

Player Moveset Restrictions

The player has a number of moves available to them, these moves are used for both traversing the environment and defeating any enemies ahead.

Restricting said moves in one way or another makes it so the player is forced to put some thought behind their actions, preventing button mashing and making the experience more interesting. With this being said, the amount of restrictions and their permissiveness will dictate how much does the playstyle lean towards high strategy and well thought out sequences of actions over reactivity, improvisation and a generally more fluent, less punishing one. Note that some clearly unfit approaches will be left out of this analysis, such as League of Legends' Mana, since it is built around a multiplayer system ment to be fair for all parties instead of a fluent single player experience.

Before getting into the actual analysis, I'd like to point out a common aspect between all approaches that might seem obvious but is still worth mentioning. All moves have a base "cooldown", simply put, the amount of time required to perform an action from start to finish. We need this basic element to be present in every move in order to prevent any overlapping and malfunctions. This rule must only be broken or bent if the design of the abilities support said overlapping.

None

As mentioned before having no restrictions whatsoever is quite dangerous, in our case specially, since there are some moves that are useful both for defense and offense -like the dash for example-. That would result in a pretty easy to execute playstyle that makes the player both unkillable and a murder machine at the same time.

Cooldown

The basics of this approach are pretty self explanatory, all abilities have a different cooldown depending on how crucial and strong they are. The abilities enter in cooldown once they are used and won't be usable until said timer reaches zero.

Next we will see a list of elements that are sometimes present when it comes to ability management with cooldowns. These elements are great at giving lots of depth for a minimal production cost, in exchange for great amounts of complexity.

- **Charges:** This element allows players to stack multiple uses of a certain ability, once one of them is used a timer is started, but the ability will still be castable until there are no charges left. This factor does not let the player spam a certain ability mindlessly, yet it still gives them the opportunity to use it in a quick succession.
- **Resets:** Once a requirement is met, the ability cooldown drops down to zero, no matter the time remaining. This can have amazing results since it spices up the gameplay a ton but it requires a very specific moveset and environment designed around it in order to be relevant.

When it comes to communicating said cooldown to the player, there are many ways of doing it. League of Legends shows it very clearly in a box on the UI, Ultrakill displays the Dash cooldown as a bar both on the bottom left of the screen as well as a semicircle around the crosshair, and lastly Hades, which does not show your Dash's cooldown at all.

Focus

The name of this section might not be as self explanatory as the other ones, but I have not found any consensus on to how to call the system that we are going to evaluate next.

The name comes from probably the most famous use of this approach, Hollow Knight. The Focus in Hollow Knight is a resource gathered when attacking enemies, and players can spend it on defense (healing) or offense (long range attacks, large explosions...). Having this two options when it comes to spending said resource is not required per se, but it adds a lot of strategy and depth to the gameplay. Because of this, having Focus without the defensive or offensive side is possible yet no ideal.

Before diving into how this approach fits into our moveset, let's first see what elements are present in it:

- **Source:** Elements that exist in the world which the player can use in order to gain Focus.
- **Obtention:** A way to get said resource, the way the player interacts with the Source. Whether it is by attacking, being close to something, interacting... An important aspect of this obtention method is that the player must always have a way to obtain resource without the need of spending any.
- **Storing:** Somewhere where the game keeps track of the amount of Focus the player currently possess, whether it is shown to the player or not.
- **Spending:** A way to spend the Focus, offensive or defensive moves that are available only through this investment.

This approach does not seem like a bad fit in the beginning, but the fact is that the moveset was not designed with this restriction in mind, and I believe they are heavily incompatible for one main reason. The current moveset is ment to be used both for combat and space traversal, using all moves constantly in order to do so. When talking about player moveset restrictions, the use of Focus implies that there is a range of moves only available at the cost of spending said resource, entering in direct conflict with the nature of the moveset. If the game constantly asks the player for a certain resource in order to do a move that was originally thought out to be quite common, it becomes:

- **Too harsh or too irrelevant (Combat):** In order for the player to use the moves as much as it was originally planned, the spending of the resource must be way smaller than the obtention, which makes the system pointless altogether. If not, the game becomes too difficult to play, since something as basic as a dash spends a resource gathered only through more combat.
- **Too boring or too noisy/annoying (Space traversal):** When it comes to space traversal, we find ourselves in a similar spot, making this resource abundant makes the system completely pointless, but if this is not the case most/some of the moves would not be available for regular use, which would make the traversal boring. In order to fix this we could scatter around different objects so that the player is able to replenish their focus as they move. This would not only solve the problem mentioned before, but it would also be a great way for skill expression, since it would convert every instance of traversal into a puzzle to solve on the go. With this being said, I still do not think this is a good fit for the game, since such a unique way of moving around would require a special focus, I see it as one of these elements that should have the whole game designed around them, and not otherwise. Making the traversal require so much of the player would result into having no actual rest zones for the player, thus ending up as an element that just puts unnecessary noise into the game, or just makes traversal an annoying experience overall.

Stamina

Stamina is somewhat of a complex system in of itself, and a lot of values play a crucial part in it's functioning, so let's start by breaking it down:

- **Resource:** The Stamina, a number that can be represented in a variety of ways in order to communicate to the player their current state. The information that they require are the current and maximum amounts.

