

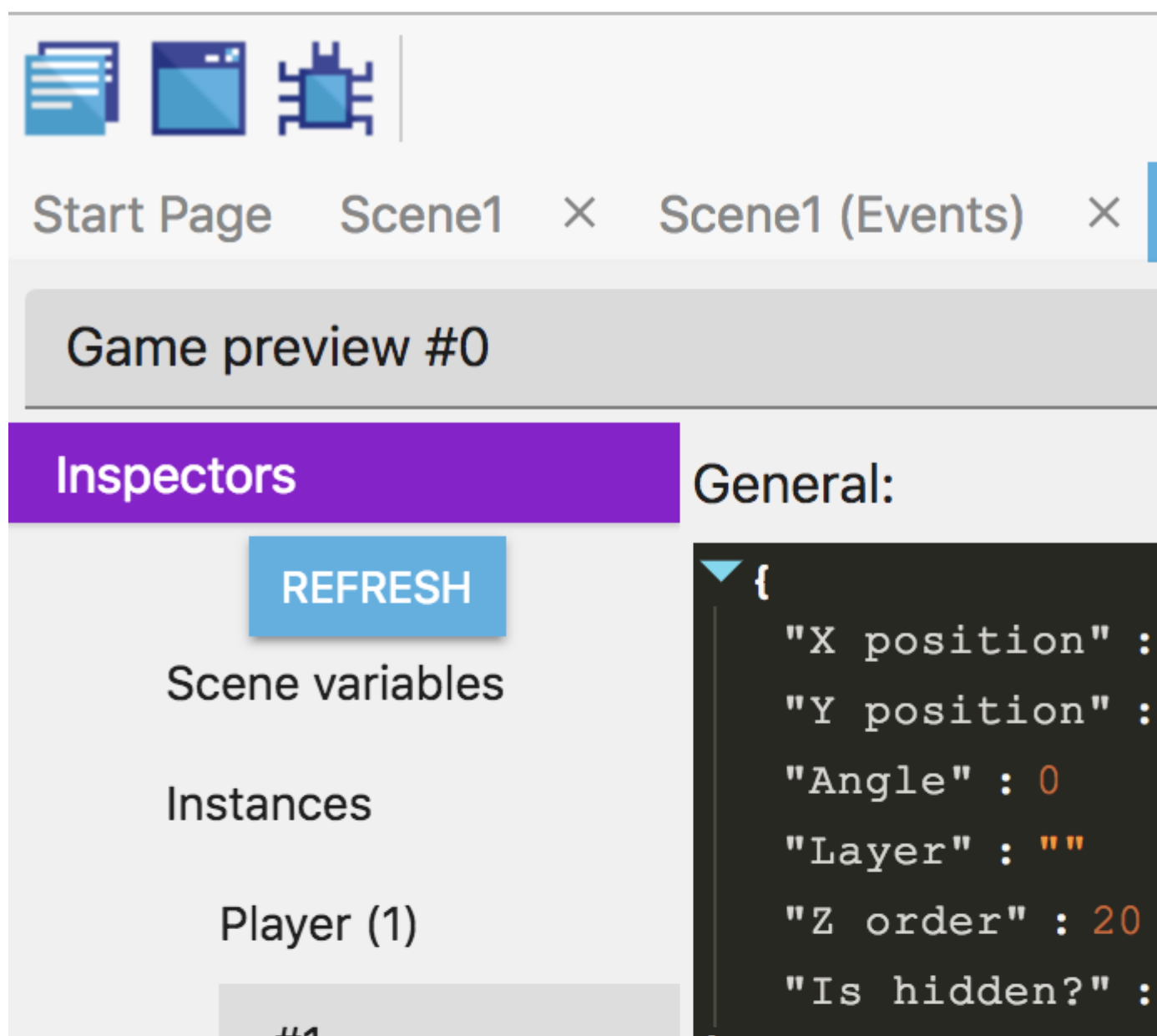
Game Debugger and Profiler

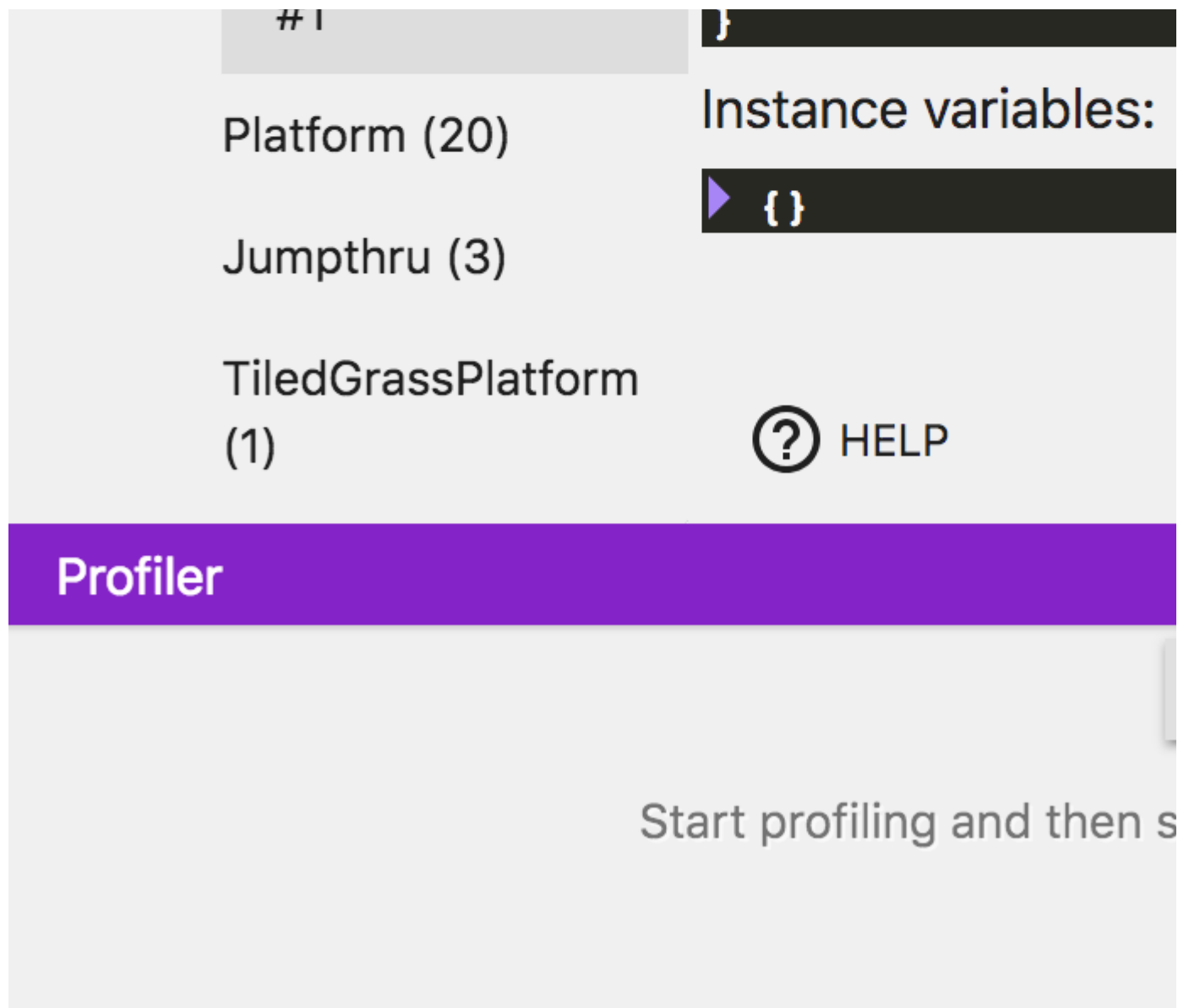
A *Debugger* is a tool that lets you inspect the content of your game. In particular, you can inspect the global variables, scene variables and variables of instances of your objects. It is a powerful tool.

You can pause the game at any time. Using the Debugger is a great way to understand why something is not working as intended or to do quick tests in the game.

