GPE :

Aggressive GPE :

This GPE has static enemies and works with a battery system.

To engage the fight with this GPE, players need to search and find a connected battery and destroy it. The turrets are immunised to damage as long as the player hasn't destroyed all batteries. Battery are protected by enemy waves

All aggressive enemies find to look at player when he’s come at the defined range, and it’s follow the player with this canon

Battery :

Battery have health and is connected at one aggressive GPE

Industrial Summoner :

The industrial summoner creates swarm enemies, these enemies can detect and follow the player.

- Summon every of (x) seconds
- Have (x) battery (Can be changed during game progression)
- Have (x) Health point
- Each GPE can have an array of swarmer enemies summonable

Save & Checkpoint system :

Save :

The save system is launched when the player triggers a checkpoint, which saves a number of parameters :

- The location where player are when he’s overlap a checkpoint
- The health state
- The ultimate point stocked
- The waves state a spawner linked

Checkpoint :

Checkpoint is temporary saved where the player can respawn when’s his died and launch a real save when player trigger it :

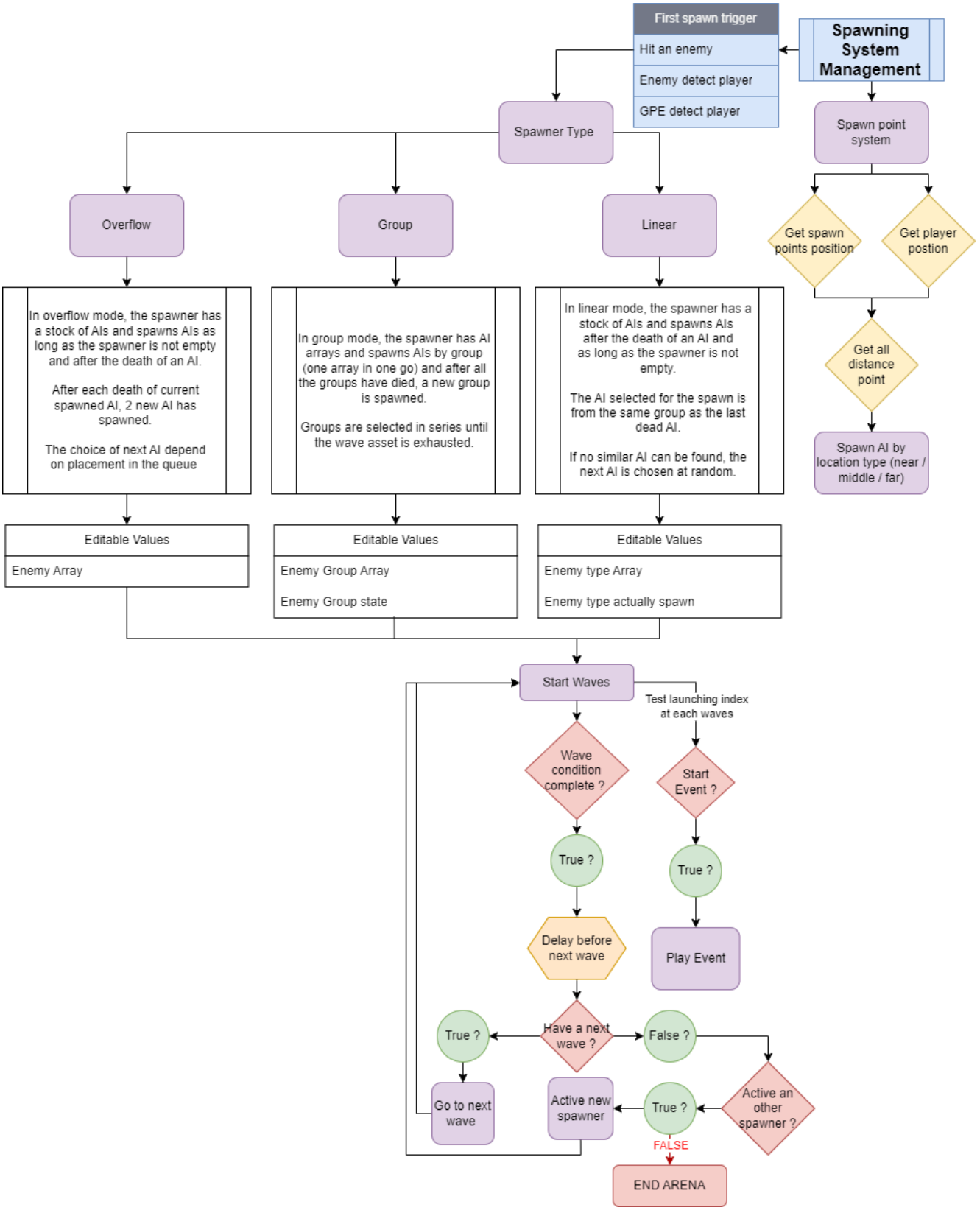
- The checkpoint save location where player respawn, his health, and his ultimate point stocked
- A checkpoint is deactivated after first overlap
- When checkpoint is triggered, a soul ammo spawn to refill his Soulfator

