

# #025 Level Design - Process - Combat - Enemy Design

Prototyping different types of NPCs with specific behaviors, strengths, and weaknesses  
From *The Level Design Book*.

## ▼ #025 Level Design - Process - Combat - Enemy Design

- *What is enemy design?*
- ▼ *Enemy roster*
  - *Balancing a roster*
- *Enemy design advice*
- ▼ *How to design an enemy*
  - ▼ *1. Define core enemy design*
    - *HEALTH*
    - *RANGE / BEHAVIOR*
  - *2. Blockout test arena*
  - *3. Prototype enemy*
  - *4. Tune enemy*
  - *5. Let it rest, return later*
- *Boss design*
- *To review...*

## What is enemy design?

**Enemy design** is about prototyping different types of hostile NPCs with specific behaviors, strengths, and weaknesses.

Enemy design is often a complex collaboration between AI designers, programmers, character artists, animators, sound designers, etc. Which is far outside the scope of this level design book.

Nonetheless, level designers must be able to conceptualize and theorize about enemy design, even if they're not the developers tasked with implementing the enemies.

# Enemy roster

Imagine we're making an action game. Here are some basic roles to fill on our **enemy roster**, the collection of enemy types and the combat roles they fulfill.

Role	Behavior
Grunt	Close range melee attack player straightforward, easy to pull
Squad	Attack from mid-range / long range, ideally take turns as a group
Leader	High survivability, buffs nearby allies somehow
Tank	High survivability, but slower and larger (easy to hit or avoid)
Swarm	Small fast attacker with low health but high close-range damage
Sniper	Slow weak long range attacker, very vulnerable without others

A finished working enemy NPC often requires designers, artists, and programmers to collaborate. It can be a lot of work, and it's far outside the scope of this level design book.

Here we will focus on the combat designer / level designer's responsibility to advocate for gameplay features and readability:

- Unique **silhouette** so players can discern enemy type at mid-range (~10+ meters away).
- Design details and animations that **telegraph** enemy state, intent, and strengths / weaknesses.
  - example: in Skyrim, the giant feels big and slow, you must out-maneuver it and wear it down

## Balancing a roster

It can be useful to organize the enemy roster with a spreadsheet / matrix:

Role	Speed	Health	Range	DPS
Grunt	<i>Medium</i>	<i>Weak</i>	<i>Close</i>	<i>Medium</i>
Squad	<i>Medium</i>	<i>Average</i>	<i>Mid</i>	<i>Low</i>
Leader	<i>Medium</i>	<i>Average</i>	<i>Mid</i>	<i>Medium</i>
Tank	<i>Slow</i>	<i>Strong</i>	<i>Close</i>	<i>High</i>
Swarm	<i>Fast</i>	<i>Weak</i>	<i>Close</i>	<i>Low</i>

Role	Speed	Health	Range	DPS
Sniper	<i>Slow</i>	<i>Weak</i>	<i>Long</i>	<i>High</i>

This can help balance the roster or point to gaps, but don't let the spreadsheet trap you into just one way of thinking. For example, in the matrix above:

- Only 1 Fast enemy? Maybe Leader should be Fast too, to feel more different vs Squad?
- There is no High-DPS Mid-Range enemy. Should there be? Would that be fun or useful?
- Is this matrix design too simple? Do we need a Size column, a Behavior column? etc.

To analyze the enemy roster in an existing game: play many levels, and then make your own spreadsheet with your own columns.

- What kind of terrain / placement works best for each enemy type?
- Which enemy types work well together? Why?
- Which enemy combinations are present in which levels?

## Enemy design advice

Some general industry consensus / advice on designing enemy types and rosters:

- Diversity. Each type should feel different to fight, avoid similar enemies.
- Hierarchy. Some types are weaker, some are stronger. Make this ranking clear.
- Longevity. Types should not become obsolete. A weak early-game enemy should still be interesting later on, or a mid boss should transition to become a regular type, etc.
- Emergence. Different combinations of enemy types should create different situations. The same tactics, used against each type in isolation, should be less effective.
- Intelligence. Give the illusion of intelligence and different emotional states (afraid, angry, etc.)
  - "Smart" enemies should have high survivability / health, or else they'll die before they can show off how smart they are.
- Consistency. Feel predictable, with clear patterns for the player to recognize and exploit.

