idTarget_LayerStateChange

> An entity that can save the player's progress as a checkpoint and-as its name implies-manipulate layers.

Checkpoint

```
entity {
	entityDef checkpoint_target_change_layer_1 {
	class = "idTarget_LayerStateChange";
	expandInheritance = false;
	poolCount = 0;
	poolGranularity = 2;
	networkReplicated = false;
	disableAIPooling = false;
	edit = {
		spawnPosition = {
			x = 1;
			y = 1;
			z = 1;
		}
		activate_NextMapLoad = {
			num = 1;
			item[0] = "game/sp/e1m1_intro/cp_01";
		}
		checkpointName = "cp_01";
		playerSpawnSpot = "checkpoint_player_start_cp_01";
		saveCheckpoint = true;
	}
	}
	}
```

Checkpoint parameters

`checkpointName` : Define a checkpoint name.

>

> If you attempt to load a save when no layer state change entity with `checkpointName` set to the value the game is expecting, it will crash to the desktop. If meathook is installed, a message box will appear telling you the missing checkpoint.

`playerSpawnSpot` : the `idPlayerStart` entity the player will spawn at when loading the checkpoint.

`saveCheckpoint` : whether or not to save a checkpoint. Defaults to true, so if you only want to manipulate layers when activated, then explicitly set this to false.

Layer Manipulation

> Entities will only react to state changes applied to its first layer. You may notice entities in vanilla levels with multiple layers; this is pointless.

> All layers in the game can be used in all levels.

`activate_NextMapLoad` : Wait until the player loads the checkpoint to activate this layer. This is typically used to activate checkpoint layers that are placed on idTarget_FirstThinkActivate entities that set the map up as if the player had progressed through it when a checkpoint is loaded.

`activate_Immediately` : Spawn all entities attached to the layers specified right away.

`remove_Immediately` : Completely remove any entities attached to these layers from the map.

> These options are used to hide and show parts of the map as the player progresses through it for optimization. Improperly handling these layers can cause visual bugs, or degraded graphical performance. Usage is recommended only for advanced modders that are certain they have a specific reason to do so. (for example, you intend the player to progress through the map backwards what is traditionally expected)

`hideRenderModels_Immediately` : Make all entities and map geometry with these layers visible.

`showRenderModels_Immediately` : Make all entities and map geometry with these layers invisible.

`dormant_Immediately` : Suspends entities with these layers, meaning they don't have an effect on the world, player, and so on.

`awake_Immediately` : Resumes entities with these layers.

"NextMapLoad" variants of all layer state changes exist, however they are unused, with the exception of the aforementioned activate_NextMapLoad.

idTrigger_VisibilityController

For optimization, visibility controller entities activate layer changes as the player traverses the map. You probably should not touch these, this is listed here for completeness in the unlikely event you need to edit one.

```
entity {
	entityDef visibility_trigger_visibility_controller_trigger_34 {
	inherit = "trigger/visibility_controller_trigger";
	class = "idTrigger_VisibilityController";
	expandInheritance = false;
	poolCount = 0;
	poolGranularity = 2;
	networkReplicated = false;
	disableAIPooling = false;
	edit = {
		demonPlayerCanActivate = true;
		playerCanActivate = true;
		onlyLocalPlayerCanActivate = true;
		spawnPosition = {
			x = -15;
			y = 295;
			z = 15;
		}
		renderModelInfo = {
			model = NULL;
		}
		clipModelInfo = {
			clipModelName =
"maps/game/sp/e3m1_slayer/e3m1_slayer/visibility_trigger_visibility_controller_trigger_34";
		}
		onActivateLayerChangeTarget = "visibility_target_change_layer_1_zone15_on";
	}
}
}
```

Created 2023-03-19 21:58:58 PDT by Alby
Updated 2023-06-24 11:53:21 PDT by Alby