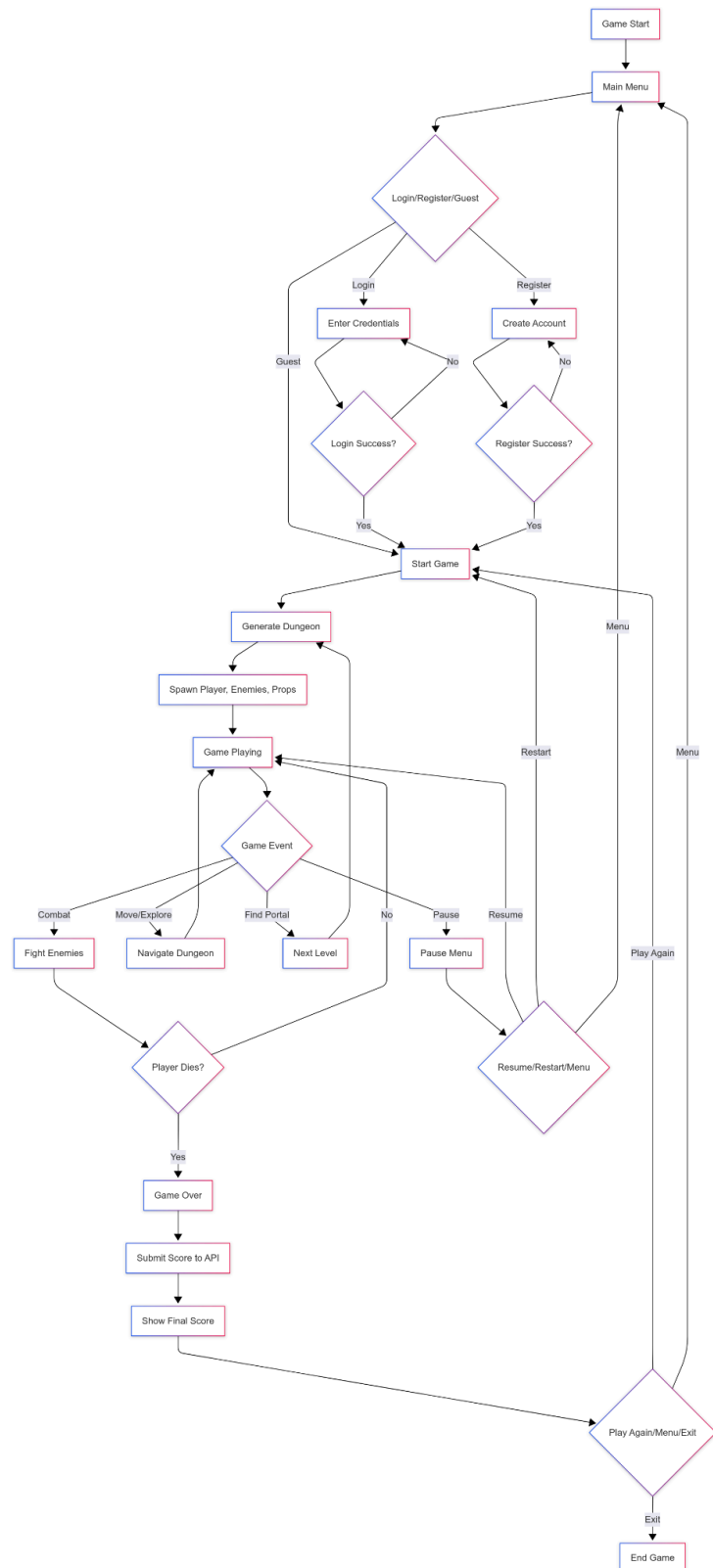
- **POST /api/auth/register** - Register new player account
- **POST /api/auth/login** - Player login with credentials
- **POST /api/auth/logout** - End player session
- **GET /api/players/session/validate** - Validate current session token

Game Runs

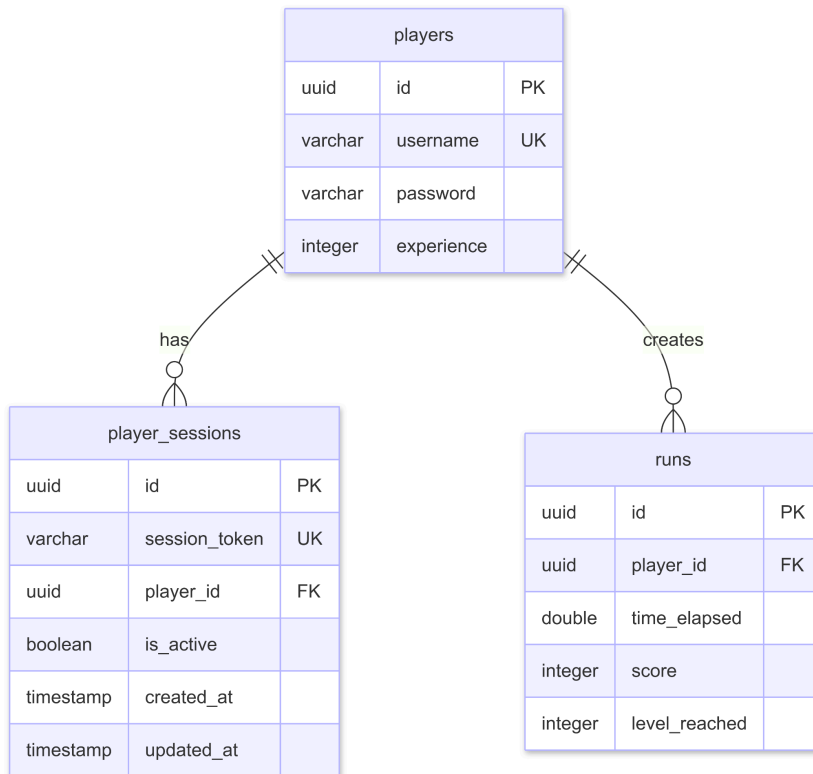
- **POST /api/runs** - Submit new game run data
- **PUT /api/runs/{playerId}** - Update run progress during gameplay

- PATCH /api/runs/{id}/end - End game run and finalize score
- GET /api/runs/{id}/end - Get all runs for specific player

K. Flowchart



L. Entity Relationship Diagram



M. Screenshots



