Horizon Worlds - Capacity Limits

This doc represents Horizon Worlds as of September 2022*.*

Charts and graphs courtesy of Vidyuu. Watch their tutorial on capacity and optimization here: https://www.youtube.com/watch?v=UzdE8psFE-I.

Table of Contents

Capacity Meter **1**
Objects 1
Geometry 1
Simulation and Animation 2
Players 5
Dynamics Lights 5
Scripts Limits **6**
Script Capacity 6
Script Runtime Limit: Events Sent 6
Script Runtime Limit: Instructions Executed 6
Script Runtime Limit: While Loop Iterations 6

Capacity Meter

Objects

The number of things with "property panels" (including script gizmos).

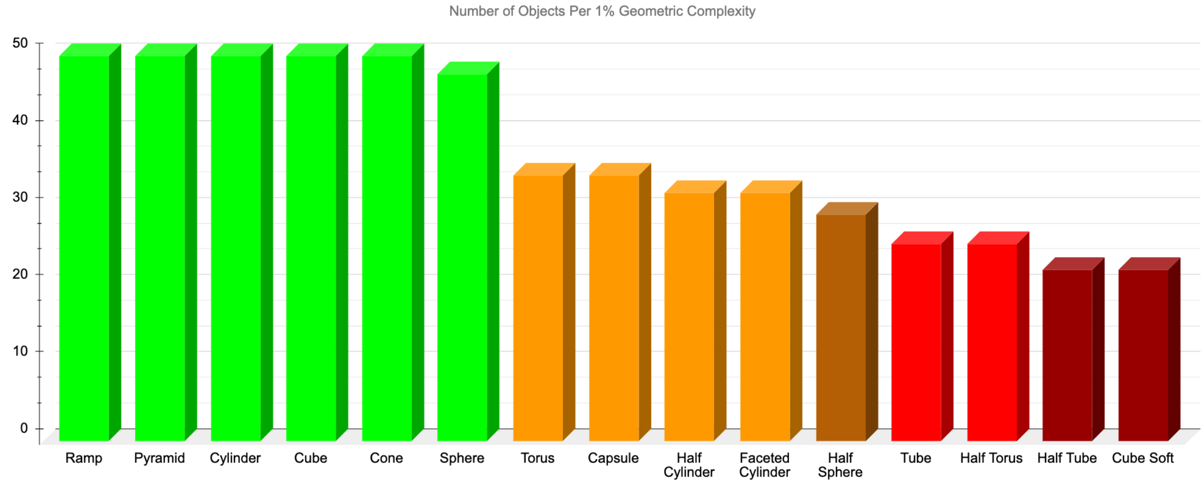
A scene can have a maximum of 3000 objects. Everything with a **property panel** is an object. Thus includes every primitive shape, every asset pulled from the asset library, every gizmo, and every group.

For example if you take 4 cubes and a sound recorder and group them together you now have a total of 6 objects (4 for the cubes, 1 for the sound recorder, and 1 for the group itself). Nesting groups in groups will also increase the number of objects.

Geometry

The amount of vertices / "meshes in the world".

There is a limit to the amount of mesh geometry that exists in a world. Every primitive shape has a set amount of complexity, as do the premade mesh assets in the asset library. Some shapes take up more complexity than others (with the rounded cube taking up the most of all the primitive shapes).