The *Profiler* is a tool that measures the time spent in different parts of the engine while your game is running. It helps to find bottlenecks or events that could take a lot of resources in your game. [Read more about using the profiler here!](#)

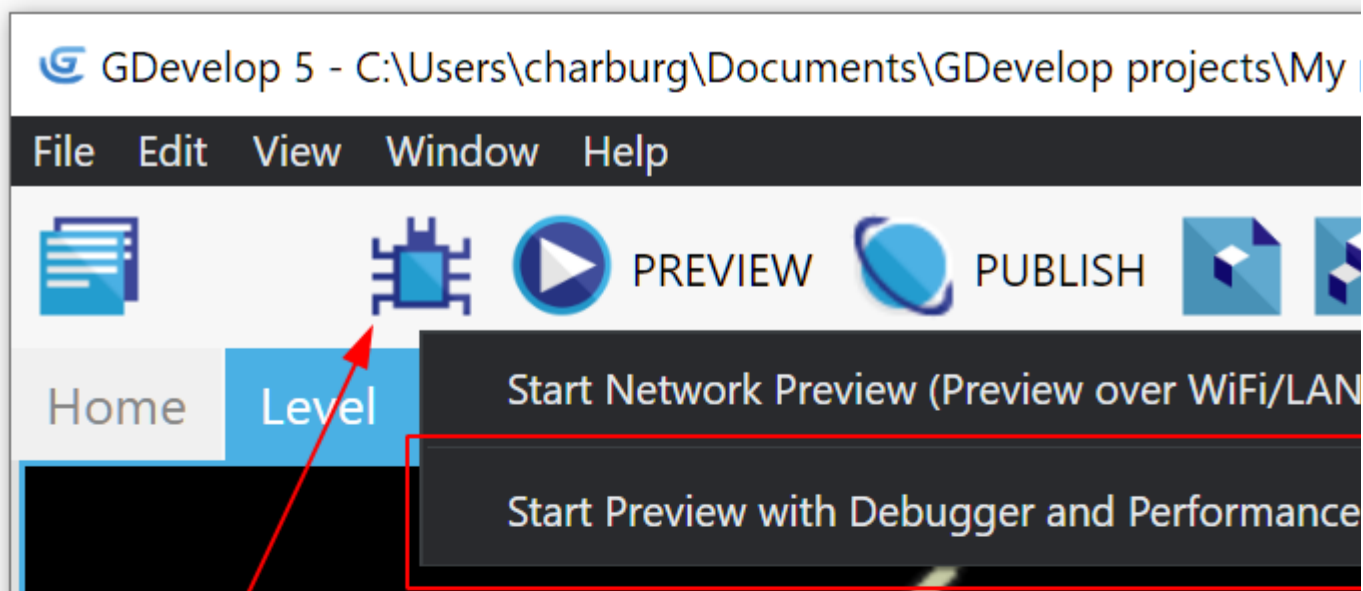
This is an advanced feature of GDevelop. If you use it, we would like you to show your support for GDevelop by [getting a subscription!](#)





Launch the debugger

To launch the debugger, in the toolbar of a scene editor, click on the “Debug” icon. In the menu, choose “Preview with debugger and performance profiler”.

