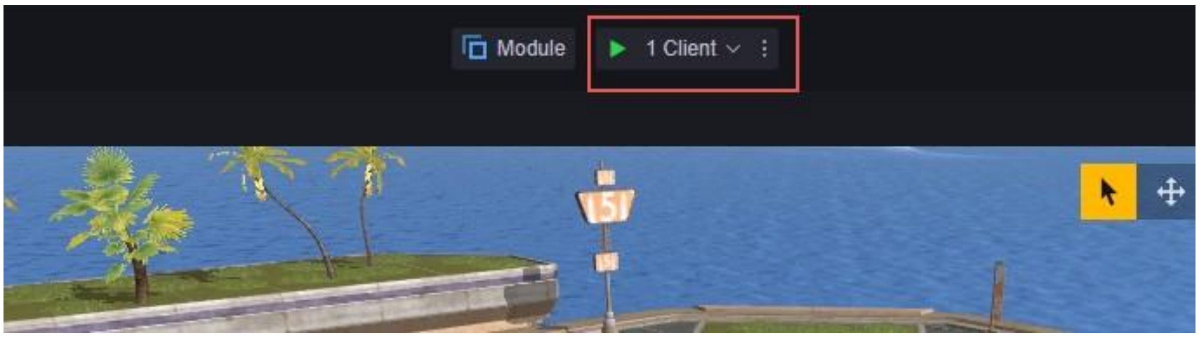
Debug - Debugging

Debugging is a way to visually view the content of a game and confirm that it is behaving as expected. Using debugging opens up at least one game process and allows you to see if the game is running as expected.

Automatically saves the entire project when using debugging.

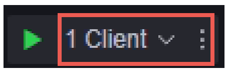
debugging portal

Debugging is available at the top of the project editing screen.

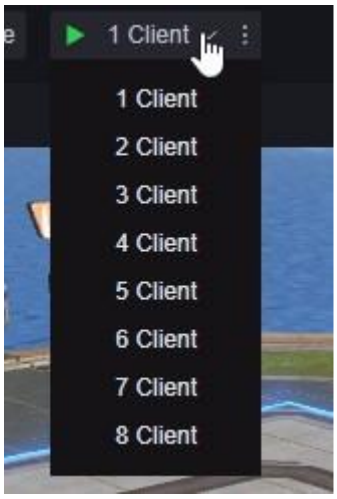


set up

Click the button on the right to set up debugging.



Multi-client debugging

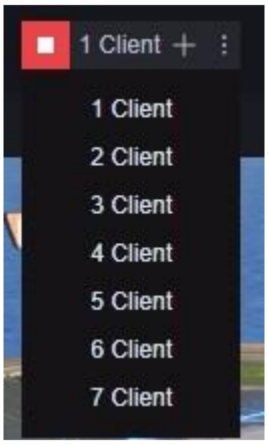


By clicking on the Clients area, you can choose how many clients to start this debugging with. Multiple clients are useful for debugging interactions between different players in multiplayer games.

Supports up to eight clients to start debugging at the same time.

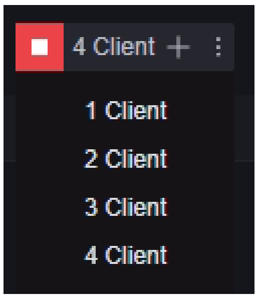
> Multi-client debugging can put a performance strain on your device

If debugging has already started, the button here will change to Add New Client.



Add up **to** eight clients for debugging at the same time. After adding new clients, the topmost client will be updated to show the number of currently open clients, and the number of clients that can be added will be reduced accordingly.

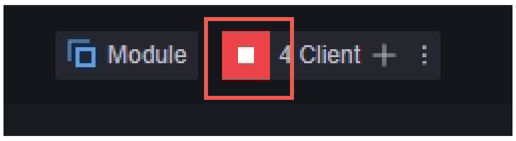
Adding three clients as in the picture above will change it to:



Newly joined clients are considered mid-join players. This may affect the validation of your design with respect to mid-join players.

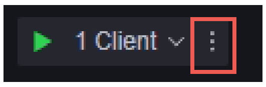
Closing a debug window is considered to correspond to a player quitting in the middle. The current number of clients and the number of clients that can be added will also change. When the last debugging window is closed, the debugging will end.

By clicking the Stop button on the Project Editor, all open debugging windows will close and this debugging will end.

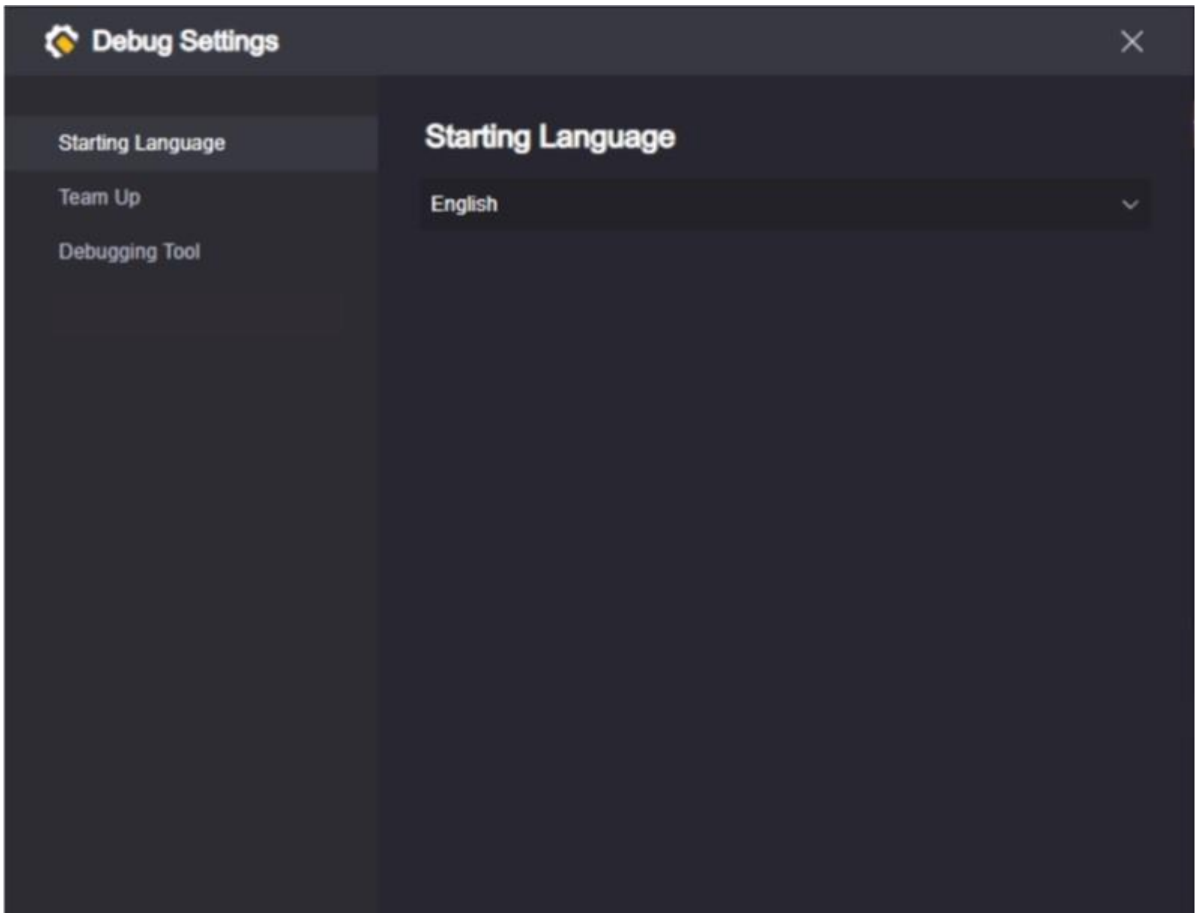


Debugging Settings

The debug menu can be opened by clicking the menu button on the far right of debug.