Simulation and Animation

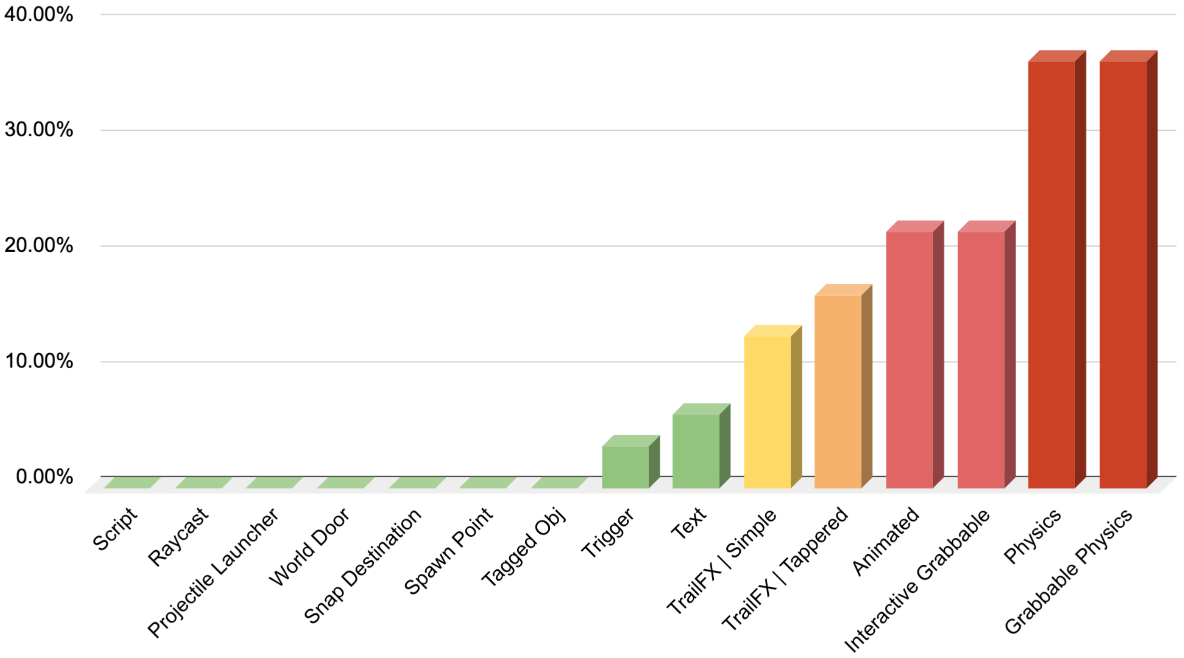
All the objects that change properties while a world is running (including all gizmos).

Animated. Any object that changes its properties while a world is running counts toward the animated budget. This includes recorded animations, objects that move/rotate/scale in a script, objects that change color / visibility / collidability, objects that get pushed / spun, etc. In order for an object to change its properties while a world is running it must be “dynamic”. When you open an object’s property panel there is a property called “interaction type” that you can set to *None*, *Animated*, or *Interactive*. If the setting is *Animated* or *Interactive* then the object is considered “dynamic”. If you connect an object with a reference pill in a property panel its setting is automatically changed to *Animated* (if it was set to *None*). If you don’t change the object (e.g. you only ever query its position) then you can put the setting back to *None* after connecting the reference wire; if you then try to modify a property while the world is running you will get an error in the debug console. There is a limit to how many animated objects you can have (see the graphs below).

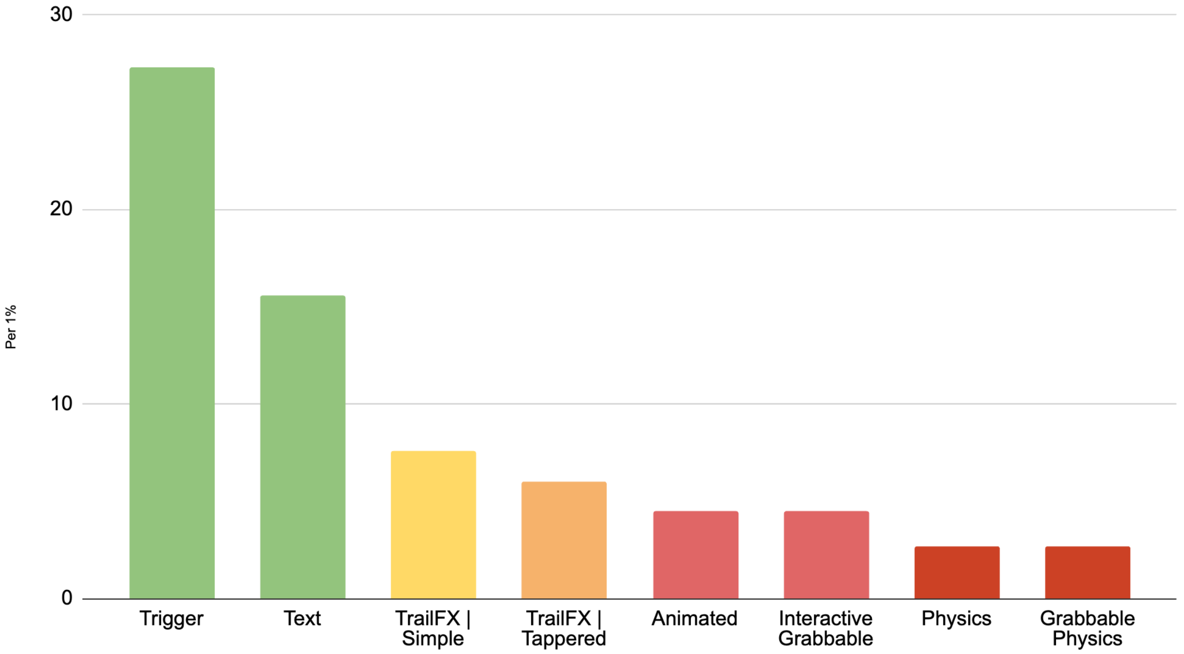
Physics. When you set the interaction mode of an object to *Interactive* the object is counted as *animated*. You can then set the object to be *grabbable*, *physics*, or *both*. If you select *physics* or *both* then the object counts toward an additional metric: the “physics capacity” of the world. This is tracking all the objects that need to be “simulated” by the physics engine.