## How to design an enemy

### 1. Define core enemy design

For most 3D action games, each enemy design needs to answer these questions, at least:

- **Health:** how strong is this enemy, how long can it usually survive?
- **Speed:** how fast does this enemy move? Slower or faster than the player?
- **Damage:** how dangerous are the enemy's attacks?
- **Range / Behavior:** does the enemy attack at close range or mid range / long range?

For an initial design, specific numbers are less important because these numbers will likely change drastically. It is enough to estimate "10% / 50% / 100%" or "low / medium / high", and tune the specific numbers later when you actually begin prototyping and playtesting.

## HEALTH

How many resources must the player spend to disable this enemy?

Variations on health: armor / shields, stun / stagger

## RANGE / BEHAVIOR

For the 3D action game Skylanders: Spyro's Adventure, Activision-Blizzard designer Mike Stout talks about how the design team conceptualized 4 main archetypes ("Near, Far, Swarmer, Heavy") as well as their behaviors. He argues melee enemies by themselves are usually pretty boring, and ranged enemies are important for level design:

"Far Enemies are super simple, they're just dudes that shoot at you. But they're very interesting because environment matters [here]. When you have Near Enemies, since they can't attack you from far away, your environment doesn't matter at all: you got them up on a ledge, behind cover -- it's all the same as fighting on a flat plane. So adding Far Enemies to a setup, and mixing them with other types, means all of a sudden your level design is going to start to matter."

-- Mike Stout, GDC 2012: "Reaching Into the Toy-Chest: A Look into Skylanders: Spyro's Adventure's Design"

Variations on range:

- Projectile design (speed, movement pattern)
- Size (a small distant thing is harder to notice and attack vs. a big distant thing)

## 2.Blockout test arena

- One big box room is often all you need, don't spend more than a few minutes on this
- For ranged enemies, also build raised ledge / gap / cover variations
- Simple basic all-purpose pattern for flanking and cover: iceworld 3 lane

### 3. Prototype enemy

- Create a placeholder object
  - Colored shapes like blocks and capsules that slide around
  - Pre-made assets, reuse existing character art but with a different color
- Script the behavior
  - if the game engine or tool has a robust scripting system or AI system, you may be able to prototype the enemy without editing game code directly
  - but usually, you will need to do some game programming
  - common AI scripting patterns: state machines, behavior trees, or a bunch of if() blocks
- Playtest the enemy prototype
  - If possible, get console commands / debug menus for spawning even more enemies in-game at runtime

### 4. Tune enemy

Once the enemy is functional in-game, your work has only just begun. You must now tune the enemy's stats and behaviors to fulfill your experience goals.

**For early tuning passes, make only big changes.** Do not make small adjustments because you won't be able to notice such small adjustments at this early stage.

### 5. Let it rest, return later

The full enemy design process can take many years over the course of the entire project, as you gradually add new enemies, weapons, player progression, levels, etc.

Your project team may suddenly cut a half-finished enemy type to meet an important deadline. Or maybe you'll need to demote a boss design to a weaker reusable "midboss" type. A perfectly-tuned enemy may suddenly feel obsolete or broken. This is just the nature of creativity and game design: the project will change and develop.

Example: HL2 chopper

## Boss design

- Basically the same process as general enemy design but now it's a big setpiece
- Except you should also keep iterating on the arena blockout while you prototype the boss

- A boss doesn't have to be a new enemy type, it can be an existing enemy type with special scripting, or even a puzzle designed around an NPC. Any climactic setpiece with conflict and danger will feel like a boss fight.

## To review...

- **Enemy design** is a lot of work and usually requires collaboration with artists and programmers to create a usable finished in-game enemy.
- Enemy rosters help you balance enemy types within an ecosystem. Avoid redundant enemies which do not have a clear purpose or use case.
  - Diversity is important. You'll likely need a mix of weak and strong, slow and fast, small and big, close range and long range, and everything in-between.
- How to design an enemy:
  1. Define core stats like **health, speed, damage,** and **range / behavior.**
  2. **Blockout** a big box test arena, but don't spend more than a few minutes making it.
  3. Prototype the enemy and **playtest**, you may need scripting or programming knowledge.
  4. **Tune** the enemy; try big changes, avoid small adjustments when prototyping.
- **Boss design** is a special case, a unique setpiece encounter tied more closely to a specific level. Keep iterating on the boss arena blockout.

**write by Hellinus - 2023.9.21**