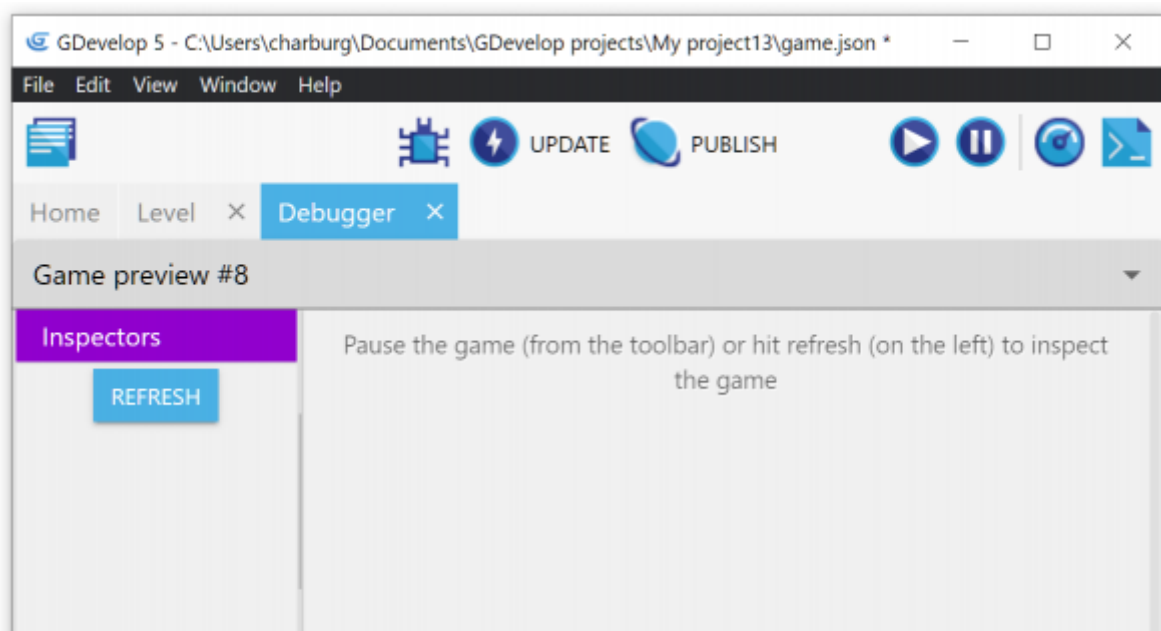


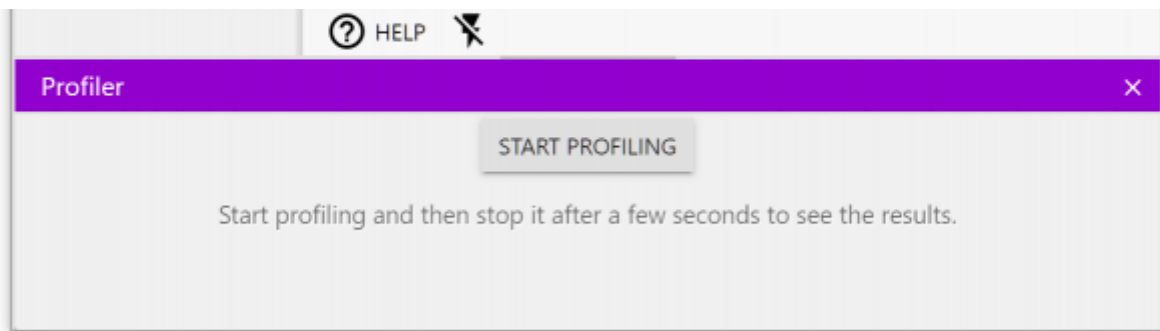


This will open a new tab called “Debugger” and will also launch a preview of your game as usual.

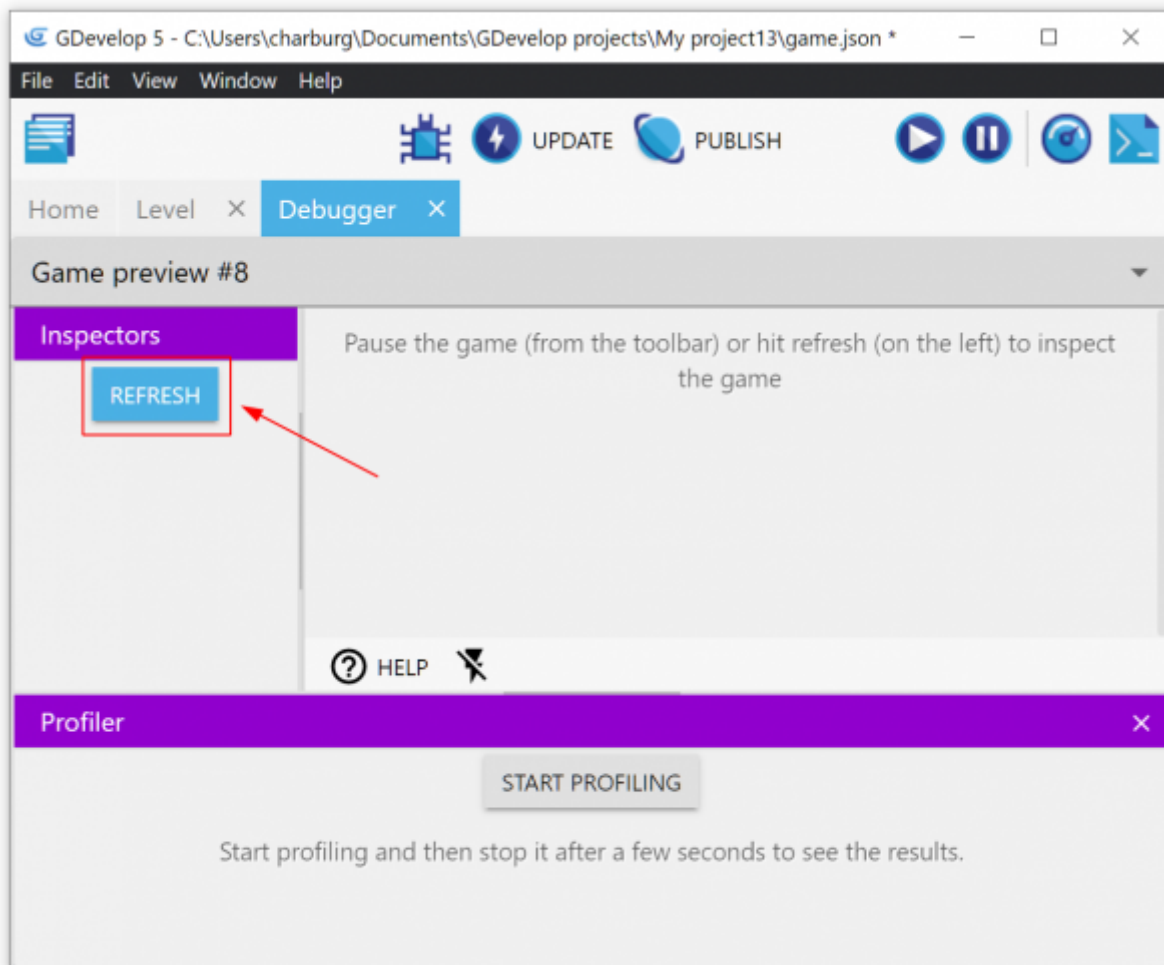
Note that even if you close the preview, you can let the debugger tab stay unclosed. This way, any preview that you launch (with the usual Preview button) will be connected to the debugger.

