# Game Design Document (GDD): Dude

## 1. Game Overview

***Title:*** Dude

***Genre:*** 2D Roguelike / Strategy

***Platform:*** PC / WebGL (Unity)

***Engine:*** Unity

***Perspective:*** Top-down tile-based grid

***Target Audience:*** Players who enjoy turn-based roguelikes and emotionally immersive gameplay experiences.

***Core Idea:***

Dude is a turn-based roguelike game where players navigate procedurally generated dungeons while managing food (energy), avoiding enemies, and reaching an exit. The core gameplay is designed to evoke tension, anxiety, and satisfaction through deliberate pacing and resource pressure.

## 2. Gameplay Mechanics

***Core Loop:***

- Player makes a move → loses 1 food point.
- Player can collect food/soda to replenish.
- After the player's turn, enemies move.
- Player aims to find the exit to progress to the next level.

***Player Actions:***

- Move in 4 directions (WASD or Arrow Keys).
- Break destructible walls.
- Pick up food/soda.

***Health Attributes:***

- Fruit: +10
- Drink: +20
- Walk: -1
- Wall hit: -1 (to break a wall- total 5 hits)
- Enemy hit: -10, -20

***Enemies:***

- Move toward the player.
- Skip every other turn on early levels.
- Attack the player if adjacent and movement is blocked.

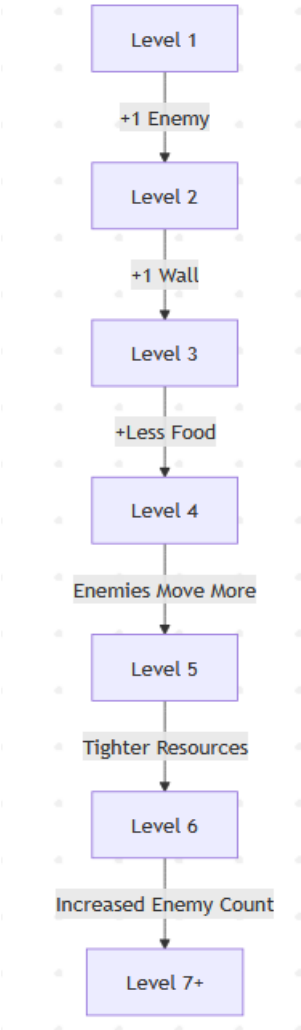
***Win/Lose Conditions:***

- **Win:** Reach the exit.
- **Lose:** Food reaches zero (starvation).

## 3. Level Design

- Procedurally generated 14x7 grid.
- Random wall, enemy, and food placement each level.
- Exit is always placed in the top-right corner.
- Increasing enemy count based on level difficulty.

***Difficulty Curve Diagram:***



## 4. Visual Style

- Pixel art, minimalist and clear.
- Dark-toned backgrounds with glowing pickups.
- Basic tile-based environment with modular prefabs.

## 5. Audio Design

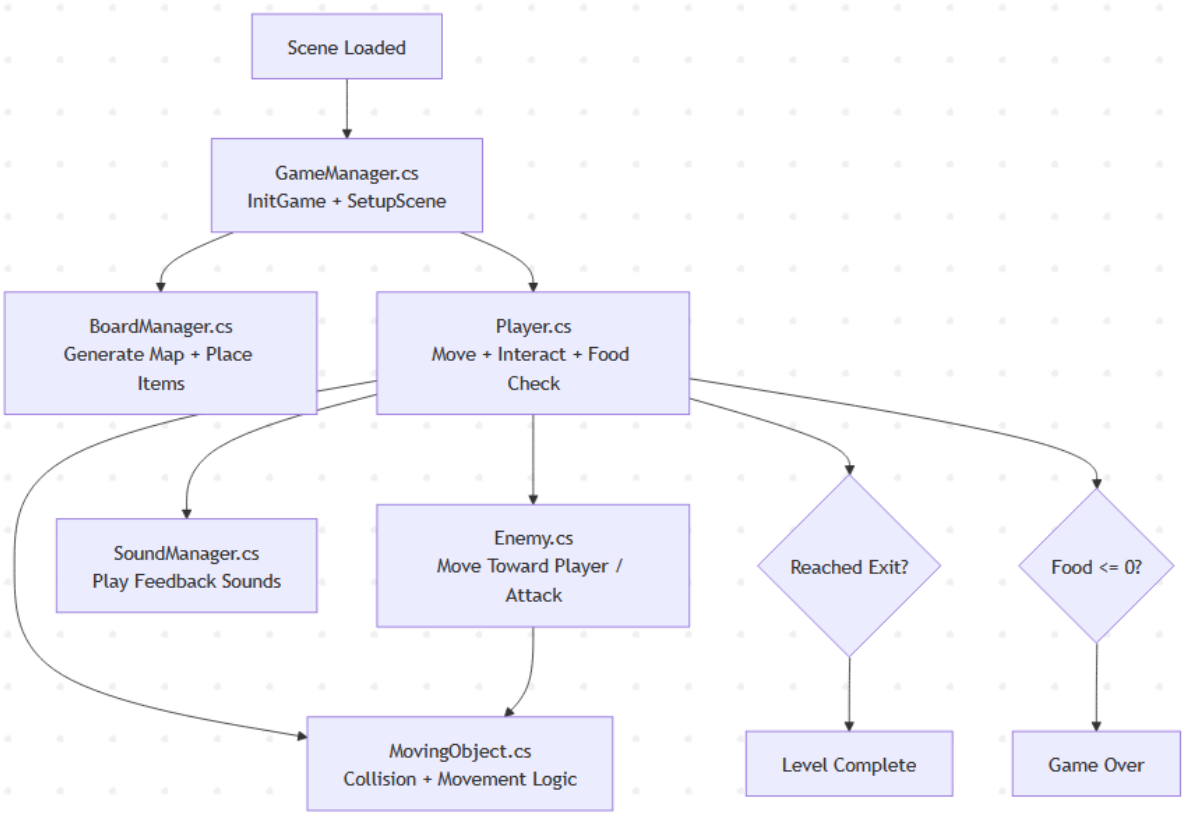
- Movement, eating, and combat have randomized SFX.
- Sound feedback enhances interactivity.
- Death and victory signal game state.

## 6. Technical Architecture

***Scripts:***

- GameManager.cs: Handles level flow, turns, and game over.
- BoardManager.cs: Generates the map and places objects.
- Player.cs: Manages input, movement, and interactions.
- Enemy.cs: Handles enemy AI and turn-based behavior.
- MovingObject.cs: Shared movement logic.
- SoundManager.cs: Manages audio playback.

***Code Flow Diagram:***




## 7. User Interface

- HUD shows food count.
- Level intro and game over overlays.
- Minimal UI to keep focus on grid.

## 8. Development Roadmap

- Core prototype built using Unity tutorial.
- Affective pacing tested through procedural generation and balance.
- Future additions: audio layers, lighting, new enemy types, deeper narrative.

# 9. Team

- **Chinmay Kawale** – Testing, Development, Reasearch, Game Design Support
- **Adityaraj Singh** – Game Design, Presentation Design, Reasearch, Testing