Puzzle Elements

- **Portal blocks:** I had already created them, but they don't work in multiplayer.
- **Illusion blocks:** Looks like a normal floor block, but it isn't. You pass through it, when you step on it. Already created in GQ.
- **Locked doors**: Doors having wheel like combinations that need to be aligned with a password. Other types like a UI Screen password one too. Some doors could be opened with keys as well.
- **Swinging Blade**
- **Swinging Blade horizontal**- Jump on rod to get to the other side. Lava underneath.
- **Spike Traps**: Some blocks with spikes on the top on them, preventing a player from walking on top of them without taking damage. Some can be pressure activated if necessary.

Triggers: Elements which trigger something to happen

- **Pressure plates that unlock a door:** Standing on it unlocks a door for a few seconds.
- **Locks with item key:** A block that requires a special key like item that when placed inside a cavity, unlocks it. Imagine a block with a spherical cavity on top, that takes in only an orb or an egg. When the orb/egg is placed, it unlocks the door or triggers something else.
- **Tripwire:** A trip wire that makes something happen.
- **Buttons:** Simple buttons on the wall to trigger something.

Actions: Things which can happen in response to some triggers.

- **An arrow/fireball launcher:** Launches from the lever near the door or other traps. (No triggers, periodic fire)
- **Blocks with spikes that fall down**
- **A rolling boulder**
- **A trap block under you that gives way, so you fall**

Pros and cons for different trap placement approaches:

Trap Placeholder with Trap Configuration Tool:

Pros:

- **Easy trap placement and selection. Trap placement is as simple as placing the placeholder block and selecting which trap from dropdown.**
- **Trap removal is as easy as mining the trap placeholder block.**
- **Creating a structure spawner, removes the trap placeholder blocks from appearing. (Inconsistent, but saves trap from destruction)**
- **Easy to edit, using trapConfigurationTool by just aiming.**

Cons:

- **Flo: Multiple steps are necessary to place a trap. You first have to place the placeholder and then activate it and then select the trap then close that dialog select the tool for configuring it and use it to select the details.**
- **Flo: Traps can't be replaced as there is no item that remembers the trap settings when the trap is mined**
- **Requires trapConfigurationTool for editing trap properties and removal (when no placeholder block exists)**
- **There is no simple place and use trap item with which the player can just play around. The player needs to read up somewhere how to change trap settings if he ever finds out that they can be configured**
- **Aiming might not work well in multiplayer.**
- **Difficult removal of traps if structure is mistakenly spawned a little off.**

<u>Root block for traps:</u>

Pros:

- **Allows easy placement and removal. Placing a block would create a trap. Mining would destroy it.**
-

Cons:

- **Difficult editing of properties. Editing only possible upon interaction with the root block which might be unreachable without cheats.**

Discussion:

- Nihal: I feel that the trap configuration tool can still be used with the root block approach. We can *also* allow editing using the configuration tool, apart from the primary editing via root block interaction.
- Flo: the question is why do we want to allow editing of blade part only via config tool when it would be easier to just allow activation
- So if you went through the video in which I use the trap config tool to set properties of multiple swinging blades in action, it really allows me to set properties for the traps as they would appear in game. Meaning, while walking past them as I would in-game, I could change their properties. Maybe create a valid configuration to allow a safe path across.
- Flo: well i mean activation with E of the blade region currently targeted by tool.
- Nihal: Okay, so you want to make the whole blade entity interactive?
- Flo: the static root entity that you seem to have targeted with tool
- Nihal: I had to create a box shape for the mesh entity just for the purpose of it being registered in the cameraTargetSystem.
- Flo: it looked static but mesh is animated???
- Nihal: Yes, the wireframe box appeared static, even though its rotation was changing. It is a rendering bug.

- Flo: well there could be just a static box like that in the root entity so that it works in network setting
- Nihal: Makes sense, we have to make it non-collidable though.
- Flo: then activation with E would also be possible
- Nihal: Fair enough. So which approach are you in favour of? The root block/entity one? We should pen down the details. Although I already started with some of the implementation.
- Flo: Well I think config tool or activation by E is a separate question.
- Nihal: Okay, so which approach?
- Flo: well if we have direct targeting of blade and if that works then it would be easiest to use. Including allowing of mining of the trap
- Nihal: By mining you mean minging the root block, right? And not damaging the trap itself.
- Flo: no I mean the static box that we give the root entity
- Nihal: Yeah, okay.
- Flo: so the complete blade would behave like a block. It is available as placeable item. It drops an item when mined
- Nihal: How do you limit damage, once again? What event handler?
- Flo: have a look at the illusion floor template of gooey's quest it has a special component that makes a region of block not mineable. So you could search for usages of that component to find the event handler
- Nihal: Okay, so let's pen down the details for the root entity setup. You wanna do this in discussion here or separately?
- Flo: let's create a list separately and maybe use sublists for discussion
- Nihal:
- Flo:

Root Entity for Traps:

- **Root entity of the trap is a block.**
    - Flo:actually if the root entity just behaves like a block then we dont need a block
    - Nihal: We can still send the damage to the block when the static box is damaged, if that's what you're thinking.
    - Flo: no. See above
    - Nihal: Not really clear.
    - Flo: the root component gets a health component and thus since it has a static box shape it can be mined. The entity has a logic that makes it drop a item that places it
    - Nihal: How do you place it?
    - Flo: E.g. via a structure template item that has a single use and spawns trap with exact settings the previously mined blade had

