

Resetting Encounter Managers

If you want Encounter Managers to reset when a round ends (all players die on either team), you will need to setup a unique exit trigger for every **idEncounterManager** entity. The **round_end** entity must target the exit trigger.

idEncounterTrigger_Exit

```
entity {  
  [entityDef multiplayer_reset_encounter_on_round_end {  
    [inherit = "encounter/trigger/exit";  
    [class = "idEncounterTrigger_Exit";  
    [expandInheritance = false;  
    [poolCount = 0;  
    [poolGranularity = 2;  
    [networkReplicated = false;  
    [disableAIPooling = false;  
    [edit = {  
      [spawnPosition = { // The spawn position does not matter  
        [x = 0;  
        [y = 0;  
        [z = 0;  
      }  
      [triggerOnce = false; // Must be false  
      [resetScriptOnExit = true; // Must be true  
    }  
  }  
}
```

round_end

This entity will always activate so long as the entityDef name is unchanged.

```
entity {  
  [entityDef round_end { // Do not change the entityDef name
```

```

inherit = "target/relay";
class = "idTarget_Count";
expandInheritance = false;
poolCount = 0;
poolGranularity = 2;
networkReplicated = false; // Must be false
disableAIPooling = false;
edit = {
  flags = {
    noFlood = true;
  }
  count = 1;
  delay = 0;
  call = "";
  repeat = true; // Must be true
  checkLiveActivator = false;
  adjustFor16msFrames = false;
  selfActivateDelay = -1;
  waitInterval = 0;
  delayMin = -1;
  delayMax = -1;
  spawnPosition = { // The spawn position does not matter
    x = 1;
    y = 1;
    z = 1;
  }
  targets = {
    num = 2;
    item[0] = "multiplayer_reset_encounter_on_round_end"; // Exit trigger
    item[1] = "invasion_ambient_music_1_new"; // Recommended to target a new ambient music
    timeline
  }
}
}

```

idEncounterManager

```

entity {
    entityDef coop_encounter_manager_test_1 {
        inherit = "encounter/manager";
        class = "idEncounterManager";
        expandInheritance = false;
        poolCount = 0;
        poolGranularity = 2;
        networkReplicated = false;
        disableAIPooling = false;
        edit = {
            enableAIHighlightOnFinish = false;
            disabledAITypeForHighlight = "AI_MONSTER_SPECTRE AI_MONSTER_BUFF_POD AI_MONSTER_TENTACLE";
            playerMetricDef = "encounter/player_metrics";
            chargeCombatGrouping = "encounter/combat_role/charge_command";
            aiTypeDefAssignments = "actorpopulation/default/default_no_bosses";
            spawnPosition = { // The spawn position does not matter
                x = 0;
                y = 0;
                z = 0;
            }
            combatRatingScale = "COMBAT_RATING_SCALE_IGNORE";
            encounterComponent = {
                entityEvents = {
                    hum = 1;
                    item[ 0 ] = {
                        entity = "coop_encounter_manager_test_1"; // Must match the entityDef name
                        events = {
                            hum = 1;
                            item[ 0 ] = {
                                eventCall = {
                                    eventDef = "designerComment";
                                    args = {
                                        hum = 2;
                                        item[ 0 ] = {
                                            string = "This is a test encounter";
                                        }
                                    }
                                }
                                item[ 1 ] = {
                                    pool = true;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```
    }
  }
}

exitTriggers = {
  hum = 1; // Must be present in addition to the encounter's normal exit trigger
  item[0] = "multiplayer_reset_encounter_on_round_end";
}
}
```

Revision #4

Created 12 February 2024 20:44:55 by Velser

Updated 18 July 2024 13:57:02 by Velser