bootstrap language



Starting Language

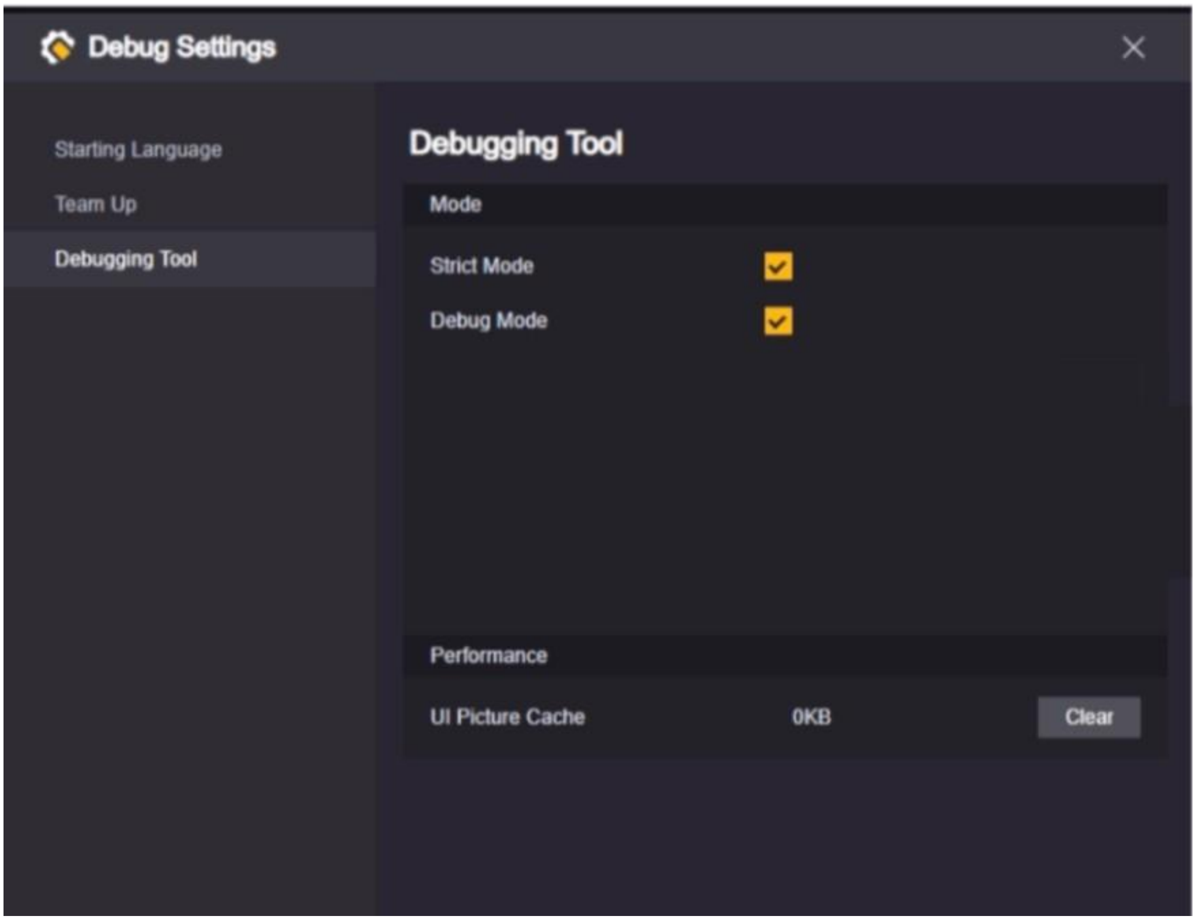
English

The startup language determines the language in which the various text displays in your debug window.

assort

To be sorted out

Debugging Tools



Debugging Tool

Mode

Strict Mode ✓

Debug Mode ✓

Performance

UI Picture Cache 0KB

strict model

In strict mode, debug mode is exited as soon as an error occurs in the process. The problem can be pinpointed more accurately.

Debug Mode

Breakpoints only work in Debug mode.

Breakpoints are described below.

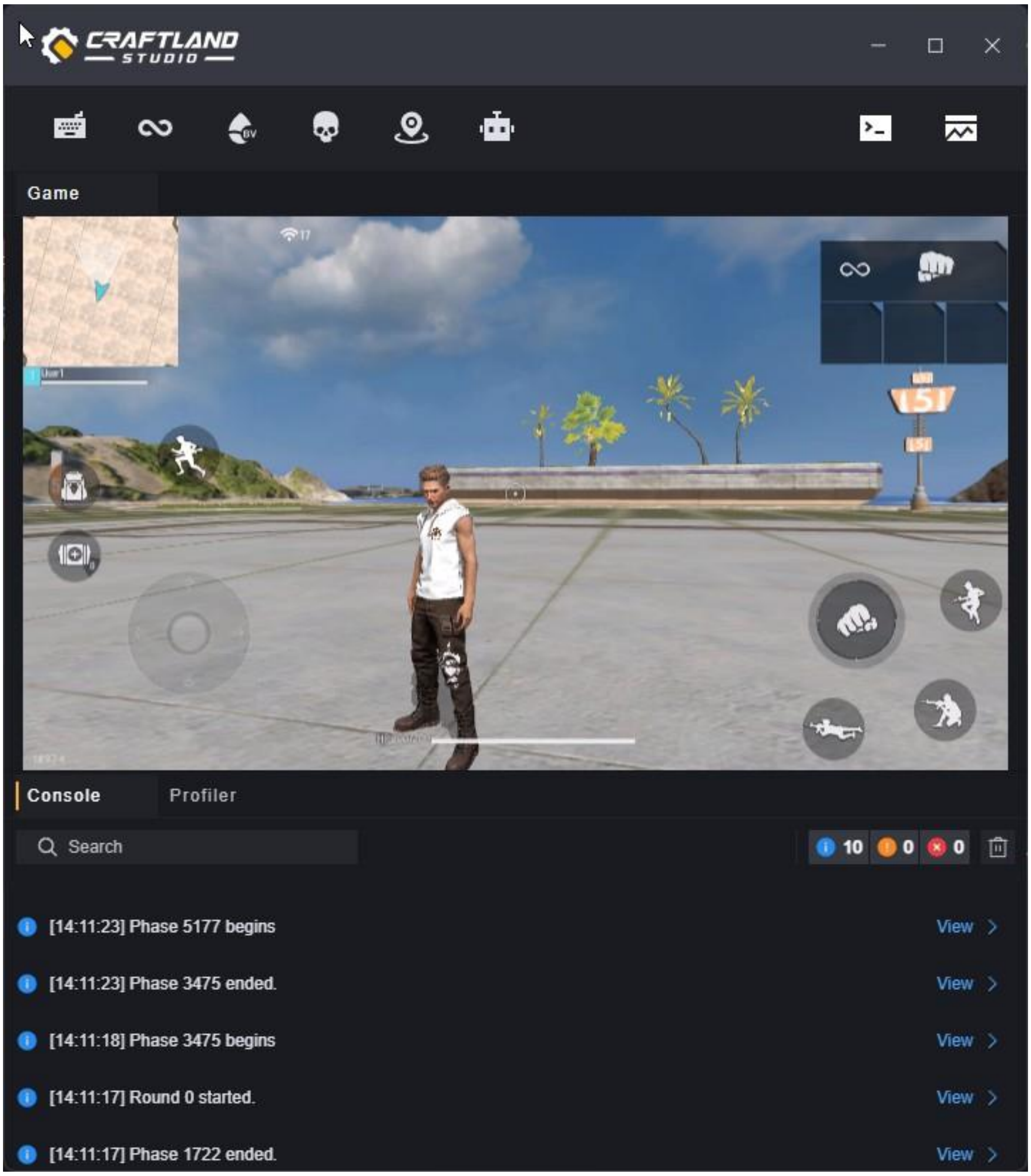
performances

Shows how much local storage space is used by the current UI image cache. You can free up space by clearing the cache, but the UI image has to be reloaded the next time you use it.

It is recommended to clear it when you have more projects open and have a lot of content in your cache that you will not use again.

