Areas

Areas are oblong-shaped collision zones which can trigger callbacks when the player or other objects enter them; they are also often used as position markers for things spawned in code such as sounds or particle effects.

HPL3 ships with several different area types - here's a quick summary of them and of some of their properties.

Trigger

The most commonly used type of area, used for all sorts of purposes.

Important Properties

BlocksLineOfSight

If true, this area acts as a blocker for AI seeing the player.

CC_Entities, CC_Funcs

Collision callbacks - for example, if you set CC_Entities to `player` and CC_Funcs to `TriggerHitArea` then TriggerHitArea will be called in your map script whenever the player enters or leaves the area.

PlayerLookAtCallback

Called when the player has the area on-screen. Various other properties in the same section modify exactly when that happens.

PlayerInteractCallback

Makes the area interactable, and triggers this callback when the player interacts with it. You can also set `CustomInteractIcon` etc.

AmbientLight

The player, by default, casts a small point light around him so that you can see things even when the room is very dark. This area type changes that ambient light to fit the current area.

Important Properties

Type

Indoors / Outdoors / None

CameraAnimation

Used as a camera viewpoint in a camera animation. See Camera Animation.

Climb

This area becomes interactable; when you interact, you will climb up to a specified foot position. It's used for things like climbing into vent shafts or climbing up on to a nearby table or something; for higher climbs, you probably want a camera animation or a ladder instead.

Important Properties

FeetPosition

Name of an entity (usually a trigger area) specifying where the feet should land after the climb.

Crouch

Specify if the player should be automatically crouched after the climb (useful for vent shafts).

Crawl

The player will be crawling on their belly while in this area, and can't run or stand up. Useful for vent shafts.

Datamine

The player can interact with this area as if it were a datamine prop - see Datamine

DatamineAudioSource, DatamineAnimNode

Currently unused.

Description

Used with the Description module. Not used in SOMA.

Distortion

This area will apply a Distortion Effect to the player while they are in it.

DoorwayTrigger

This area is really useful for telling you when the player has walked into a particular part of the map. You set up a doorway with the Z-axis pointing in to the 'inside' and away from the 'outside'. Now when the player walks in through the door, the DoorwayCallback is called with an `alState` parameter of `1`. When the player walks out through the door, the callback is called with an `alState` of -1.

You can also link sets of doorways, which is a great way to figure out if the player is inside an irregular space. If you specify the `DoorwayGroup` property, then you can use the function `Doorway_PlayerIsInGroup` to check if the player has crossed into the space delimited by these doorways. (Essentially, if you cross any of the doorways going 'inwards' so `alState==1`, then you are in the group; crossing any going 'outwards' so `alState==`-1 means you are outside the group).

Important Properties

DoorwayEntity

This acts just like CC_Entities on an area - it just specifies which entity to check for collisions with. This is usually `player`.

DoorwayCallback

Called when the `DoorwayEntity` goes in or out of the doorway.

Hide

This marks of a specific area as somewhere that the player can hide i.e. they will no longer be visible to AI. This is not used in SOMA.

InteractAux

This is used to expand the interaction area of another entity such as a prop; so that you can, for example, give a small button a bigger interact area. For all purposes, interacting with the area is treated as if you're interacting with the prop itself. If you set interaction disabled on the prop, the area will deactivate too.

Important Properties

InteractParent

The entity to interact with.

Ladder

This is used to allow the player to climb ladders. Position the ladder area up the ladder prop. Note that in SOMA rungs have to be 0.5 units apart, and the ladder area itself has to be a multiple of 0.5 units tall and aligned to the rungs for the animation to work properly.

Important Properties

Exit_Top, Exit_Top_Crouching

Use these to specify an area to move to when you step off the top of the ladder (usually a Trigger area).

Exit_Bottom, Exit_Bottom_Crouching

Use these to specify an area to move to when you step off the bottom of the ladder (usually a Trigger area).

Liquid

Creates an area of liquid - useful for pools of water or for use in SOMA's airlocks. You can control waves, surface colors, fog under water etc. - properties should be reasonably self-explanatory.

MapTransfer

Map Transfer areas are used in map transitions; anything within a Map Transfer area is preserved as the next map is loaded.

PathNode

Pathnodes are points rather than areas; they are used in AI for the pathfinding system.

PlayerStart

Player Start areas are where the player will appear as the result of a map load (by default, the player appears at `PlayerStartArea_1`, although that can be changed) or as the result of a `Player_Teleport()` call.

Important Properties

Crouching

Specifies whether the player should be crouching when they appear at that spot.

PosNode

Not used in SOMA.

Rope

Not used in SOMA.

Sit

Sit areas specify an area that the player can interact with to sit down.

Soundscape

Soundscape areas are used to set up the ambient sound for particular areas. You'd generally have one soundscape area for every room. See Soundscape Area for more details.

Sticky

Sticky areas are used to attach other entities to (normally `Prop_Grab`) - when the entities are pulled away, there'll be resistance, and if you don't pull hard enough they'll snap back. This is used to implement things like pulling power cables from sockets or attaching cables to sockets.

Important Properties

AttachableBodyName

Name or names of bodies that can be attached to this area. Accepts wildcards.

CanDetach

If true then the connected body can be detached.

RequiresInteraction

If true then bodies will only detach or attach if the player is actually interacting with them (as opposed to e.g. walking into an object to knock it free).

MoveBody, RotateBody

Whether to orient the body to the attach position.

AttachFunction

Called when an entity is attached to this area.

DetachFunction

Called when an entity is detached from this area.

SpringyActive, SpringyForceMul, SpringyDetachTime

Use these to give some resistance to the detachment or attachment - so that the attach area 'sucks in' the body.

Tool

If the player walks into a tool area and has the specified tool in their inventory, then it will automatically be equipped (i.e. brought out and shown on screen in their hands). This is what is used for showing the Omnitool next to an airlock control panel.

Important Properties

ToolsToEquip

A list of tools to bring out when in this area.

ConnectedToolAreas

Other tool areas that are part of the same group; this is to make it easy to set up irregularly shaped tool areas by combining multiple rectangular areas.

LookEntities

Specify a look entity, and the tool will only be brought out if the player is facing that entity. In most cases, it would be the object that they can interact with e.g. the airlock panel. This is used so that the tool isn't brought out if the player backs in to a tool area.

AgentRepel

This area type repels AI creatures who enter it.

Important Properties

ExitArea

Specifies a particular point (usually a TriggerArea) to head towards if the agent enters this area.

VisibilityArea, VisibilityPortal

These two area types are used to toggle the visibility of large amounts of geometry, which is useful in optimising maps. See Visibility Areas

Zoom

See also Readables.

A zoom area shows a magnifying glass icon when the player interacts with it; when they interact, the player's viewpoint is moved in towards the area, so that they're focused on the contents. You can also add optional text or GUI to pop up over the area.

This is used in SOMA for things like whiteboards, maps and pictures on the wall.

Important Properties

TextCategory, TextEntry

Specify the text to appear when you click interact for a second time. This is used for simple descriptive text.

OnGuiFunction

Instead of text appearing, if you specify an OnGuiFunction that will be called instead. This is useful for drawing more complex text layouts such as the key locations on a map.