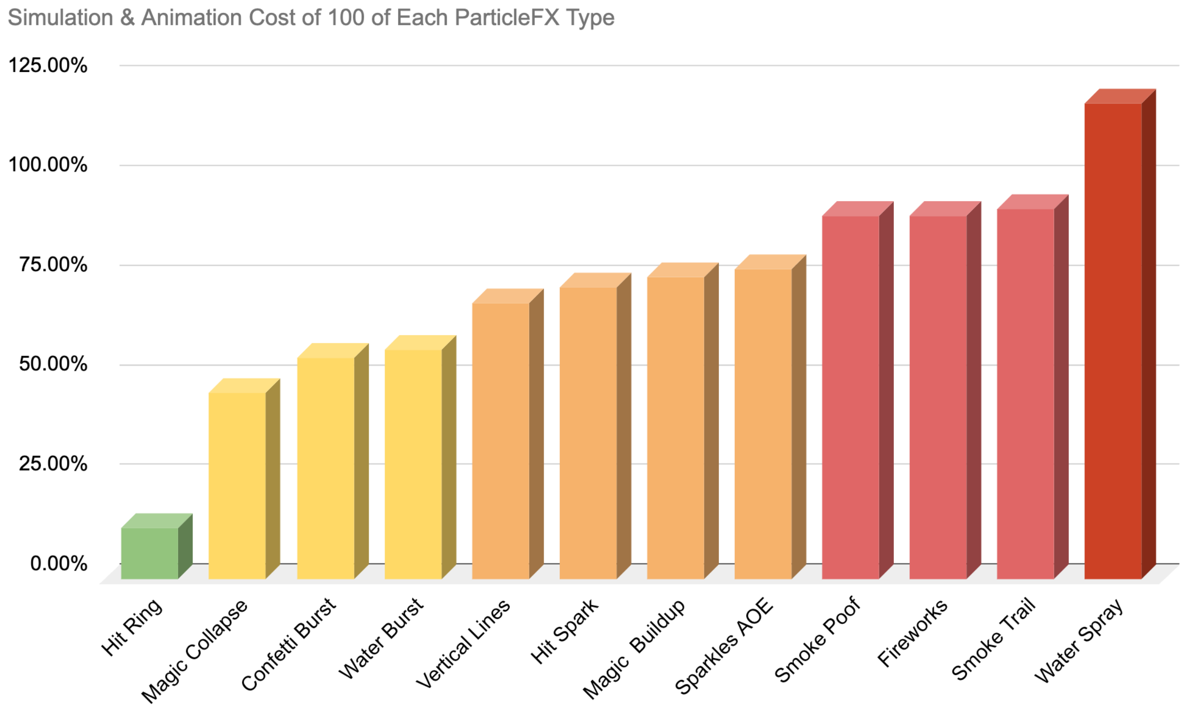
Triggers, Texts, VFX. Each trigger, text, and VFX object gets counted as well when calculating a worlds’ capacity. See the graphs below for their contributions.