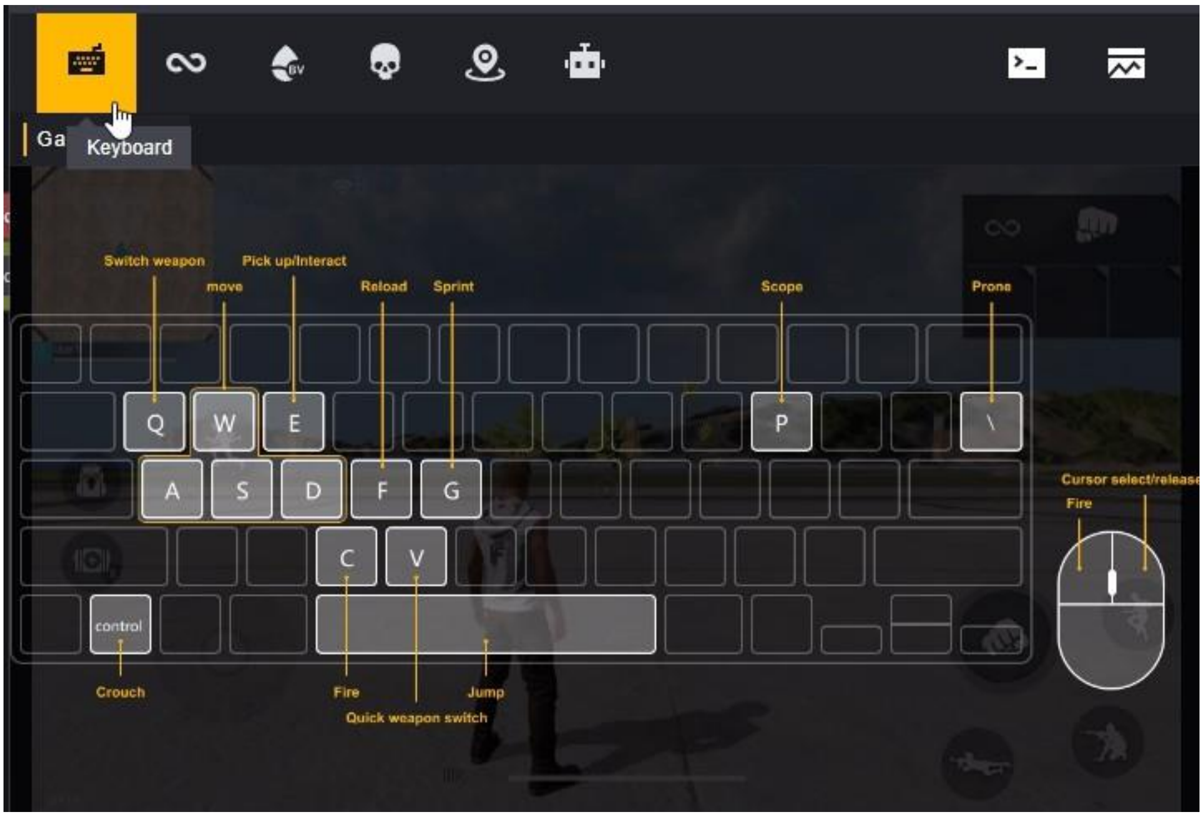
debug window

Game window



```
[14:11:23] Phase 5177 begins          View >
[14:11:23] Phase 3475 ended.          View >
[14:11:18] Phase 3475 begins          View >
[14:11:17] Round 0 started.           View >
[14:11:17] Phase 1722 ended.          View >
```

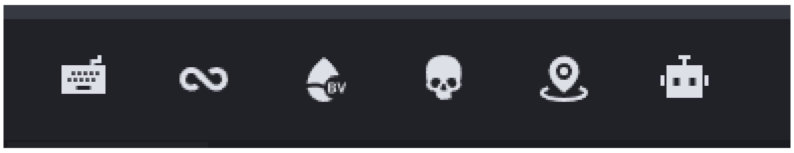
In the debug window, you can simulate a trial play of the map you have made. By default, your mouse is free to move, and you can simulate the actions you would perform on a mobile phone screen by tapping or long-pressing the left mouse button (this can be a bit clumsy). By pressing the right mouse button in the screen area, you can enter the emulator operation mode, at which time your mouse movement will operate the viewpoint rotation, the left mouse button will change to the attack command, and you can use the keyboard to perform the operation at the same time. Keyboard shortcuts can be found in the "Key Tips" screen on the top left:



Click the right mouse button again to exit the emulator operating mode.

menu

We have provided some default instructions and GM functions available for you to use:

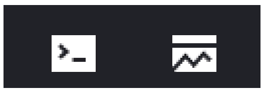


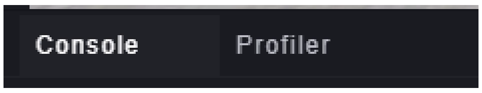
Key Hints: Shortcut key hints for emulator operation.
Invincible: switch invincibility, invincible players do not take damage and the player will turn gold as a hint that the status is on. Note that this command toggles invincibility, when the player is in invincibility, use this command to cancel invincibility.
Restore Blood: directly sets the player's blood level to the upper limit.
Not valid for dead players.
Suicide: kills the current player.
Teleport to Birth Point: teleports the player to the location where he or she was born. If you have configured a birth point on the scenario that is currently available to the player, the player will be born at the birth point by default. If there is no birth point configured or the configured birth point is not currently available to the player (e.g., the birth point requires a team that does not match the player), the player will be born near (0,0,0), and the y-axis coordinates may be adjusted slightly.

Add bot: Add a bot to the current player's team, this command cannot be executed if the current player's team is full.

Console and Performance Monitoring

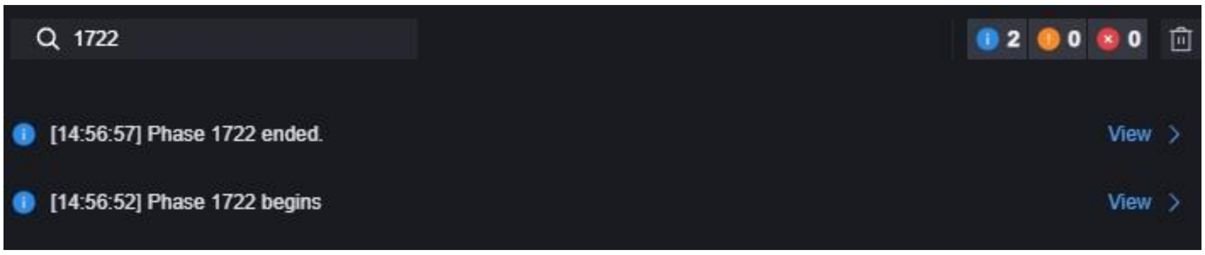
You can toggle the console or performance monitoring display using the buttons in both the upper right and lower centre:





Console:

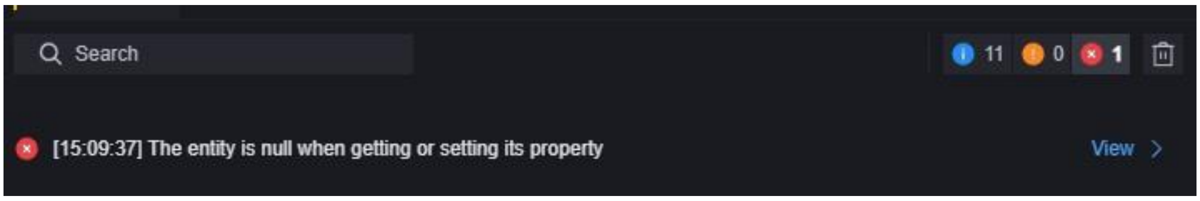
You can see server logs, warnings and error reports here. Searching for displayed messages is supported:



[14:56:57] Phase 1722 ended. View

[14:56:52] Phase 1722 begins View

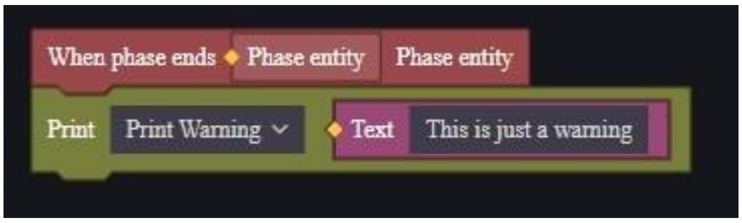
Click the button on the right to filter the type of information displayed:

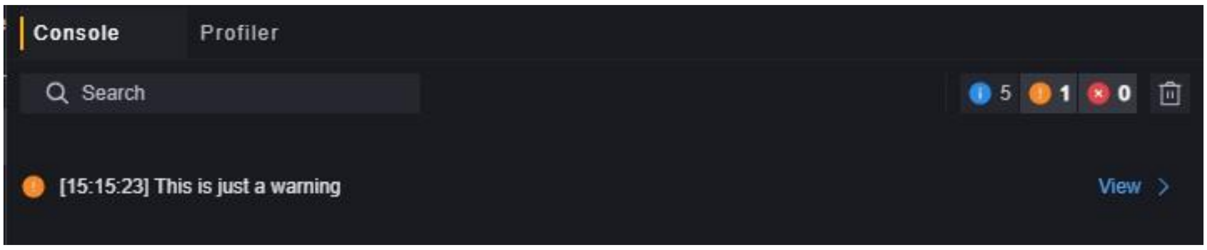


[15:09:37] The entity is null when getting or setting its property View

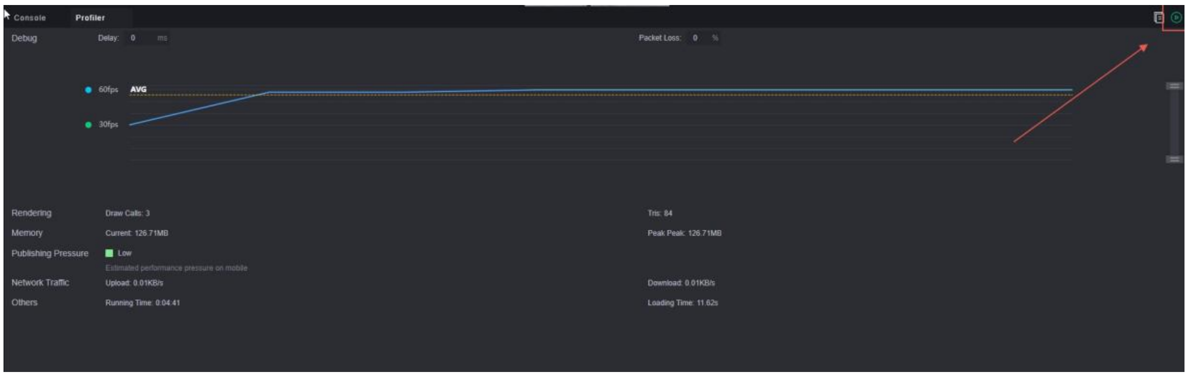
> Error message only
>
> Searching changes the number of items displayed, while filtering does not. The interaction between the two may result in you not being able to find the information you are looking for.

Use the Print node in your script to output the information you want in the corresponding category.





[15:15:23] This is just a warning View



Debug Delay: 0 ms Packet Loss: 0 %

Rendering Draw Calls: 3 Tris: 84

Memory Current: 126.71MB Peak Peak: 126.71MB

Publishing Pressure 🟩 Low
Estimated performance pressure on mobile

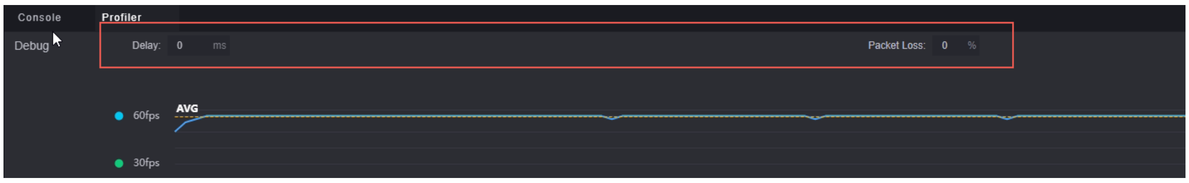
Network Traffic Upload: 0.01KB/s Download: 0.01KB/s

Others Running Time: 0:04:41 Loading Time: 11.62s

Performance Monitoring:

Click the button in the upper right corner to start the performance test.

After opening the test, various performance data will be displayed in real time, which is convenient for you to monitor the resource consumption of editing maps. You can also artificially adjust the latency and packet loss rate of the game to test the performance of the

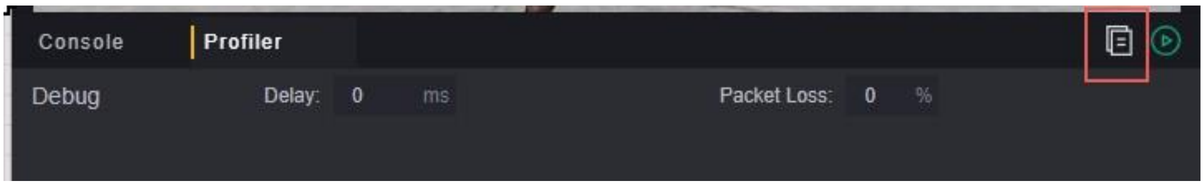


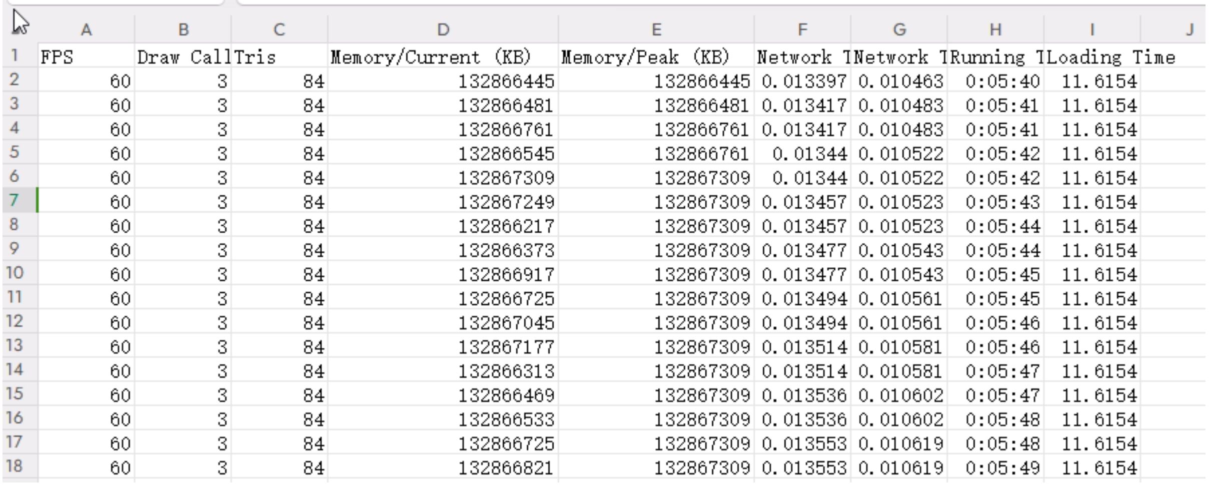
Debug Delay: 0 ms Packet Loss: 0 %

game in a weak network environment.

