

NO MAN'S SKY HARVEST GAME LOOP

Breakdown / Analysis - Tristan Ledieu

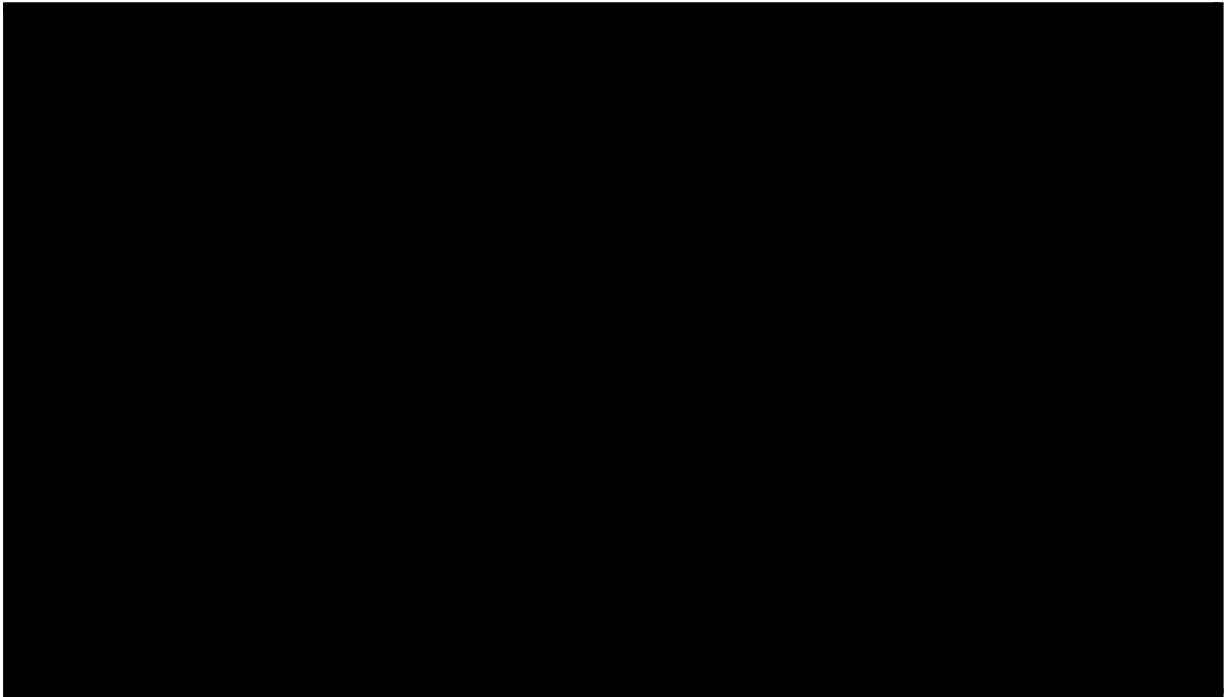
Estimated reading time: **10-15 minutes**

Note that you can **click on the diagrams** of this document to get their direct link and open them in **full size**, in case you would like to see them in detail.

INTRODUCTION

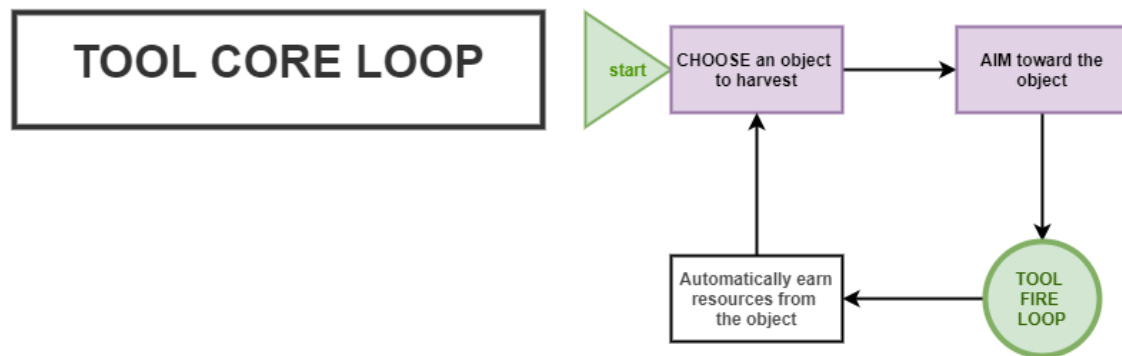
Before we begin, let's state the context around this **harvest feature**, as well as its basic way of working.