- Nihal: I am a little skeptical about this approach for 2 reasons. 1, this destroys the point of easy-to-use just place-a-block to generate trap.
- Flo: how so?
- Nihal: Would the trap appear in white wire frames before placing? I don't think this would be as intuitive to use as a block which creates the trap.
- Flo: Well if we want to we can even show a preview of the blades position. Basically structure template items create bordered region entities whenever the targeted block changes. So we could maybe introduce a MeshPreviewComponent that makes items that have it show that mesh
- Nihal: Well, even with that. Say I have a swinging blade. I spawn it on the ground. (It usually has the handle just spawn below the ground at this point). A player might get confused where did it go? With a root block, this wouldn't be the case, and he would actually feel like something was placed.
- Flo: of course the blade would be shown and placed above the ground.
- Nihal: Hmm, okay. Let's move to point 2.
- Nihal: 2, If we do spawn it like like this, then while generating the structure template, it would be harder to save it in the json string. Like the chest is added simply to the Json when a chest block is detected. If we have a root block, we can easily account for it while creating a Structure Template.
- Flo: I think the code that generates components from a ingame template can just be modified to ask the entity manager for all entities that have a SwingingBladeComponent(assuming that this is the component the root entity has) then it could deter
- Nihal: The entity manager has no getBlockAt(pos) method afaik. Iterating over all trap entities would be more expensive than iterating over only blocks in the given structure region.
- Flo: yes but there won't be 10000s of traps and structure template creation like this happens rarely and not during normal play. So you could do a quick check the origin is in the set of block positions that belong to template. If it becomes a performance issue we could let structure template origin blocks track what entities belong to them
- Nihal: Alright. I still prefer the root block approach, though. Feels more intuitive to me. With the block approach too, we can make the static box interactive.
- Flo:not so sure about that
- Nihal: Why not? It would be the same way you'd do it without the block.
- Flo:no the difference is that the entity that got activated differs
- Nihal: I know, but for practical purposes, we can easily obtain the root entity by getOwner().
- Flo: you mean the root block?
- Nihal: Yeah the root block entity. The root block entity would have the SwingingBladeComponent which needs to be changed to alter properties.
- Flo: still not sure. Currently the interaction screen component can be used for directly activated entities

- Nihal: There is a root block that has a parent prefab as the root entity. Similar to the chest. The root block is interactive with the UIScreen that allows properties to be changed. Now, the mesh can be given a static box that on interaction gives the same UI Screen.
- Flo:i guess that workaround would work for sure
- Nihal: What do you mean by interaction screen can work for differently activated entities. Like how I meant “gives the same UI Screen.”?
- Flo: to what do you refer what I said?
- Nihal: “Currently the interaction screen component can be used for directly activated entities”. What does this mean?
- Flo: I mean that if a parent has the interaction screen component then activating a child would not result in an activation of it.
- Nihal: Okay, I did not mean it like that. I meant that the mesh entity would just find the SwingingBladeComponent from the owner, and be attached to the same UIScreen.
- Flo: so would blades placed by structure templates also have a root block? I think they should for consistency
- Nihal: Yeah they would. To limit editing, you said that’s already something we have to solve.
- Flo: would that block if mined remember trap settings?
- Nihal: It can be made to. I’ll tell you what I figured out today. So I created the root block for the swinging blade. I realised that it created the SwingingBladeComponent as soon as I did a `give swingingBladeRoot`. The problem was that the OnActivatedComponent for the SwingingBladeComponent has the logic which creates the rod, blade and mesh entities. So as soon as I get the item, even though it’s in my inventory the blade gets created somewhere.
- Nihal: What I did to solve this was: I created a SwingingBladeRootComponent. When a root block is placed, the SwingingBladeRootComponent is removed and the SwingingBladeComponent is added. Vice versa happens when the block is destroyed. So essentially the SwingingBladeRootComponent could have the trap settings even after the trap is mined
- Flo you could also just activate the mesh when also the block component gets activated
- Nihal: Yeah. Didn’t think of that. :/
- Flo: have a look at origin block. There is some block <-> item logic to write
- Nihal: block item logic to write?
- Flo:I think you already did what I mean. There is a event when a block becomes an item and vise versa
- Nihal: Okay, so are you ok with using block as the root? Or do you want no-block, entity-only root?
- Flo: if you want it I am fine with it. It has some advantages to have a origin block. How would it look like? Invisible block at origin or a wood block?

    - Nihal: Hah, I used a wood block for now. The same as the chest block. But I feel different traps should have different how does the block look requirements. Also you yourself said, some traps could be block based. Like a floor which gives way.
    - Flo: yes. I guess we will find a unique look for it and if it is just a slightly smaller and less high wood block with a blade symbol on top side that is normally hidden by ceiling
    - Nihal: Yeah, for different traps changing the root block tile should also be easy.
    - Flo: so just to confirm: the root block would not allow trap selection but would also allow property edition
    - Nihal: No trap selection. Only property editing. Basically the same UI Screen as you saw for the trap config tool in the video.
    - Flo: ok sounds good and simple to use
    - Nihal: Nice, let's proceed with the design discussion.
- **Each trap will have a unique looking root block (custom tiles and shape)**
- **Root block is owner of the root entity (like chest)**
- **Root block should have a UI Screen that has the settings for the trap**
- **The structure for a general trap would look like this:**

Root Block/Entity

- **Server children entities**
- **Client children entities**
    - **One client side entity can have a static box which can be made interactive to open the same UI screen as the root block.**

    - Flo: have to go. Sounds good so far
    - Nihal: Okay, I'll update a little and get to work.
    - Flo:
    - Nihal:
    - Flo:
    - Nihal:
    - Flo:
    - Nihal:
    - Flo:
    - Nihal:
    - Flo:
    -