

maintainAICount

An eventCall to maintain a certain quantity of respawning idAI2 in an idEncounterManager.

Usage

```
item[0] = {
    eventCall = {
        eventDef = "maintainAICount";
        args = {
            num = 8;
            item[0] = {
                eEncounterSpawnType_t = ""; // spawnType
            }
            item[1] = {
                int = 5; // desired_count
            }
            item[2] = {
                int = 10; // max_spawn_count
            }
            item[3] = {
                float = 5; // min_spawn_delay
            }
            item[4] = {
                int = 1; // min_ai_for_respawn
            }
            item[5] = {
                entity = ""; // spawnGroup
            }
            item[6] = {
                string = ""; // group_label
            }
            item[7] = {
                float = 10; // max_spawn_delay
            }
        }
    }
}
```

```
}
```

- `spawnType` are the `eEncounterSpawnType_t` that you want to be spawned. Multiple types can be listed, and the spawned ai types will be randomly chosen from this list. `ENCOUNTER_SPAWN_ANY` can also be used.
- `desired_count` is how many ai this eventCall should spawn up to. To avoid overlapping spawns, `desired_count` should be $\leq$ the number of `idTarget Spawns` listed under your `idTargetSpawnGroup`.
- `max_spawn_count` is the maximum number of ai that can be spawned by this eventCall in total. This can be set to `-1` for infinite respawns.
- `min_spawn_delay` is the minimum delay between each spawn, in seconds. This value also affects how soon before this eventCall spawns its first ai.
- `min_ai_for_respawn` triggers the ai respawns when they fall below or equal to the value it's set to. `min_ai_for_respawn` should also be $<$ `desired_count`. If it is greater than `desired_count`, the quantity of demons maintained will not properly reflect `desired_count`, possibly exceeding it.
- `spawnGroup` is the `idTargetSpawnGroup` you want to use.
- `group_label` is what label you want to give to the ai spawned by this eventCall. This parameter is optional, and can be left blank.
- `max_spawn_delay` is the maximum delay between each spawn, in seconds.

Example Usage

```
item[0] = {
    eventCall = {
        eventDef = "maintainAICount";
        args = {
            num = 8;
            item[0] = {
                eEncounterSpawnType_t = "ENCOUNTER_SPAWN_ZOMBIE"; // spawnType
            }
            item[1] = {
                int = 10; // desired_count
            }
            item[2] = {
                int = 20; // max_spawn_count
            }
        }
    }
}
```

```

    }
    item[3] = {
        float = 3; // min_spawn_delay
    }
    item[4] = {
        int = 5; // min_ai_for_respawn
    }
    item[5] = {
        entity = "ai_target_spawn_volume_76"; // spawnGroup
    }
    item[6] = {
        string = ""; // group_label
    }
    item[7] = {
        float = 5; // max_spawn_delay
    }
}
}
}

```

In this example, 10 Zombies will be spawned, with a 3-5s delay between each (re)spawn. When only 5 Zombies remain, they will begin respawning until there are 10 Zombies again. Once 20 Zombies have been spawned in total, this eventCall is complete.

Notes

This eventcall appears to have functionality to adjust its kill requirements if not enough enemies are available, to prevent the encounter from locking up, due to otherwise unattainable kill requirements.

If the quantity of currently alive required enemies are greater or equal to the required kills, then the eventcall functions as normal. However, if the quantity of currently alive required enemies are less than the required kills, then the amount of enemies you need to kill is lowered to match the quantity of currently alive required enemies.

This anti-softlock functionality fails if a `maintainAICount` is followed by a `waitKillCount` sometime later in the same encounter manager, while the `maintainAICount` is still active. If there are not enough enemies available in this scenario, then the `waitKillCount` will use the completion of the `maintainAICount` as one of it's requirements. Simply killing the maintained demons until its `max_spawn_count` is depleted will avoid soft locking in this situation. Do note that if `max_spawn_count` is set to `-1`, then the soft lock cannot be avoided.

This functionality has only been tested for in Doom Eternal.

See Also

- [waitMaintainComplete](#)
- [stopMaintainingAICount](#)

Revision #23

Created 5 September 2021 00:11:35 by SamPT

Updated 31 October 2022 18:57:12 by Velsr