In No Man's Sky, you can use an **upgradable tool** to **harvest** different **objects** in the environment. This tool works like a **gun**, firing a **continuous laser** that damages the **health** of a targeted object by a given amount every brief time interval. Once the health of the targeted object **reaches 0**, the object is **destroyed** and the player earns **resources**. The type of resources that can be earned from an object depends on the **object type** and can be **identified** and **memorized** by the player beforehand. The player can then **use** these resources for **other features** of the game, like crafting or trading.



THE CORE LOOP

The base of the Harvest mechanic use a fairly simple core loop:

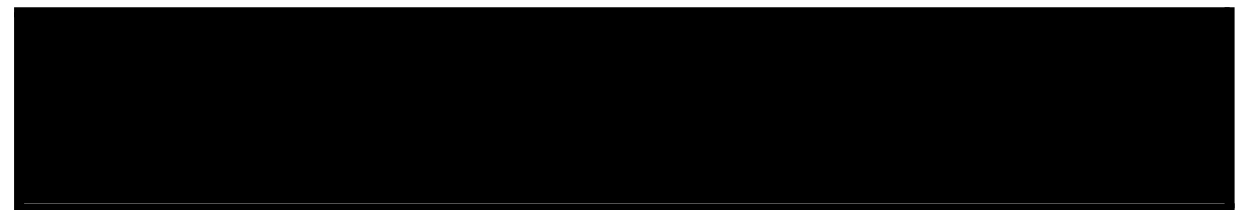


For this loop by itself, there is not a lot more to say than the diagram. First, the player **chooses** which object they will harvest, then they **aim** at this object. After this comes the tool fire loop, basically they **hold the fire button** to channel the harvest beam, damage the object until its destruction, then **release the fire button**. There is way more to say about this “fire loop”, but we’ll get to it in a few lines. Finally, they automatically **recuperate** the resources of the destroyed object.

This simple loop already creates a “simple but effective” harvest/mining/gathering **mechanic** similar to what we can find in a lot of games that implement that kind of feature. The **main challenge** of this mechanic simply being to be able to **keep aiming** to the target until its destruction.

Nonetheless, in No Man’s Sky, this core mechanic is layered and complexified by another one: the **tool heat mechanic**.

TOOL HEAT MECHANIC



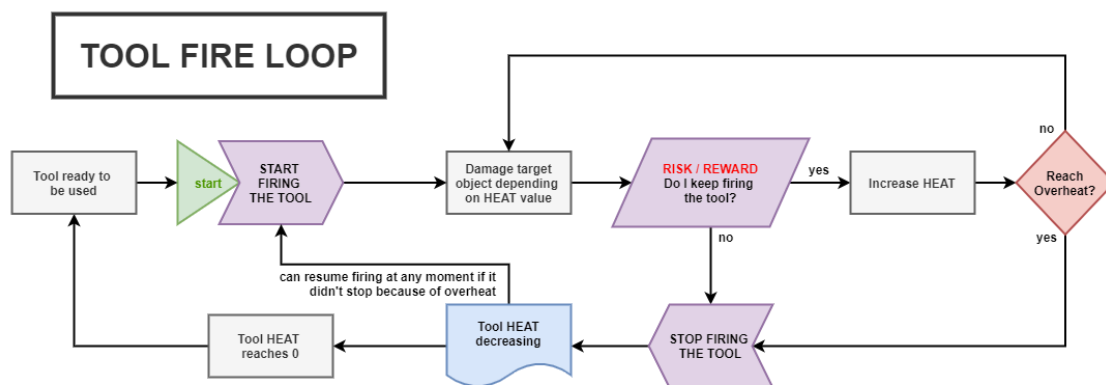
The **harvest tool** has a **heat** value which **increases slowly** while firing, and **decreases quickly** while not firing. This heat value is represented as a **jauge** to the player. If the jauge **fills completely**, the tool **overheats** and cannot be used for a few seconds. The player thus has to **juggle** with the heat, alternating between firing the weapon to harvest, and letting it cool down a little.