Player load and respawn :

If the player wants to respawn without dying, he can return to the last checkpoint or restart the level again.

Load at last checkpoint load last player state saved, and restart level recreate a new save

Spawner GPE :



Modes :

Spawning modes have a different method of spawn, this method has attributed data assets. Each method have one particular data asset type

Overflow :

Overflow is a spawning type which is used to create combat, with tension rising with each new spawn.

- Data asset composition
    - Table of enemies
    - Number of enemies in the first spawn group (is a quantity of enemies spawned simultaneously "**count variable**")
    - Value of the number of enemies added after a kill

Behaviour of wave :

- Generates enemies according to the **count variable** of the first enemies
- After each kill, the following wave generates a quantity equal to the kill value.
    - The enemies chosen are the next index on the data asset array
- When the data asset is totally clear, the wave end

Group :

Group is a spawning type which is used to create combat, with voltage peak.

- Data asset composition
    - Table of enemies
    - Array of table of enemies
    - Count of enemies remaining before the next spawn

Behaviour of wave :

- Spawn the first group of enemies in the array group of data assets
- When the current array have a count of enemies alive equal to defined value the next wave spawn
- When the data asset is totally clear, the wave end

Linear :

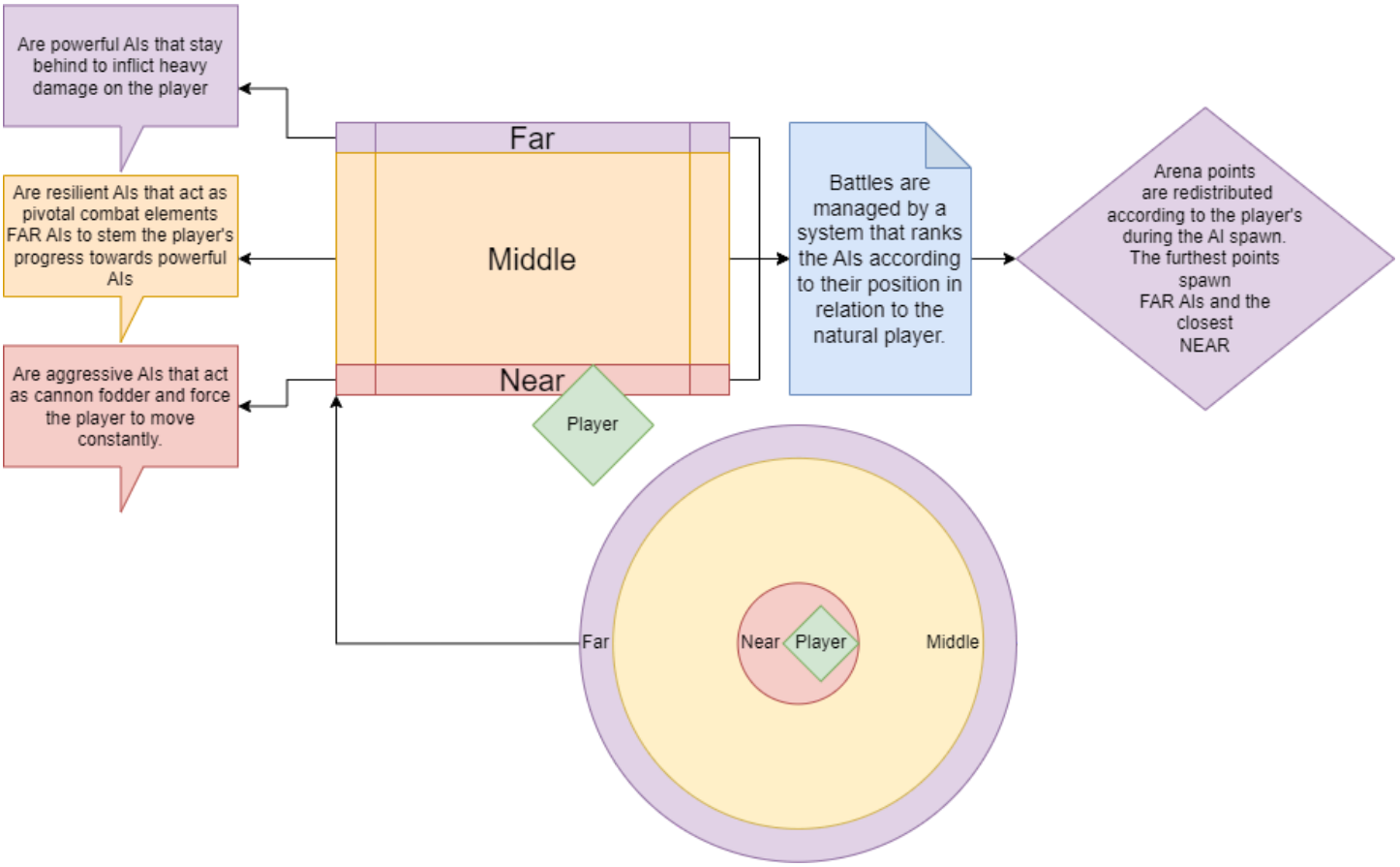
Linear is a spawning type which is used to create, with the same voltage peak during all fights.