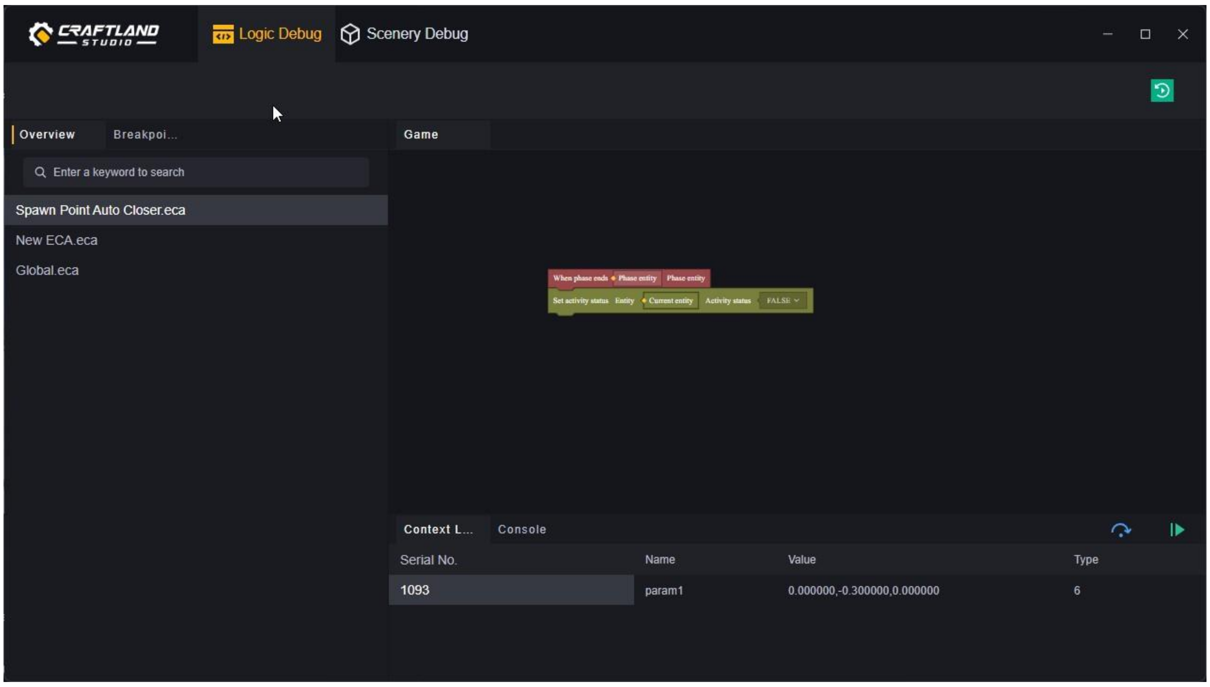
> The latency and packet loss entered here are additional values, the actual game latency and packet loss are equal to your original latency and packet loss plus the values entered here.

Supports debug log export, clicking the log button next to the test will open the exported CSV file locally to you.





	A	B	C	D	E	F	G	H	I	J
1	FPS	Draw Call	Tris	Memory/Current (KB)	Memory/Peak (KB)	Network T	Network T	Running T	Loading Time	
2	60	3	84	132866445	132866445	0.013397	0.010463	0:05:40	11.6154	
3	60	3	84	132866481	132866481	0.013417	0.010483	0:05:41	11.6154	
4	60	3	84	132866761	132866761	0.013417	0.010483	0:05:41	11.6154	
5	60	3	84	132866545	132866761	0.01344	0.010522	0:05:42	11.6154	
6	60	3	84	132867309	132867309	0.01344	0.010522	0:05:42	11.6154	
7	60	3	84	132867249	132867309	0.013457	0.010523	0:05:43	11.6154	
8	60	3	84	132866217	132867309	0.013457	0.010523	0:05:44	11.6154	
9	60	3	84	132866373	132867309	0.013477	0.010543	0:05:44	11.6154	
10	60	3	84	132866917	132867309	0.013477	0.010543	0:05:45	11.6154	
11	60	3	84	132866725	132867309	0.013494	0.010561	0:05:45	11.6154	
12	60	3	84	132867045	132867309	0.013494	0.010561	0:05:46	11.6154	
13	60	3	84	132867177	132867309	0.013514	0.010581	0:05:46	11.6154	
14	60	3	84	132866313	132867309	0.013514	0.010581	0:05:47	11.6154	
15	60	3	84	132866469	132867309	0.013536	0.010602	0:05:47	11.6154	
16	60	3	84	132866533	132867309	0.013536	0.010602	0:05:48	11.6154	
17	60	3	84	132866725	132867309	0.013553	0.010619	0:05:48	11.6154	
18	60	3	84	132866821	132867309	0.013553	0.010619	0:05:49	11.6154	



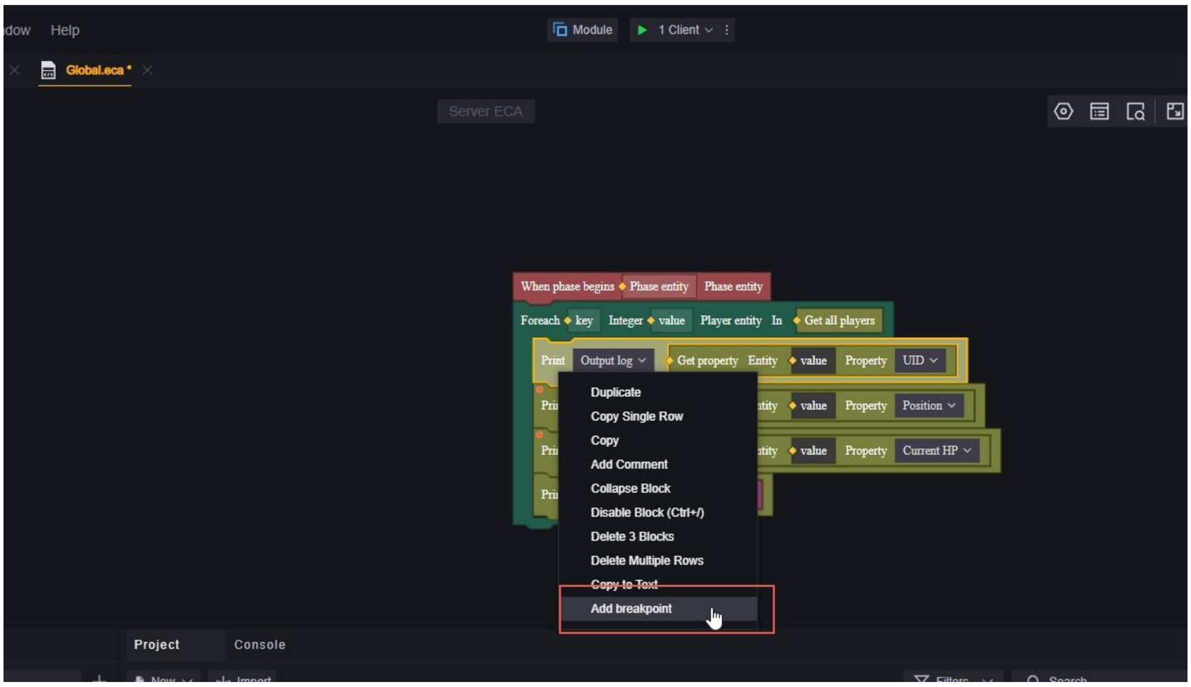
Serial No.	Name	Value	Type
1093	param1	0.000000,-0.300000,0.000000	6

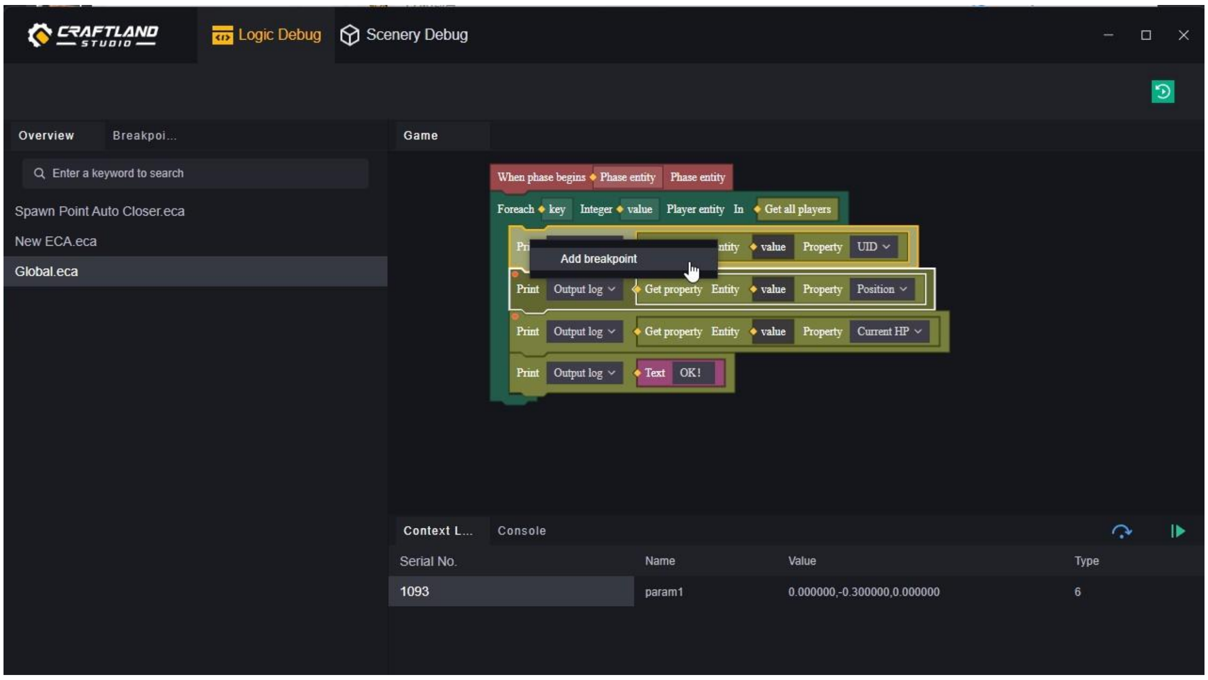
Logic and Scenario Debugging Window

After you start debugging, a logic and scenario debugging window will be opened at the same time as the debugging window. It is minimised by default:

Logic Debugging

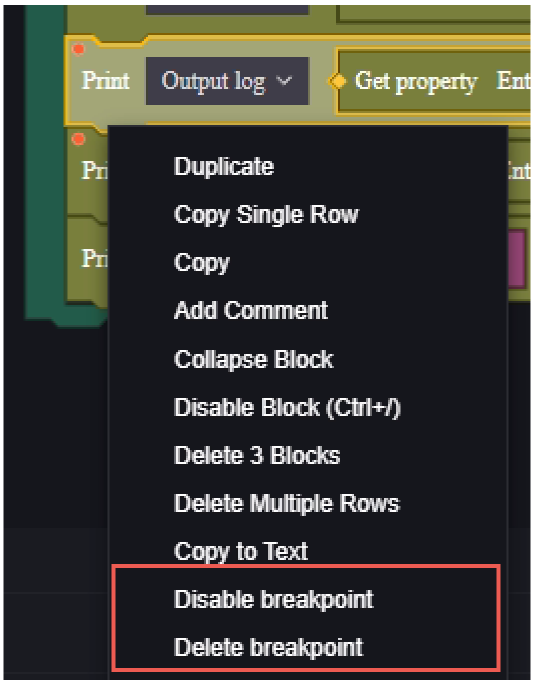
You can monitor script sessions you care about by means of breakpoints. By adding breakpoints to nodes within the script editor or at logical debugging points, you can pause your game process at a breakpoint and output the value of the node's variable to the context list below.



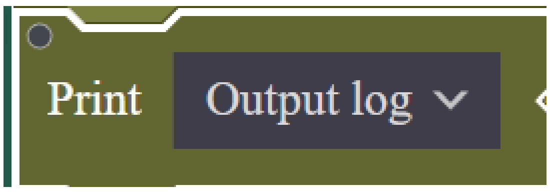


Serial No.	Name	Value	Type
1093	param1	0.000000,-0.300000,0.000000	6

For nodes with breakpoints already added, you can close the breakpoints or remove them via the context menu.

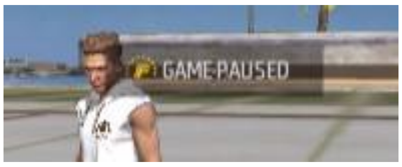


Closing a breakpoint means that this breakpoint will not take effect, but you will still see a grey breakpoint flag on the node.

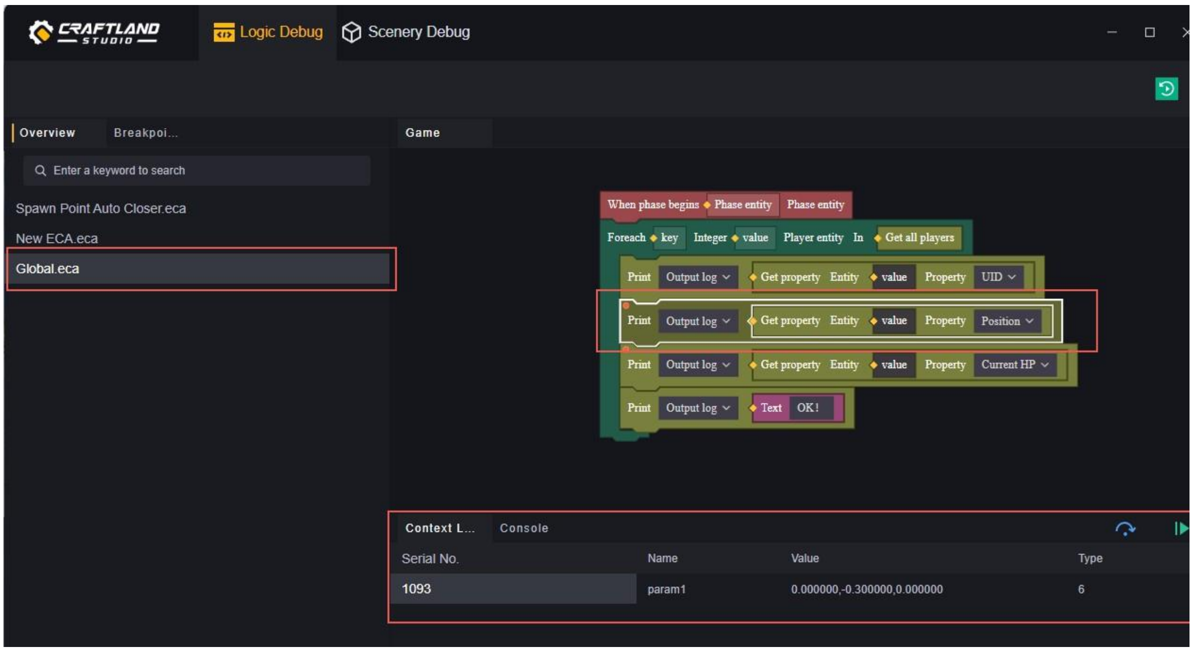


Edits made to the script during the debugging run will not be reflected in the game immediately, and you will need to restart the debugging process for the new changes to be applied.
The editing of breakpoints in logical debugging also does not affect the script, and the operation of breakpoints in logical debugging is to check what needs to be confirmed temporarily in the current debugging.

When a breakpoint is in effect, the game goes into pause.



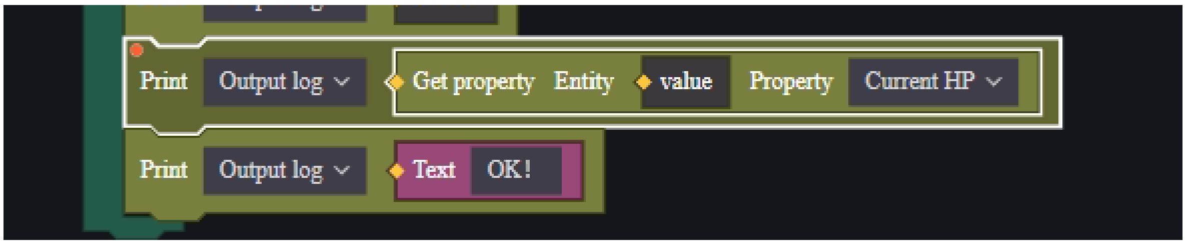
The Logic Debug window defaults to showing the script where the current breakpoint is located and highlights the variables.

