idTarget_Spawn

> An entity to spawn other entities. Primarily used for idAI2 entities.

Usage

```
entity {
	entityDef example_target_spawn_2016 {
	inherit = "target/spawn";
	class = "idTarget_Spawn";
	expandInheritance = false;
	poolCount = 0;
	poolGranularity = 2;
	edit = {
		"idEntity::entityFlags_t" flags = {
			"bool" noFlood = true;
		}
		spawnEditable = {
			overwriteTraversalFlags = true; // if set to true, use traversalClassFlags to overwrite the ai's flags
			traversalClassFlags = "CLASS_A";
			spawnAt = ""; // idEntity; the entity (demon) spawned by this spawn target will spawn at the listed entity.
			copyTargets = true; // // if true the targeted entity's target list will be copied to the spawned entity
			additionalTargets = {
				num = 0;
				item[0] = ""; // idEntity
			}
			spawnAnim = ""; // aiStateOverride must be set to AIOVERRIDE_PLAY_ENTRANCE_ANIMATION in orde to use spawnAnim
			aiStateOverride = "AIOVERRIDE_TELEPORT"; // aiStateOverride_t
		}
		spawnConditions = { // Inherited from idTarget_Spawn_Parent's idSpawnConditions
			maxCount = 0; // maximum number of entities (demons) that this spawn target will spawn
			reuseDelaySec = 0; // delay in seconds, between uses
```

```
□□□doBoundsTest = false;
□□□boundsTestType = "BOUNDSTEST_NONE"; // boundsTest_t
□□□fovCheck = 0; // Whether the spawn target is (not) in the player's FOV, in degrees. Use positive values for within, and negative values for outside of
□□□minDistance = 0; // The player cannot be closer than the specified distance to the spawn target.
□□□maxDistance = 0; // The player cannot be further than the specified distance to the spawn target.
□□□neighborSpawnerDistance = -1;
□□□LOS_Test = "LOS_NONE"; // targetSpawn_LOSTest; to be combined with an FOV test. LOS_NONE no test, LOS_TRUE must be in LOS, LOS_FALSE must not be in LOS
□□□playerToTest = "PLAYER_SP"; // which player to test for this spawn target; leave as is for the primary player
□□□conditionProxy = ""; // idInfo_SpawnConditionProxy; entity at location to do LOS, FOV, and distance checks
□□}
□□spawnPosition = {
□□□x = 0;
□□□y = 0;
□□□z = 0;
□□}
□□spawnOrientation = { // 0 0 0
□□□mat = {
□□□□mat[0] = {
□□□□□x = 1;
□□□□□y = 0;
□□□□□z = -0;
□□□□}
□□□□mat[1] = {
□□□□□x = 0;
□□□□□y = 1;
□□□□□z = 0;
□□□□}
□□□□mat[2] = {
□□□□□x = 0;
□□□□□y = 0;
□□□□□z = 1;
□□□□}
□□□}
□□}
```

```
□□entityDefs = { // entity list to be spawned by the spawn target
□□□num = 1;
□□□item[0] = {
□□□□name = "example_ai_imp_1";
□□□}
□□}
□□targets = { // target list to be spawned by the spawn target; this can be excluded/left empty
if targetSpawnParent is set, it will draw targets from there instead
□□□num = 0;
□□□item[0] = "";
□□}
□□targetSpawnParent = "example_spawn_parent_1"; // idTarget_Spawn_Parent
□}
}
}
```

It should be noted that most of these parameters can be excluded unless needed. An idTarget_Spawn can be simplified down to this, as an example, if all the player wishes to do is be able to spawn in demons with the teleport anim/fx and no other frills.

```
entity {
□entityDef game_target_spawn_mod_1 {
□inherit = "target/spawn";
□class = "idTarget_Spawn";
□expandInheritance = false;
□poolCount = 0;
□poolGranularity = 2;
□edit = {
□□"idEntity::entityFlags_t" flags = {
□□□"bool" noFlood = true;
□□}
□□spawnEditable = {
□□□traversalClassFlags = "CLASS_A";
□□□aiStateOverride = "AIOVERRIDE_TELEPORT";
□□}
□□spawnPosition = {
□□□x = 8717;
□□□y = 8363;
□□□z = 6633;
□□}
□□entityDefs = {
```

```
□□□num = 1;
□□□item[0] = {
□□□□name = "mod_ai_imp_1";
□□□}
□□}
□□targetSpawnParent = "mod_master_spawn_parent";
□}
}
}
```

Created 2022-08-24 17:50:40 PDT by Velser
Updated 2023-08-22 19:56:42 PDT by Velser