idSummoningTemplate

> An entity to control summoning behaviour of a Summoner.

Usage

```
entity {
	entityDef example_summoning_template {
	class = "idSummoningTemplate";
	expandInheritance = false;
	poolCount = 0;
	poolGranularity = 2;
	edit = {
		fixedSummoningWaves = { // List of tiered summoning waves, executed sequentially.
			num = 1;
			item[0] = {
				tier1_SpawnList = { // An Imp or Possessed Soldier can be summoned for this tier.
					num = 2;
					item[0] = "ENCOUNTER_SPAWN_IMP";
					item[1] = "ENCOUNTER_SPAWN_HELL_SOLDIER";
				}
				tier2_SpawnList = {
					num = 2;
					item[0] = "ENCOUNTER_SPAWN_IMP";
					item[1] = "ENCOUNTER_SPAWN_ZOMBIE";
				}
				tier3_SpawnList = {
					num = 2;
					item[0] = "ENCOUNTER_SPAWN_SHOTGUN_SOLDIER";
					item[1] = "ENCOUNTER_SPAWN_BEAM_SOLDIER";
				}
				completionTest = "SUMMON_CONDITIONS_ALL_DEAD"; // SUMMON_CONDITIONS_ALL_DEAD,
SUMMON_CONDITIONS_FIXED_TIMER, or SUMMON_CONDITIONS_WEIGHTED_TIMER
			}
		}
		randomSummoningWaves = { // List of tiered summoning waves, executed in order but looped
```

```
until the summoning encounter is complete.
□□□num = 3;
□□□item[0] = {
□□□□tier1_SpawnList = {
□□□□□num = 2;
□□□□□item[0] = "ENCOUNTER_SPAWN_IMP";
□□□□□item[1] = "ENCOUNTER_SPAWN_ZOMBIE";
□□□□}
□□□□tier2_SpawnList = {
□□□□□num = 1;
□□□□□item[0] = "ENCOUNTER_SPAWN_REVENANT";
□□□□}
□□□□completionTest = "SUMMON_CONDITIONS_ALL_DEAD";
□□□}
□□□item[1] = {
□□□□tier1_SpawnList = {
□□□□□num = 2;
□□□□□item[0] = "ENCOUNTER_SPAWN_ZOMBIE";
□□□□□item[1] = "ENCOUNTER_SPAWN_ZOMBIE";
□□□□}
□□□□tier2_SpawnList = {
□□□□□num = 1;
□□□□□item[0] = "ENCOUNTER_SPAWN_PINKY";
□□□□}
□□□□completionTest = "SUMMON_CONDITIONS_ALL_DEAD";
□□□}
□□□item[2] = {
□□□□tier1_SpawnList = {
□□□□□num = 3;
□□□□□item[0] = "ENCOUNTER_SPAWN_IMP";
□□□□□item[1] = "ENCOUNTER_SPAWN_ZOMBIE";
□□□□□item[2] = "ENCOUNTER_SPAWN_IMP";
□□□□}
□□□□tier2_SpawnList = {
□□□□□num = 2;
□□□□□item[0] = "ENCOUNTER_SPAWN_PINKY";
□□□□□item[1] = "ENCOUNTER_SPAWN_REVENANT";
□□□□}
□□□□completionTest = "SUMMON_CONDITIONS_ALL_DEAD";
□□□}
```

```
  }
  fixedSummonGroups = { // List of groups that is sequentially run
    item[0] = {
      aiSpawnTypeList = {
        item[0] = "ENCOUNTER_SPAWN_BEAM_SOLDIER";
        item[1] = "ENCOUNTER_SPAWN_BEAM_SOLDIER";
        item[2] = "ENCOUNTER_SPAWN_IMP";
      }
    }
    item[1] = {
      aiSpawnTypeList = {
        item[0] = "ENCOUNTER_SPAWN_BEAM_SOLDIER";
        item[1] = "ENCOUNTER_SPAWN_IMP";
        item[2] = "ENCOUNTER_SPAWN_HELL_KNIGHT";
      }
    }
  }
  randomSummonPool = { // List of group to randomly run after all fixed encounters are complete
    item[0] = {
      aiSpawnTypeList = {
        item[0] = "ENCOUNTER_SPAWN_IMP";
        item[1] = "ENCOUNTER_SPAWN_BEAM_SOLDIER";
        item[2] = "ENCOUNTER_SPAWN_BEAM_SOLDIER";
      }
    }
    item[1] = {
      aiSpawnTypeList = {
        item[0] = "ENCOUNTER_SPAWN_IMP";
        item[1] = "ENCOUNTER_SPAWN_BEAM_SOLDIER";
        item[2] = "ENCOUNTER_SPAWN_HELL_KNIGHT";
      }
    }
  }
  spawnPosition = {
    x = 8170;
    y = 6877;
    z = 8000;
  }
 }
}
```

```
}
```

- `fixedSummoningWaves` : The summoner summons up to three demons, each one drawn from the given tier list. Once `completionTest` is fulfilled, the rift will summon the next wave. These waves are executed in the listed order, ie. item[0] -> item[1] -> item[2] etc. Ends once all waves are complete.
- `randomSummoningWaves` : Similar to `fixedSummoningWaves`, but will loop upon completion, instead of stopping.
- `fixedSummonGroups` : List of groups that is sequentially run (?); the function of this summoning behaviour is not fully understood at this time.
- `randomSummonPool` : One the preceding fixedSummoningWaves or fixedSummonGroups is complete, the summoning rift will randomly cycle through the defined aiSpawnTypeLists and summon.

- `completionTest` : All `idSummoningTemplates` in the vanilla game use `SUMMON_CONDITIONS_ALL_DEAD`. While `SUMMON_CONDITIONS_FIXED_TIMER`, or `SUMMON_CONDITIONS_WEIGHTED_TIMER` are also options, it is currently not understood how to use those conditions.

Note that once the summoning rift is created, these behaviours will execute on their own, as long as the Summoner remains alive. If killed, the rift will close, and anything defined by the idSummoningTemplate will stop immediately.

Created 2022-08-24 18:03:28 PDT by Velser
Updated 2023-08-22 19:56:42 PDT by Velser