When the game is just launched, the debugger is empty:

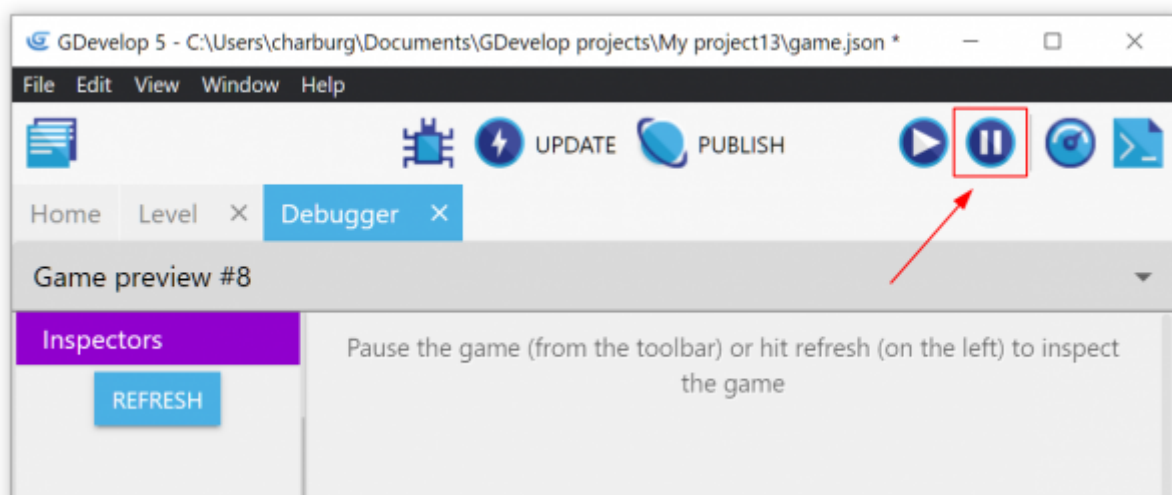


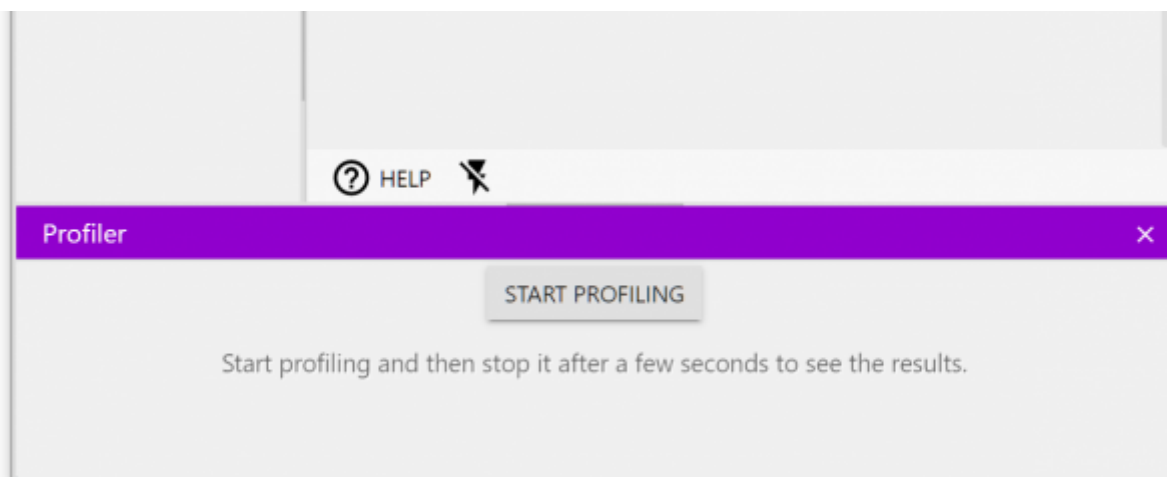


Click on **"Refresh"** at any time to fetch the data of the game:



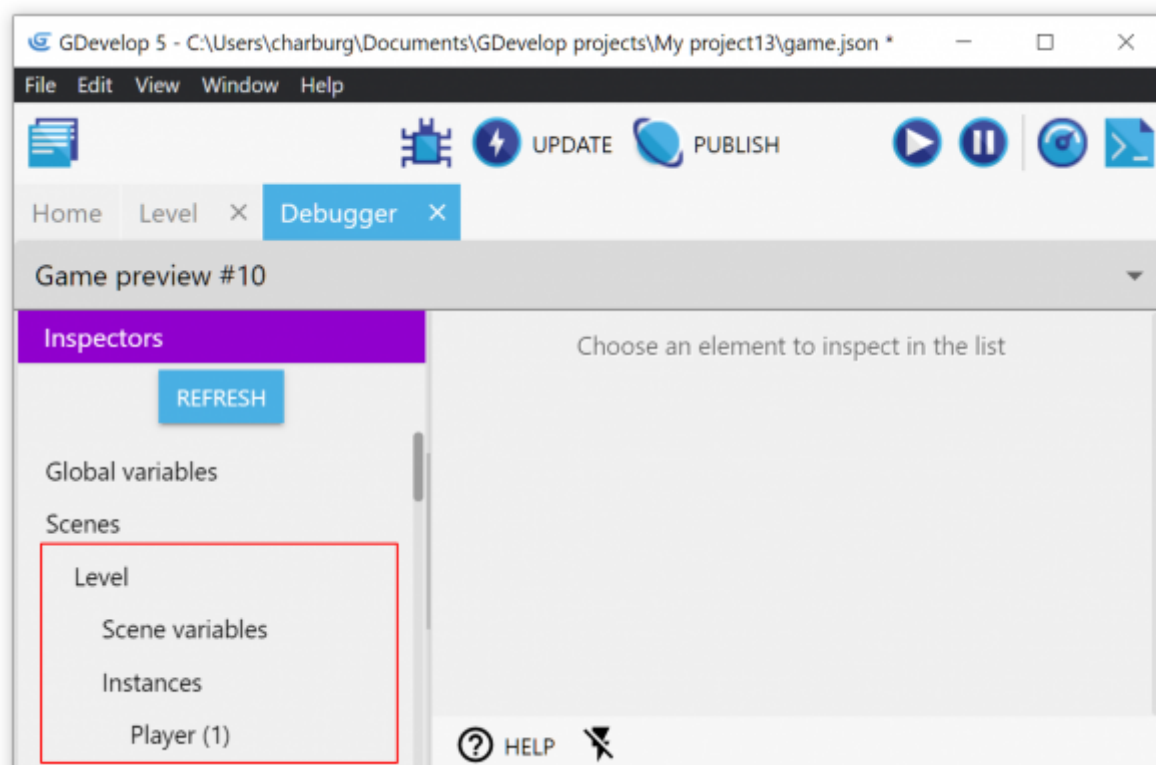
You can also use the **Pause** button in the toolbar:





Inspect the scenes of the game

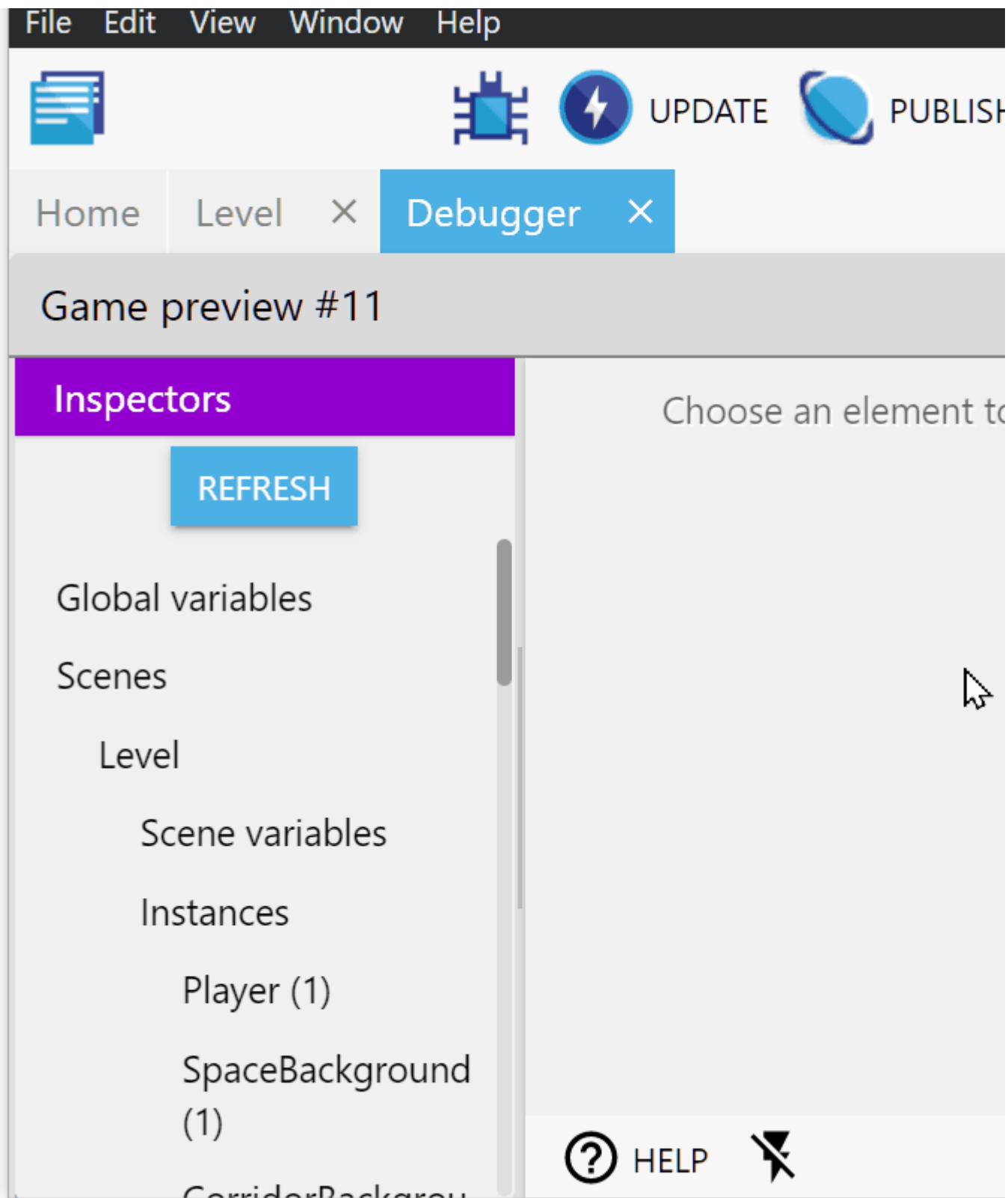
The currently running scenes are listed below Scenes item in the list:



If you click on a Scene name, you will display on the right the inspector for this scene. This inspector contains the layer of the scene and a button to create a new instance of an object on the scene.

