Building A Game:
Procedurally Generated Roguelike-Dungeon-Crawler

Key Features of Roguelikes & Dungeon Crawlers:

These elements define the structure and appeal of roguelike/dungeon crawler games:

- Procedural Generation – Unique dungeon layouts on every run

- Real-Time or Turn-Based Combat – Engaging player VS enemy systems

- Permadeath (Optional) – Adds tension and consequence to failure

- Loot & Pickups – Rewards exploration and encourages replayability

- Player Progression – Through items, skills, or power-ups

- Enemy AI – Varying patterns or states of behavior

- Replayability – No two runs feel the same
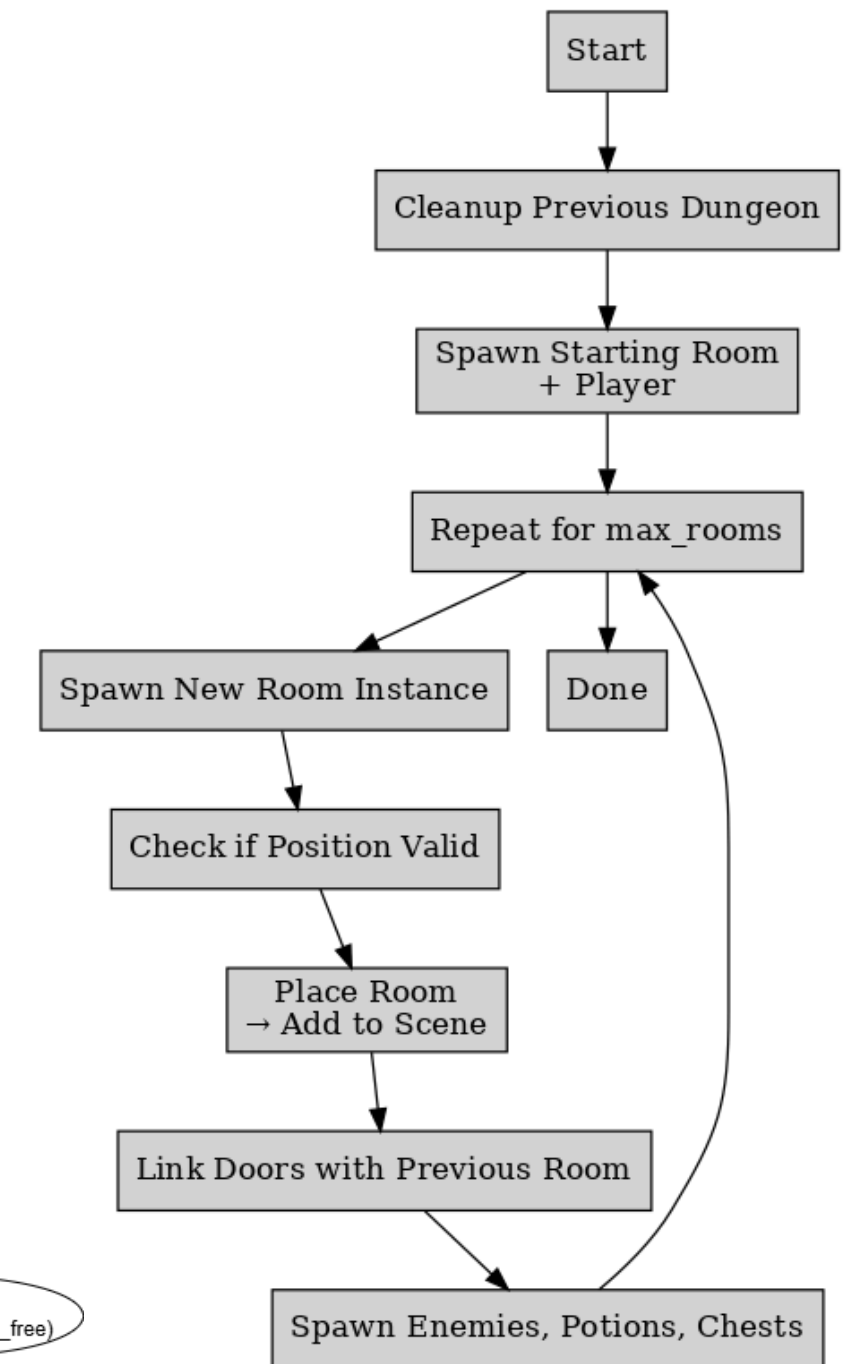
- Project Aims and Motivations -

To create a 3D roguelike dungeon crawler with the core focus on procedurally generated dungeons, real-time combat and enemy AI.

The goal is to achieve an engaging, replayable gameplay loop with strong technical foundations and fun for the player!

Procedural Generation:

- Modular Grid-Based Approach
- Designed for fast runtime generation with flexible expansion

```
Start
  ↓
Cleanup Previous Dungeon
  ↓
Spawn Starting Room + Player
  ↓
Repeat for max_rooms
  ↓
Spawn New Room Instance        Done
  ↓
Check if Position Valid
  ↓
Place Room → Add to Scene
  ↓
Link Doors with Previous Room
  ↓
Spawn Enemies, Potions, Chests
```

Enemy AI:

- Enemy behaviour is governed by a finite state machine.

IDLE (Sleeping/Floor) — See Player → WAKING (Play animation) — Awake → CHASING (Move to player) — In Range → ATTACKING (Melee or Cast Spell)
Lose Sight → IDLE
Out of Range → CHASING
Health <= 0 → DEATH (Play death, queue_free)

Score: 0
High Score: 1900

Prithvi Sathyamoorthy Veeran
– Third Year UG Student –
✉ ZLAC206@live.rhul.ac.uk

ROYAL HOLLOWAY UNIVERSITY OF LONDON

GODOT
Game engine