But on top of that, the tool becomes **more efficient** and thus harvest **faster** as its **heat approaches the maximum value** (up to x2.5 faster). It means that the player is rewarded for keeping its tool heat at a high value, taking the risk to reach overheat if not managed correctly. The tighter they juggle with the heat, the more risk they take but the faster will be the harvest. Basically, this creates a simple **challenge** with a **RISK/REWARD** lever which relies on a **Player Skill** specific to this mechanic. This simple thing completely **changes** the “pace”, the **dynamic**, of the harvest core mechanic as it now rewards the player for **actively investing brain load** in the action instead of

passively pushing the button. Despite using a quite “**low difficulty**” challenge, this makes the action more **engaging, capturing** the attention of the player by forcing them to stay alert.

Different elements of the game show that the designers put a **particular attention** on playing with this specific mechanic. For example the fact that the **tool heat increases slower** as its heat **becomes higher**, and that it **does not instantly overheat** when reaching 100% heat, instead letting a ~1s time frame for the player to stop firing and “juggle” with the heat. This compliant approach was clearly chosen to **facilitate** “high heat” **juggle**, and encourage players to **take risks** when using the tool. Another example is the fact that we can find **upgrades** for the tool which **reduce the heat up speed** of the heat juggle, allowing the player to stay longer in high heat value, thus facilitating the juggle.

So, if we link together everything we saw above, we can now detail the “**TOOL FIRE LOOP**” visible in the CORE LOOP that we simplified earlier:



So, we saw that this **simple mechanic** adds a **lot of investment** from the player to the **harvest action**. But let’s not stop here. This simple mechanic also affects the **behaviour** of the player even **outside of the action** by itself: it affects its **decision making**.

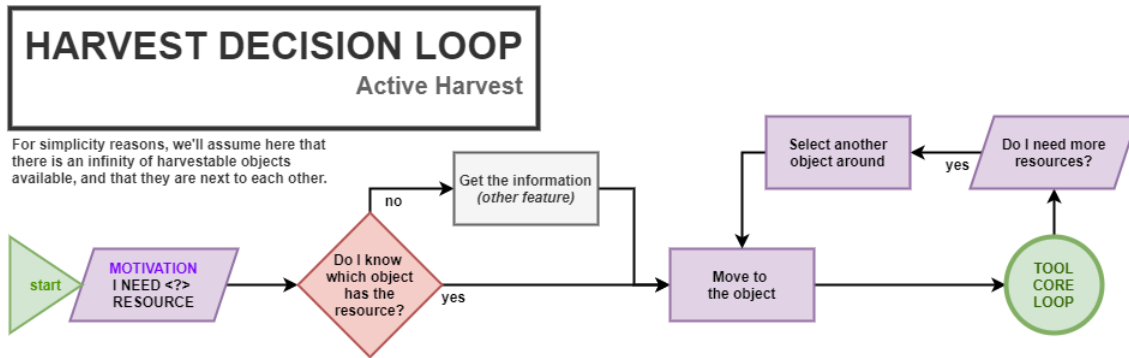
To properly demonstrate this, we’ll first dive into the player **decision loops** that drive this whole Harvest feature without thinking of this mechanic, then we’ll see how it affects it..

