

skills_component

Skills component adds the ability to cast skills to it's parent entity.

Interfaces:

cast_skill(id, target)

casts the specified skill, optionally on a given target. target can be a reference or location (? figure out if thats good, maybe look up in the skill db whether the skill is entity-target or ground target?)

get_stats()

if the skill can be affected by stats, the skill component checks if the parent has a stats component, if so it reads the relevant stats and applies them to the skill.

spawn_skill_scene(data, target)

once ready, the skills component instantiates the necessary scenes for the skill, and places them at the given target.

Necessary initialisation:

skills_db:

the component needs a skill db to know what skills there are and which scenes to spawn for each.

skills_scenes:

we need to have pre-made scenes available for each skill. These can either be just a script, for example some buff that just modifies stats, or whole 3d scenes with animations, particles, collision boxes etc.

connecting signals:

to keep flexibility, we should implement the interfaces with signals - so skills_component needs to listen to it's parent's cast_skill signal. This also allows things like a mana component to listen to them, and we don't need to call 5 different components each skill cast. (cooldown should probably be handled by UI?)

player_skills_unlocked:

~~we need to get a list of skills that are available to the player, otherwise we'd be able to cast whatever.~~

Specifically excluded:

skills_UI:

we exclude the UI parts - mobs can cast skills but don't need this. The UI can also read data from the skills_db.

The UI will additionally only display skills the player has unlocked, so we do NOT need to check this within this component.

Revision #1

Created 2023-07-19 08:39:05 UTC by Admin

Updated 2023-07-19 08:51:56 UTC by Admin