- Data asset composition
    - Tank of enemies (number of enemies that will spawn from each enemy type)
    - Number of each types of enemies spawned simultaneously

Behaviour of wave :

- Spawn group type composition
- With each kill, a new enemy is created, equivalent to the last dead enemy.
    - If no one of equivalent type is currently available, a random type is selected in other tank of enemies
- When the data asset is totally clear, the wave end

Spawning location :



The process to spawn enemies needs to take the position of the player and the distance between the player and each point of the spawner.

The enemies spawn depending on the distance ratio of all points in the distance array.

- Near enemies type spawn on the first 20% low distance point saved
- Far enemies type spawn on the last 20% far distance point saved
- Middle enemies type spawn on the other 60% distance point saved

The point distance was recalculated before each spawn and ordered to the less distance at the high distance.

AI fight manager (manage current fight):

This manager is used to manage AI in combat behaviour. It works with attack ticket system and global group fight system.

Global System behaviour :

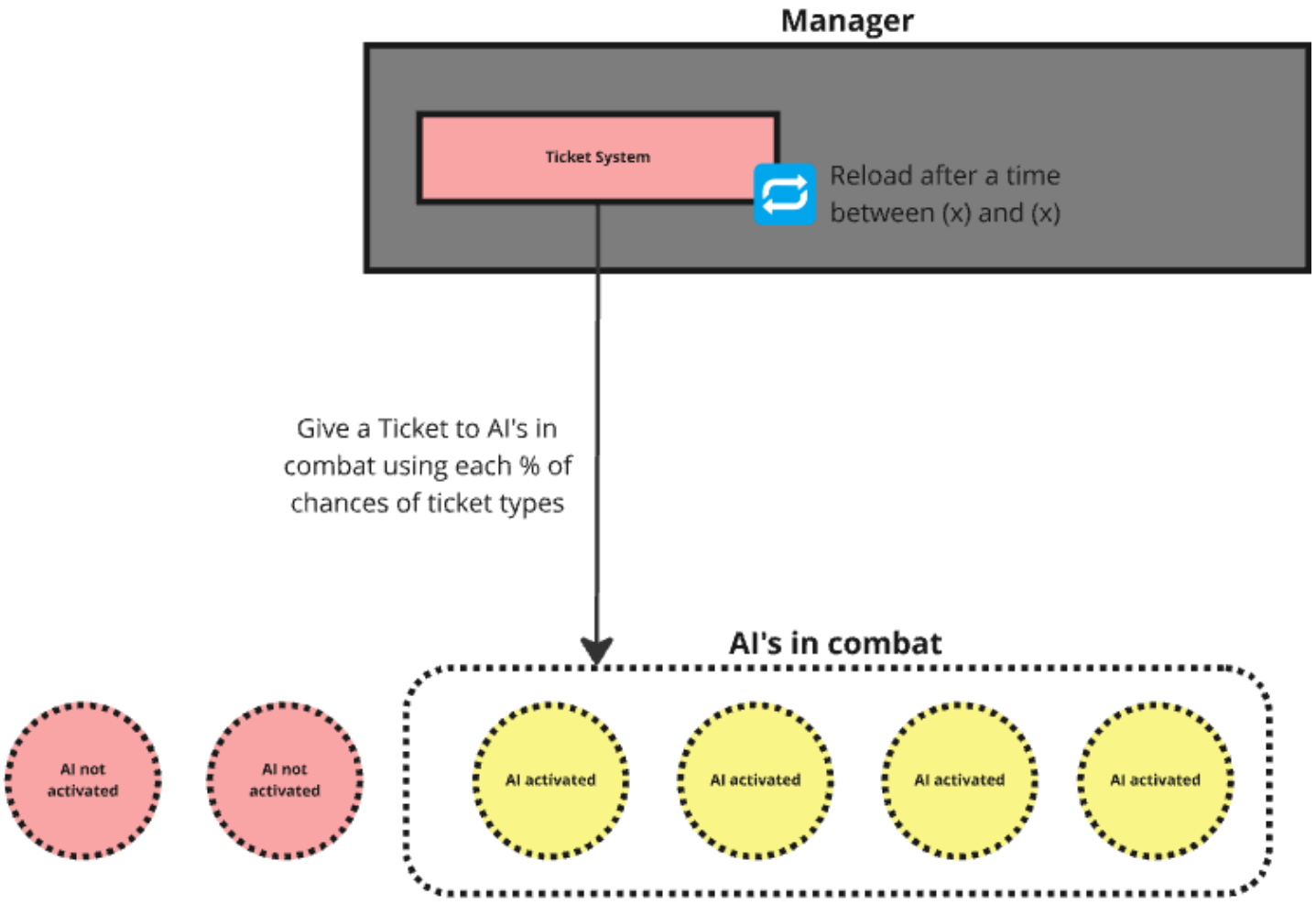
The AI group has rules to manage their behaviour between us.

- ***Avoiding path behaviours :***
  - The AI changes path if it is at a shorter distance between (x) and (x) *(chosen randomly when the AI is activated or spawned)* from another AI if a path is available.

- ***Flee behaviours :***