Create a new instance on the scene

To create a new instance, click on the name of the scene where you want to create the instance. Write the name of the object and click on Create:

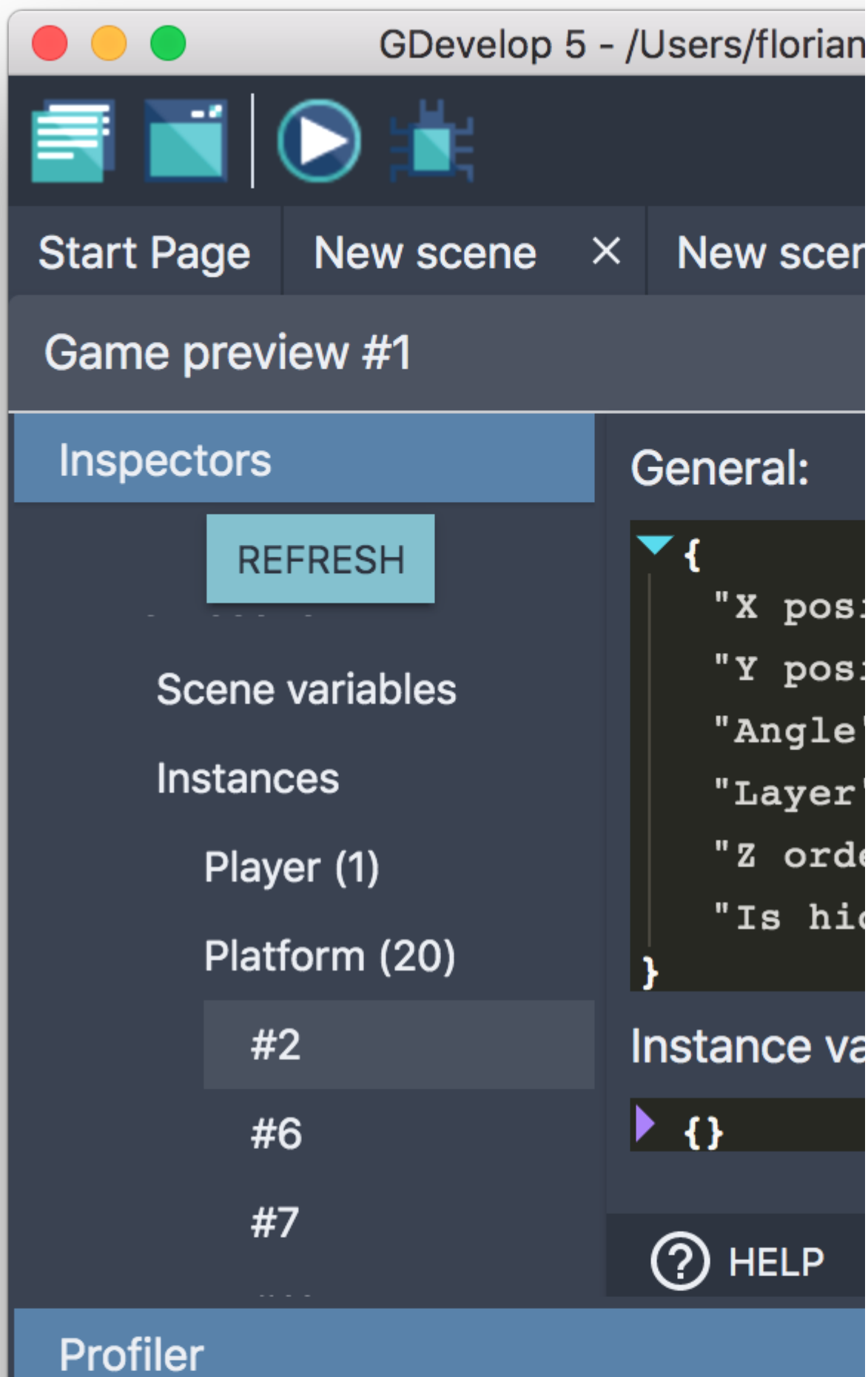


The instance will be created at position 0;0, on base layer with Z-order 0. You can then find it in the list of instances and change its position, angle, layer...

