

waitMultipleConditions

An eventCall for using multiple wait eventCalls in an idEncounterManager.

This eventCall is spelled incorrectly in the engine ("Multiple"), and must be used this way.

Usage

```
item[0] = {
  eventCall = {
    eventDef = "waitMultipleConditions";
    args = {
      num = 2;
      item[0] = {
        int = 2; // condition_count
      }
      item[1] = {
        encounterLogicOperator_t = ""; // logic_operator
      }
    }
  }
}
```

- `condition_count` is the amount of wait conditions
- `logic_operator` is the logic operator used for the wait conditions. The only operators are `ENCOUNTER_LOGICAL_OP_OR` (any wait condition is fulfilled) and `ENCOUNTER_LOGICAL_OP_AND` (all wait conditions are fulfilled).

Example Usage #1

```
item[0] = {
  eventCall = {
    eventDef = "waitMultipleConditions";
    args = {
      num = 2;
```

```

    item[0] = {
        int = 2; // condition_count
    }
    item[1] = {
        encounterLogicOperator_t = "ENCOUNTER_LOGICAL_OP_OR"; // logic_operator
    }
}

item[1] = {
    eventCall = {
        eventDef = "wait";
        args = {
            num = 1;
            item[0] = {
                float = 10;
            }
        }
    }
    item[2] = {
        eventCall = {
            eventDef = "waitKillCount";
            args = {
                num = 3;
                item[0] = {
                    eEncounterEventFlags_t = "ENCOUNTER_SPAWN_ANY";
                }
                item[1] = {
                    int = 2;
                }
                item[2] = {
                    string = "";
                }
            }
        }
    }
}

```

In this example we have two conditions, set to **OR**. This way, when either 10s pass or any 2 ai types are killed, the encounter proceeds.

Example Usage #2

```
item[0] = {
    eventCall = {
        eventDef = "waitMultipleConditions";
        args = {
            num = 2;
            item[0] = {
                int = 3; // condition_count
            }
            item[1] = {
                encounterLogicOperator_t = "ENCOUNTER_LOGICAL_OP_AND"; // logic_operator
            }
        }
    }
}

item[1] = {
    eventCall = {
        eventDef = "waitAIRemaining";
        args = {
            num = 3;
            item[0] = {
                eEncounterSpawnType_t = "ENCOUNTER_SPAWN_BARON";
            }
            item[1] = {
                int = 0; // desired_remaing_ai_count
            }
            item[2] = {
                string = ""; // group_label
            }
        }
    }
}

item[2] = {
    eventCall = {
        eventDef = "waitAIRemaining";
        args = {
            num = 3;
            item[0] = {
```

```

    eEncounterSpawnType_t = "ENCOUNTER_SPAWN_MANCUBUS";
}

item[1] = {
    int = 0; // desired_remaing_ai_count
}

item[2] = {
    string = ""; // group_label
}

}

item[3] = {
    eventCall = {
        eventDef = "wait";
        args = {
            num = 1;
            item[0] = {
                float = 5;
            }
        }
    }
}

```

In this example, we have three conditions, set to **AND**. This way, there needs to be no Barons and Mancubi, and 5s must have passed as well, in order for the encounter to proceed.

Revision #3

Created 8 September 2021 02:31:04 by elizabethany

Updated 13 August 2024 10:17:23 by SamPT