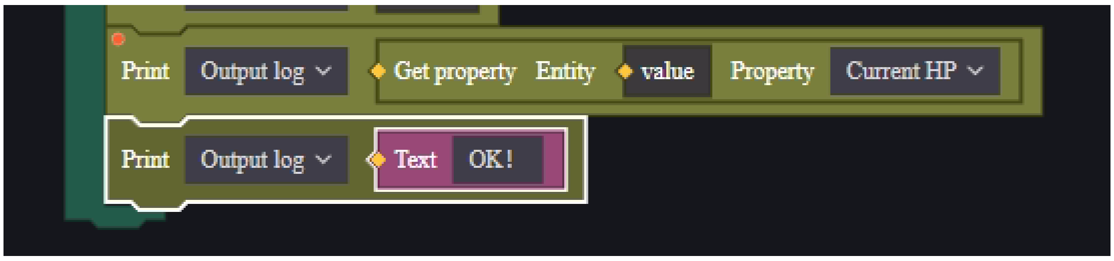


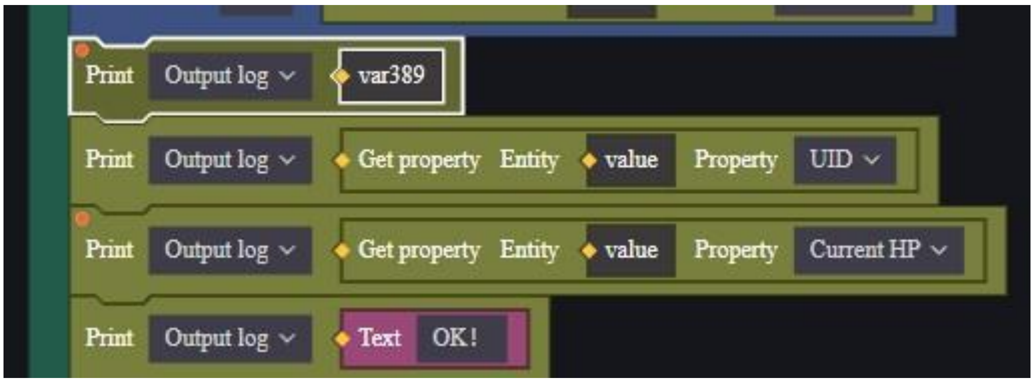
```
When phase begins ◆ Phase entity  Phase entity
Foreach ◆ key  Integer ◆ value  Player entity  In ◆ Get all players
    Print  Output log ◆ Get property  Entity ◆ value  Property  UID
    Print  Output log ◆ Get property  Entity ◆ value  Property  Position
    Print  Output log ◆ Get property  Entity ◆ value  Property  Current HP
    Print  Output log ◆ Text  OK!
```

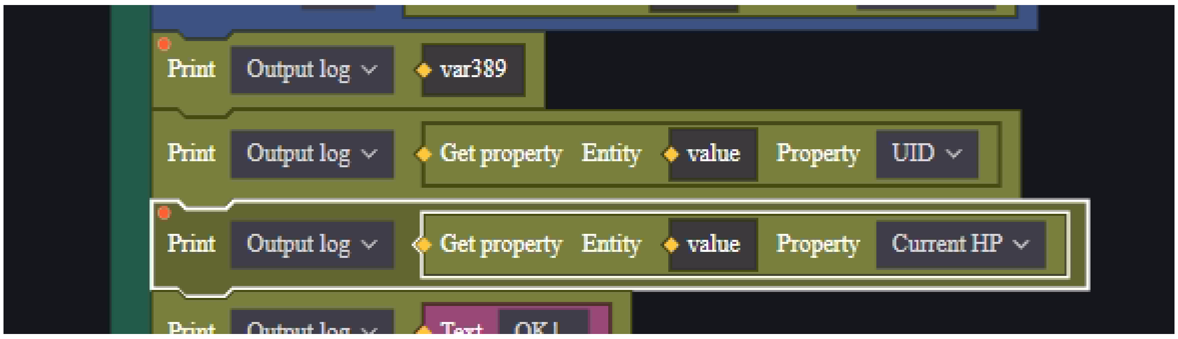
Serial No.	Name	Value	Type
1093	param1	0.000000,-0.300000,0.000000	6

By clicking on the step, the game will run one node down, in nodes.


```
Print  Output log ⌄  Get property  Entity  value  Property  Current HP ⌄
Print  Output log ⌄  Text  OK!
```

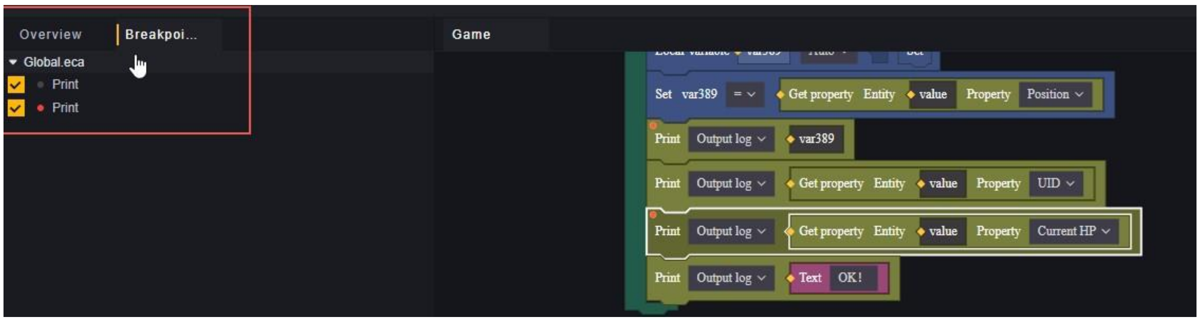

```
Print  Output log ⌄  Get property  Entity  value  Property  Current HP ⌄
Print  Output log ⌄  Text  OK!
```


```
Print  Output log ⌄  var389
Print  Output log ⌄  Get property  Entity  value  Property  UID ⌄
Print  Output log ⌄  Get property  Entity  value  Property  Current HP ⌄
Print  Output log ⌄  Text  OK!
```


```
Print  Output log ⌄  var389
Print  Output log ⌄  Get property  Entity  value  Property  UID ⌄
Print  Output log ⌄  Get property  Entity  value  Property  Current HP ⌄
Print  Output log ⌄  Text  OK!
```

Click Continue and the game will run to the next breakpoint. If there is no next breakpoint it will follow the normal game progression.

You can also browse all breakpoints at the list of breakpoints, which are arranged in a pattern of script name - node name where the breakpoint is located.

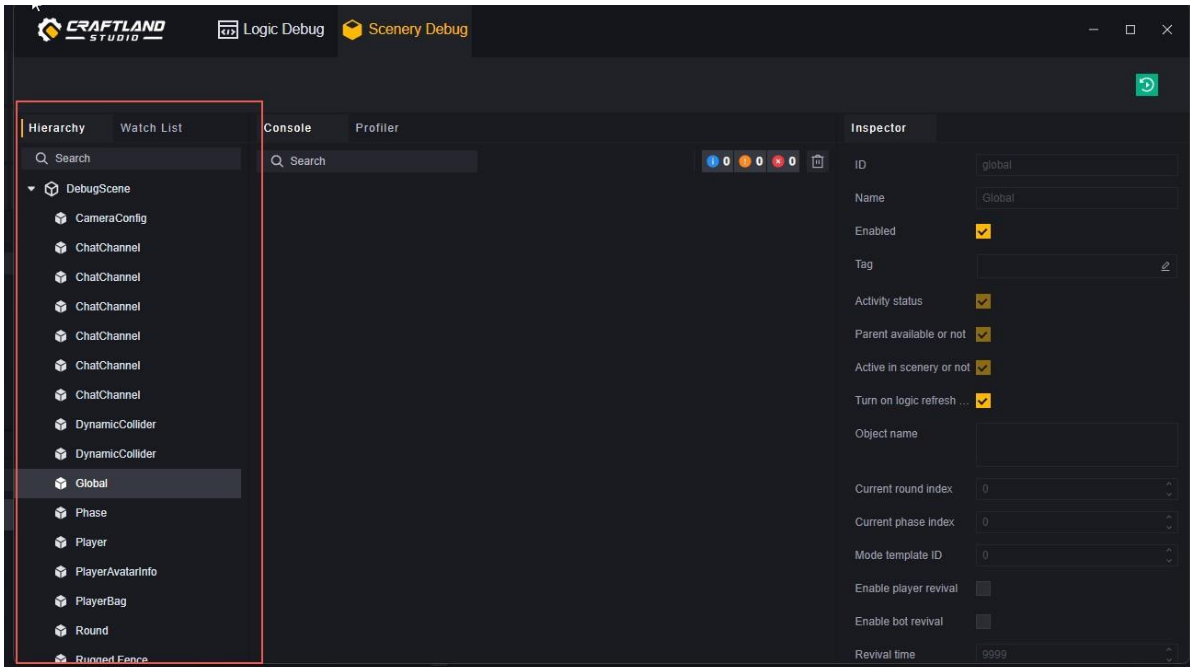


```
Set  var389  =  Get property  Entity  value  Property  Position
Print  Output log  var389
Print  Output log  Get property  Entity  value  Property  UID
Print  Output log  Get property  Entity  value  Property  Current HP
Print  Output log  Text  OK!
```

You can quickly change the enable status of a breakpoint by ticking the box in front of the breakpoint (only for current debugging). The dots in front of the breakpoint: red means the breakpoint is currently stopped, grey means the breakpoint is not triggered.