Inspect and edit the instances

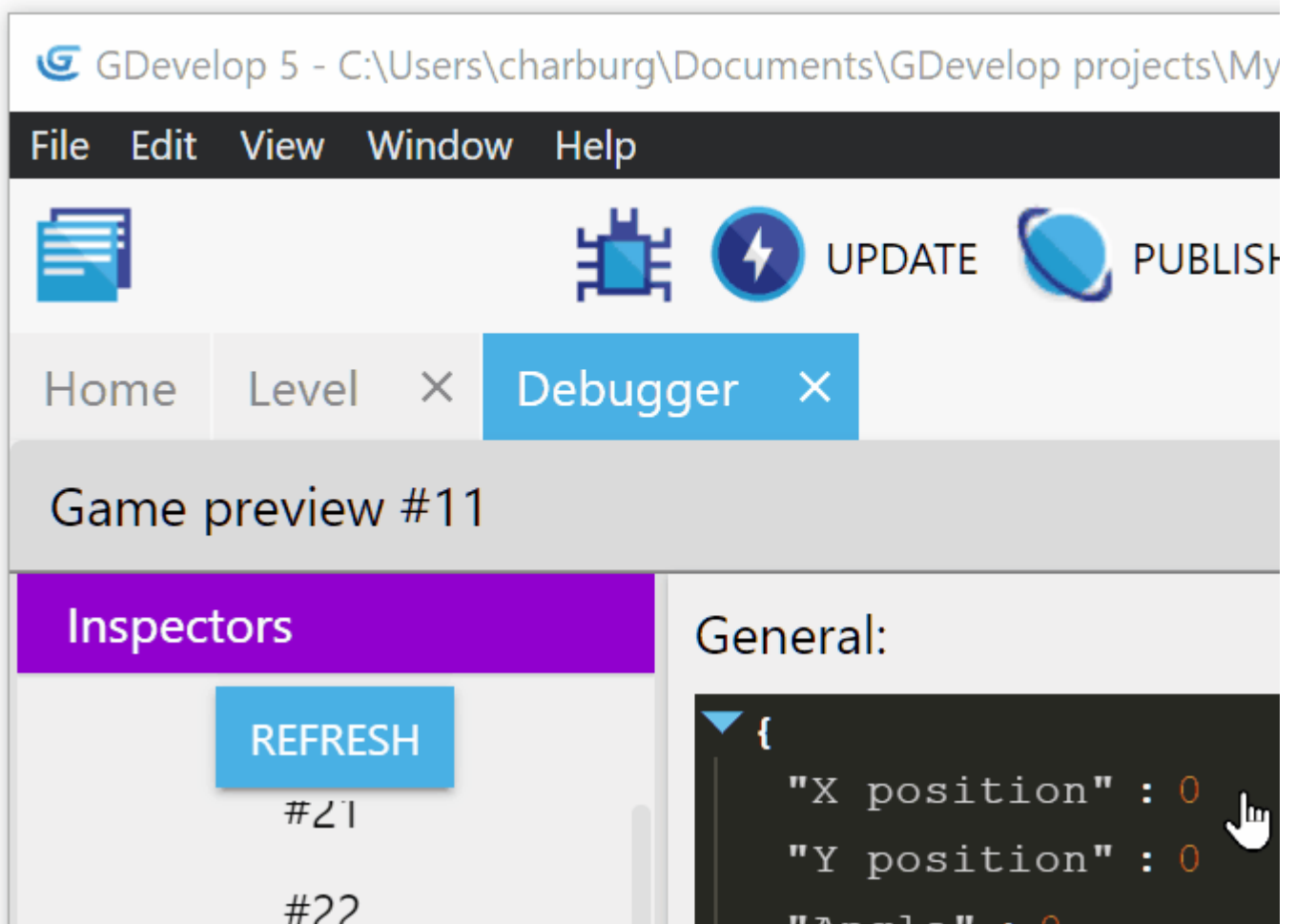
In the list, you can find **Instances** of every running scene. Instances are grouped by objects name. Click on an object name to see the list of instances of this object living on the scene. Click on an instance number (each instance has a unique

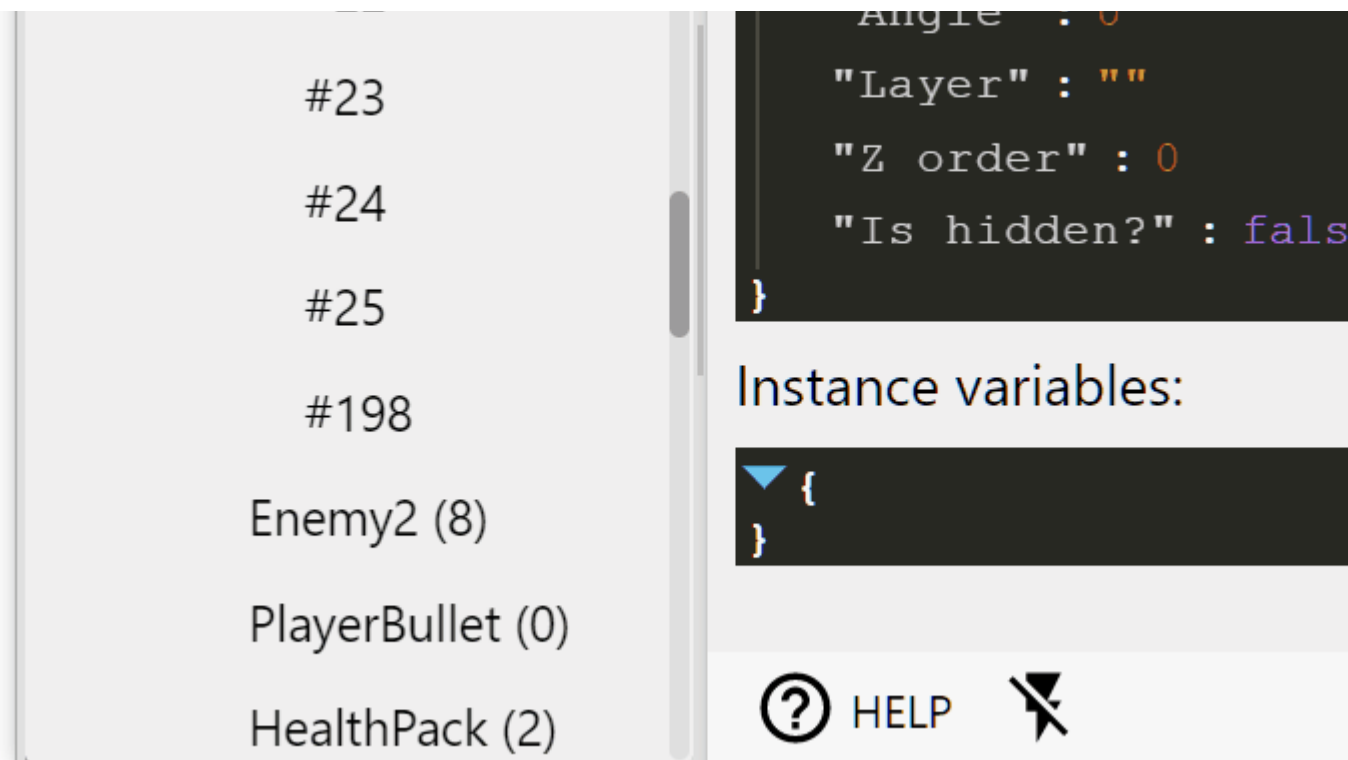
number) to inspect it:



A dark blue rectangular panel with the text "Start profiling and" in a light gray font, partially visible on the right side.