DECISION LOOPS

As you can guess from the few game elements introduced before, the **MOTIVATION** which leads the player to the **DECISION** to **harvest something** is always: “***I need these resources***”. There are mainly **two different contexts** where this motivation appears. We will call them “**Active harvest**” and “**Passive harvest**”.

ACTIVE HARVEST

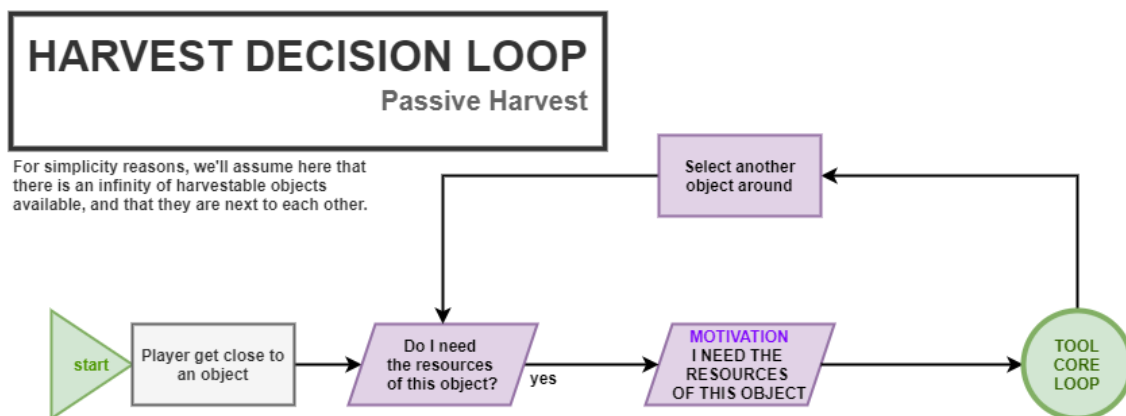
The **Active Harvest** is the situation where the player **realizes** that they **need a specific resource** that they can harvest nearby for another feature of the game. The **motivation** is there from the beginning, the player will go **harvest** for a **precise reason**;



Note that in this loop, the “Do I need more resources?” decision of the player can also be fueled by the anticipation of **future needs**, which can become a motivation to **continue looping** because “While I’m at it, I might as well harvest more”, but this motivation usually stays **reasonable**.

PASSIVE HARVEST

The **Passive Harvest** is the situation where the player just decides **on the fly** to harvest an object. They are **not particularly searching** for this object or its resources, they just **stumble upon it** and casually decide to **harvest** them.



In this *passive* situation, the **motivation** of the player is kinda “reset” after the TOOL CORE LOOP. It means that after harvesting the object that motivated them, the player theoretically **stops being motivated** and gets back to its default “passive” state, ready to be triggered again by another object.

