

# Component system

We use components to avoid having to copy paste code, while also not running into inheritance issues that class systems have.

(classes are still used, but mainly to have pre-configured classes already containing components that are correctly linked to each other)

- [skills\\_component](#)

# 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.