The inspector shows general properties of the instance, for example, position, rotation, layer... For each of these properties, you can **edit them by clicking on the value**:





Validate your changes by clicking on the small **validate button** (the one with the tick icon).

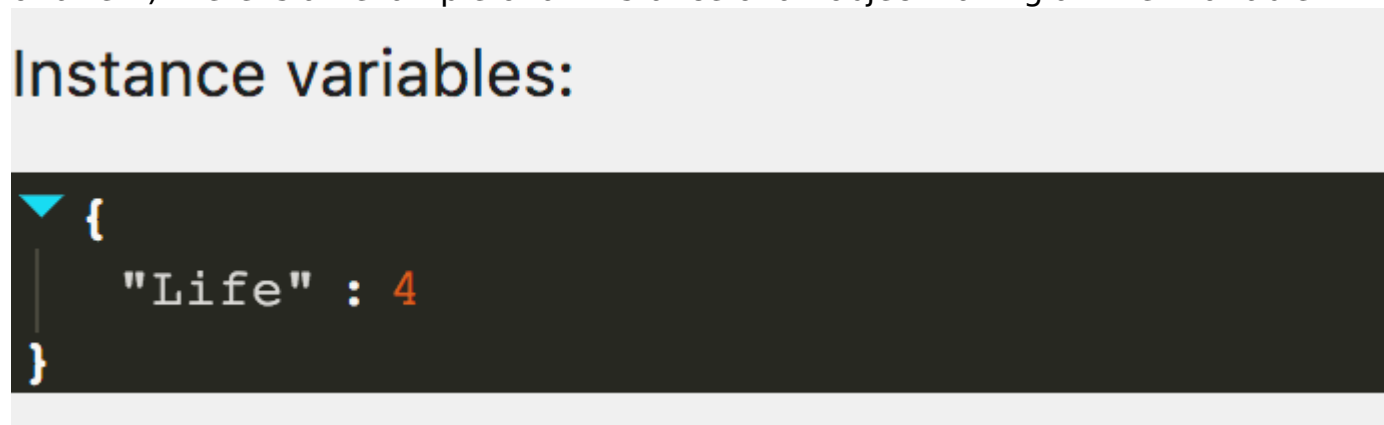
You can also see the **variables of the instance** and modify them (see below for more information about variables).

Inspect and edit the variables

You can inspect variables of the game (global variables), of a scene (scene variables) or of a particular instance of an object (instance variables).

To do this, in the list, click on **Global Variables**, **Scene variables** or the instance number that you want to inspect.

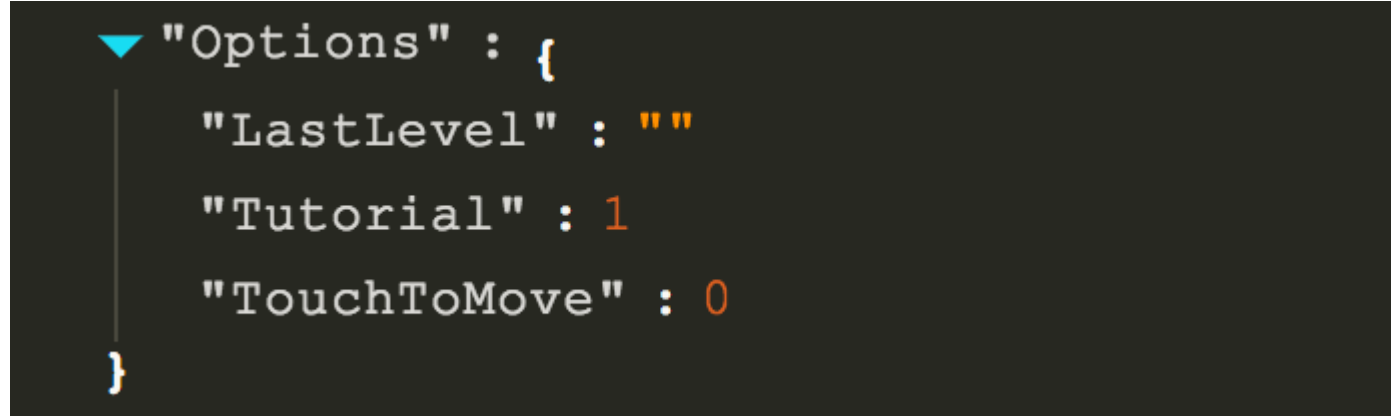
Variables will be shown in a list showing variable name and values (either a number or a text). Here is an example of an instance of an object having a “Life” variable:



Like other fields, you can **click on the value of the variable** to change it. It's a good way to see how your game is behaving according to the value of variables and test that your events are working properly.

Variables can also be **structures**, meaning that they hold other variables. In this case, instead of seeing the value (number or text) of the variable, you will see a sub-list containing the children variables:

Here is an example of a variable "Options", containing children variables to store the options of a game:



```
▼ "Options" : {  
  "LastLevel" : ""  
  "Tutorial" : 1  
  "TouchToMove" : 0  
}
```

You can also change the value of the children variables like any other.

Profile the performance of your game.

A profiler is a tool included in the Debugger that measures the time spent in various part of the engine while your game is running. It then shows a breakdown of each section, including events. [Read more about using the profiler here!](#)