DYNAMIC AND MOMENTUM

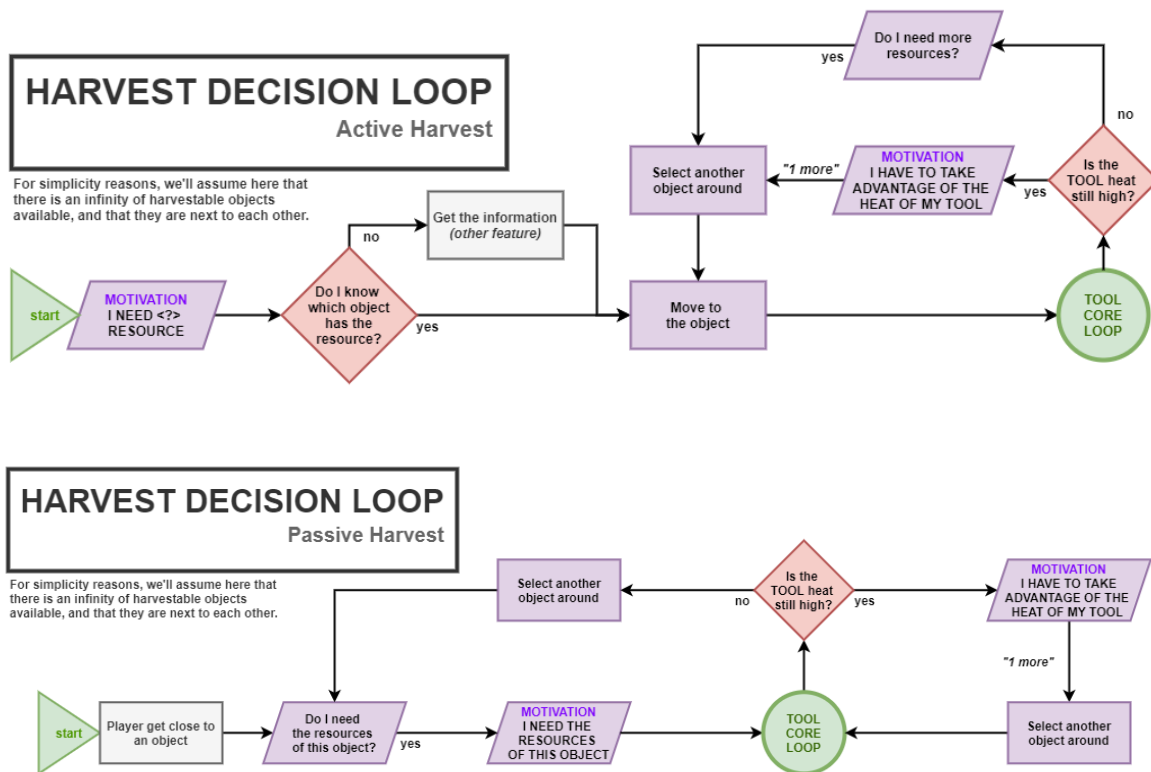
Now that we’re clear about the **foundations** of these **decision loops**, we can demonstrate how they are affected by the **tool heat mechanic**.

We saw before the **heat mechanic** in the context of harvesting **one** object. But we have to keep in mind that the **heat** is not reset after harvesting an object. Which means that if the player has **heaten up** their tool while harvesting one object, they are in an **advantageous state** as they can **harvest** another object with high heat without having to heat up the tool again. And more importantly, they will **lose this powerup** if they stop harvesting now.

This simple thing creates a strong **“ONE MORE” dynamic, motivated** by the fact that the player wants to **take advantage** of their “powerup” to make a maximum of **profits** of it. Having to **quickly chain** with another object to keep the heat creates a **strong pressure** on them to **quickly** make the decision of “Do I really need more resources?”. This **does not let them time** to properly get information about if the object is really interesting for them. Thus, most of the time, they won’t even bother thinking about this question and will just harvest “ONE MORE” object because they **“have to take advantage of their bonus”**. They thus enter a kind of **frenzy**, harvesting all the objects around until they **fail juggling** with the heat and **reach overheat**, or until there are no **more interesting objects** around.

Basically, this creates a **dynamic** where the player **thinks rationally** when they **begin harvesting**, but then totally **go with the flow** after this, to make use of their **heaten up tool**.

We can then update our two **decision loops** with this dynamic we just stated:



CONCLUSION

By just adding this **very simple heat mechanic** over the base harvest mechanic, the designers of No Man’s Sky achieved to create a game feature driven by a strong **dynamic** that keeps the player **brain loaded without forcing them** to do so. The player chooses to **invest** in this juggling mechanic because they want to be **more efficient** in their action. Even **without a difficult challenge**, this feature is enough to **retain** their attention and add **commitment** to a feature that would be a simple “input press” without it. This also makes the player reach a **feeling of mastery** with the juggle, making them take more **risks** to be more and more **efficient** as their skill increases.