- **Spending:** A way to spend the Stamina in exchange for the ability to make a certain move.
- **Regeneration:** Stamina regenerates passively when a certain amount of time passes from its last use.
- **Punishment:** Running out of stamina is usually punished in one way or another, making it so that the stamina system is built so the player ends up learning how to never completely run out of it, or at least, only spend everything with a plan in mind, consciously.

Variables involved in any generic stamina system:

Max. stamina - [MS] (stamina)

Regen. rate - [RR] (stamina/s)

Regen. delay - [RD] (s)

Recov. Regen. rate - [rRR] (stamina/s)

Recov. Regen. delay - [rRD] (s)

Ability related:

Cooldown - [CD] (s)

Cost - [CS] (stamina)

Cast Time - [CT] (s)

Thanks to this complexity, it is also very flexible. In order to make the stamina system more permissive we could do a number of things:

Action	Consequence
Lower overall costs	Mindless casting is less punished. It also allows for longer combos
Lower the amount of moves that require Stamina	Certain moves become available to the player at all times (other gameplay requirements to use said abilities aside)
Decrease the amount of time required to start recovering stamina after any spending	Less downtimes/waitings between combos. If this value becomes too small the player might end up regenerating more stamina than spending it
Increase the rate at which stamina is recovered	Smaller downtimes/waitings between combos
Remove or tweak the punishment, e.g.: Not lowering the player's speed. Not forcing them to drop the anchor. Not delaying the Stamina regeneration. Not limiting stamina spendings as the stamina itself is regenerating.	Higher permissiveness when it comes to running out of stamina. If this element is more permissive, it is also easier to make players fall here and learn how to avoid it in a less frustrating way. Going too far will make the whole system useless.

Balancing

The equation we will see later on will show us a variety of things when spamming an ability or a sequence of them.

We will also add an extra variable, one that takes into consideration the amount of time that it takes for a human player to cast an ability once again. This might seem pointless for certain abilities, but is extremely important for others, especially if they have any requirements for casting, such as aiming, repositioning or similars.

Human reCast Delay - [HrCD] (s)

If there are two abilities thought out to be used together a lot of the times and one of them has no cost whatsoever, then both of the cooldowns should be added in order to accurately represent the time passed between stamina spendings.

$$\begin{aligned} \text{Stamina regenerated between spendings} &= \text{Stamina spent} \\ ((CD + CT + HrCD) - RD) * RR &= CS \end{aligned}$$

 $((CD + CT + HrCD) - RD) \rightarrow$ Total time in which the player is regenerating stamina

In order to take into consideration a set of abilities that meet the following criteria:
First cast an ability that spends stamina followed by an undetermined amount of moves that are free.

$$\begin{aligned} &\text{Ability(Cost)}^1 \rightarrow \text{Ability(Free)}^n \rightarrow \dots \\ &((CD^1 + CT^1 + HrCD^1) + (CD^n + CT^n + HrCD^n) + \dots - RD) * RR = CS \end{aligned}$$

The bigger the left side is in comparison to the right one, the more permissive the stamina system is, since the player will regain stamina by using this set of abilities.

If both sides are equal, that means that no stamina will ever be lost.

The smaller the left side is in comparison to the right one, the faster the stamina will deplete.

This equation means almost nothing on it's own, but if we combine the results from all abilities and combos we will get a pretty accurate view on the current state of our stamina system.


If we move the CS to the left side of the equation, we get the amount of stamina we are left of after completing the combo as a result. For more complex combos we only need to check if the remaining stamina is enough to use the next move and concatenate more calculations.

Options

What are we looking for?

Before diving into any of the different possibilities, we should first have a clear goal. What is it that we are looking for an ideal system?

- Frequent and mostly unpunished use of abilities for a satisfying space traversal.
- Prevent mindless combat.
- Give room for a variety of ways to use the moveset efficiently (depth).
- Simple to communicate, easy to play and understand (low complexity).

 With this things in mind it is somewhat obvious that we will need to sacrifice one thing or another. Since depth always comes at the cost of a higher complexity and we do not have the resources to make a long game, if it comes to it, the best decision would be to sacrifice in depth for the sake of a more easy to enjoy experience overall.

Specific notable cooldown additions

This approach is based on keeping the “base cooldown” or a very low one for part of the moveset and add a significant one to the other, specially moves that are strong. In our case it would end up something along the lines of.

- Base Cooldown: Throw, Retrieve.
- Significant Cooldown: Roll, Dash.

✖ Without any kind of resets or charges, this system gets really shallow and has no depth nor any flexibility. This approach would only be viable if we end up going for a much more complex ruleset.

Stamina rework

When it comes to stamina, there is one question that rises above the rest, a question that completely dictates the players relationship with the move set:

What are we looking for, small downtimes or less downtimes?

I believe that it is best to wait less more frequently, since otherwise mindlessly spamming abilities could end up being the dominant strategy. This is accomplished through increasing the rate at which stamina is recovered and decreasing the time it takes for the player to start this regeneration after the last spending.