  - If the AI reaches a number of HP lower than (x) and the player is attacked by another AI, the AI takes cover or is at a distance equal to (x) from the player to wait for healing

  - Behaviour and location are calculated at launch and are not recalculated until the AI is on the location

Ticket system :



All values can be defined in the manager, with the value of each ticket distributed as a percentage. The total of percent distributed is equal to 100%

- ***Dumb ticket :*** this ticket modifies the AI's behaviour so that it ignores the global group rules and rushes towards the player in a straight line while playing these behaviours.

- ***Static ticket:*** this ticket modifies the AI's behaviour by making it move to a position far from the other AIs and fire these patterns until the player is out of range. The AI first moves to a distance equal to half its range.

- ***Bypass ticket:*** this ticket causes the AI to attempt to flank the player without attacking**.**

- **No Ticket :** this ticket has only global rules and AI have his properly classic behaviours

When the AI receives a ticket, the ticket system is restarted randomly between (x) and (x) seconds.
- If an AI spawns after a ticket has been delivered, it defaults to "No ticket".

Shell Kaboom and Tel-Explosion are not affected by the ticket system.

Combat Behaviors are priority so if an AI receives a ticket during a combat behavior it has to end its combat behavior before doing the ticket.

So in their behavior trees, AIs have this Selector for the Ticket System