

Interactables

Interactables are objects in the game world that can be interacted with, and trigger a specific effect when this is done. Effects can be spells cast on the interactable target, or other specified targets, or changes made to the game world. Technically, virtually anything can be done.

Constructing an Interactable

Interactables require a scene with a `CharacterBody3D` root node, and an attached script that inherits from `BaseInteractable`. Mandatory child nodes are a `Label3D` node named "interact_prompt", an `Area3D` node named "range" that scans collision layer 2, and which has a `CollisionShape3D` child node named "range_shape" that uses a `SphereShape3D` as shape, and finally a `MultiplayerSynchronizer` node named "mpsynchronizer".

The attached script needs to set the multiplayer authority to 1 (the server) in the `_enter_tree` function, and needs to call `create_prompt_text` and `initialize_base_interactable` from the `BaseInteractable` class. `create_prompt_text` needs to be called locally for all players, i.e. for all peers that are NOT multiplayer authority. The initialization of the base interactable needs to be called on the server, i.e. for the multiplayer authority.

The script further requires a trigger function, which uses the root node of the interacting player scene as an argument. The `BaseInteractable` class contains a trigger function, which will warn the server via a printed message if it is not overridden in the script of a specific interactable.

If a spell is to be triggered by interacting, the spell must be triggered directly, and not via the `spell_container` entrypoint. This is to allow for the easy reuse of the `spell_container` scene, but to still be able to pass the interacting player as an argument as the target of the triggered spell.

BaseInteractable

The `BaseInteractable` class is the class from which interactables inherit. It contains a `stats_current` variable, which is required to calculate spell effects, contains the maximum range at which interactions can occur, and contains the array of spells the interactable has available.

The `BaseInteractable` class further contains a number of functions:

initialize_base_interactable takes the unit ID of the interactable to read the stats from the `data/db_stats_interactable.json` file, sets up the spell container with all required spells, calls the `connect_signals` function described below, and sets the maximum interaction range.

connect_signals connects the `body_entered` and `body_exited` signals of the "range" child node to the `add_interactable` and `remove_interactable` functions described below. The shape and size of the

body of the "range" node is determined by its "range_shape" child node, and should be set to a SphereShape3D.

add_interactable adds an interactable to the array of interactables of a player when they enter the body of the interactable.

remove_interactable removes an interactable from the array of interactables of a player when they exit the body of the interactable.

Note: A player can only interact with the nearest interactable within their interactables array. The nearest interactable is searched for every frame by the `get_nearest_interactable` function in the player script. This function also uses `rpc` to show or hide interact prompts as necessary for the corresponding player. The interact prompt is only shown above the current nearest interactable, to clearly indicate which interactable is triggered if the interaction hotkey is used.

create_prompt_text creates the text that is shown in the `interact_prompt` label. The prompt is "Interact [%s]", with %s being the hotkey for interactions, where the " (Physical)" suffix is trimmed, if it is part of the hotkey.

trigger is a function that provides a warning in the server standard output if it is not overridden by a trigger function in the interactable script that inherits from `BaseInteractable`.

Adding a New Interactable

When adding a new interactable, it should be placed in the "interactable" subdirectory of the "scenes" directory, and the script for it should be placed in the "interactable" subdirectory of the "scripts" directory. A basic template for interactables can be found in the testing subdirectory of these directories, as `interact_absorb`, `interact_damage`, and `interact_heal`.

Every interactable also needs an entry in the `data/db_stats_interactables.json` file. A new ID must be assigned, and a dictionary specified that contains all required information. This ID must then be passed to the `initialize_base_interactable` function as an argument during the `_ready` function of the new interactable. The required keys in the dictionary are

unit_name: Specifies the name of the interactable that appears in the combat log.

spell_list: The list of spells that the interactable can use. Can be left empty, if no spell is required, but the key itself is mandatory.

interact_range: The maximum range at which the interactable is present in a players interactables array.

If the interaction triggers a spell, the dictionary must further contain all stats required for the spell to function, such as a primary value, damage modifiers, heal modifiers, etc.

Note: If a triggered spell should target the interacting player, the interactor can be passed to the spell's trigger function to achieve this. This requires that the spell in question takes a target as an argument in its own trigger function.

Revision #4

Created 2024-03-19 20:52:22 UTC by cheese

Updated 2024-03-20 13:18:45 UTC by cheese