Simulation & Animation Cost of 100 of Each Object Type



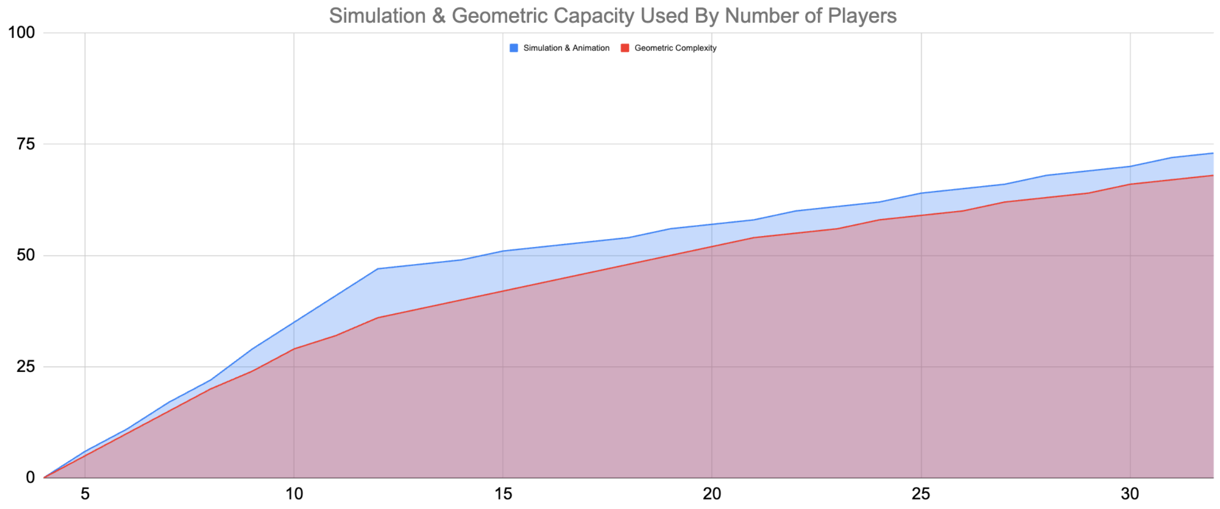
Number of Objects Per 1% Simulation & Animation





Players

When creating a world you can set the maximum number of players allowed per instance. Because avatars have to render, they take up some of the geometric complexity of the world, and since they move they also take up dynamic complexity. The higher you set the slider for the maximum number of players, the less capacity you have left for content within the world.



Dynamics Lights

A world can have a maximum of 20 dynamic lights enabled at once. Unlike all other capacity limits, this one is enforced *while* a world is running. You can see the current number of dynamic lights listed in the property panel for any light (whereas all the other capacity information appears in the Capacity section of the World tab in the Creator Menu).

Scripts Limits

Script Capacity

There is a limit to the amount of “stuff” you can put inside each script. This is roughly equivalent to “how much data does this script take to persist?” This capacity is shown in the bottom-right corner of the script editor window. Every code block in the script will use up this capacity, as will variables, comments, and initial variable data. So if you make a script with a string list that you write a lot of text in (in the editor) then all of those strings will also take up script capacity.

Script Runtime Limit: Events Sent

Horizon counts how many times the “send event” and “send event with delay” code blocks execute. This count is per-client (e.g. there is a separate count on the server and on each player’s headset) and the count is reset *every frame*. The current limit is **2048** “sends” per client per frame.

Script Runtime Limit: Instructions Executed

Every code block in Horizon executes instructions when they run. Most code blocks are 1 instruction each but some are multiple instructions. Also, if you run a loop you will be running the same code blocks repeatedly. If your scripts run too many code blocks you will hit the instruction limit. The exact limit isn’t published, however it is calculated per-client, ***per event handler***. Every time Horizon starts to execute a “when X happens” code block the counter is set back zero. With “while” loops it is really easy to hit this limit (since you are repeating the code blocks over and over in the same frame). If you hit the limit, consider breaking up complex calculations across multiple frames.