Scene Debugging

In Scene Debugging, you can view a list of entities in almost any game.



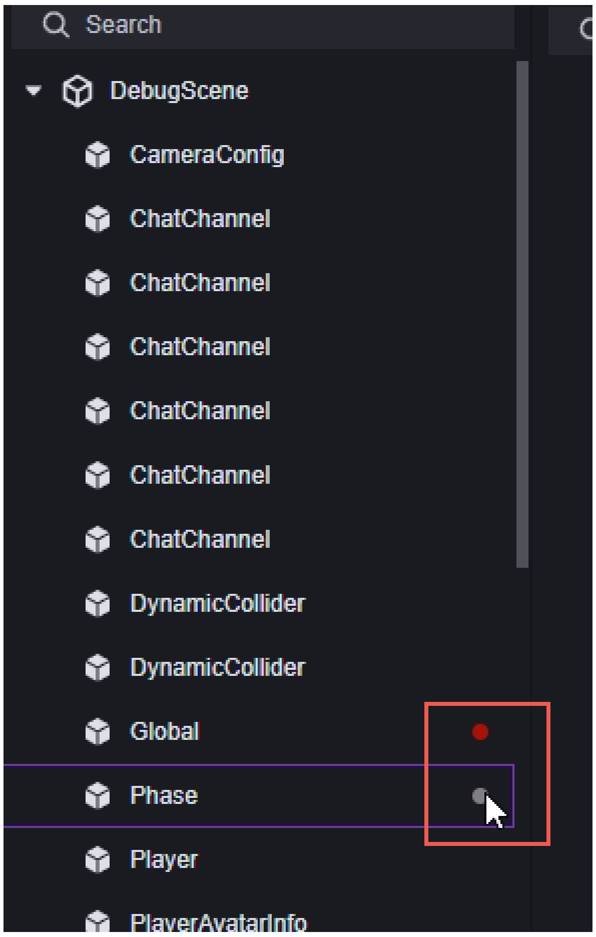
DebugScene
CameraConfig
ChatChannel
ChatChannel
ChatChannel
ChatChannel
ChatChannel
ChatChannel
DynamicCollider
DynamicCollider
Global
Phase
Player
PlayerAvatarInfo
PlayerBag
Round
Rugged Fence

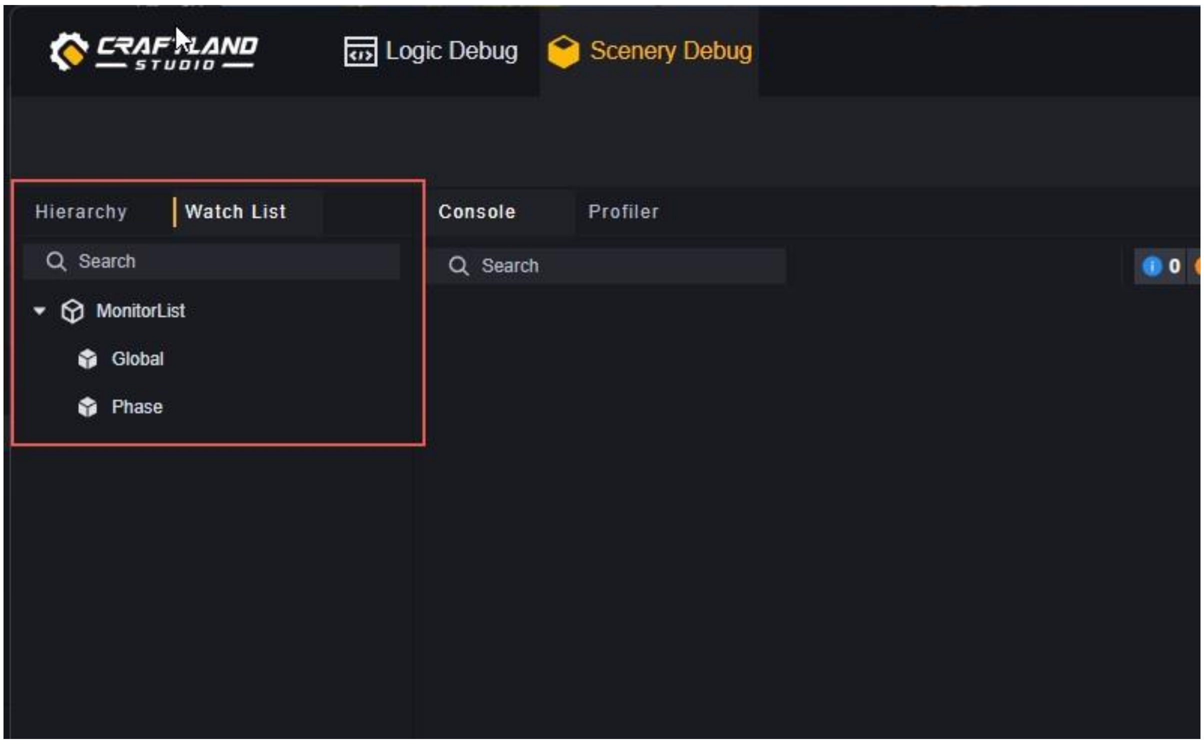
ID	global
Name	Global
Enabled	✓
Tag	
Activity status	✓
Parent available or not	✓
Active in scenery or not	✓
Turn on logic refresh ...	✓
Object name	
Current round index	0
Current phase index	0
Mode template ID	0
Enable player revival	
Enable bot revival	
Revival time	9999

The Scene Debugging Hierarchy panel displays more than just the entities that you customise in your scene; abstract entities (e.g. global, rounds) and hidden entities are also displayed here.

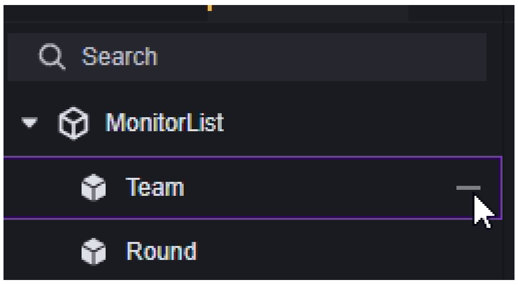
You can modify the properties of an entity in real time through the right-hand view panel and see the corresponding performance in the debug window. This helps you to confirm the performance of some designs in real time.

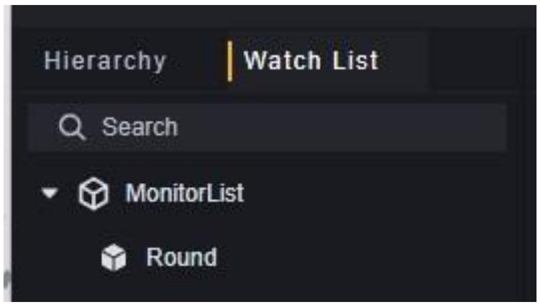
Entities can be added to the watch list by clicking the dot to the right of the entity in the left hierarchy panel.

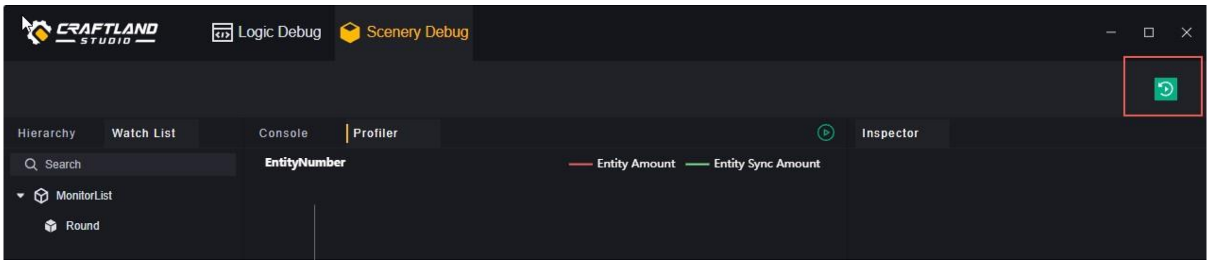




Entities added to the watch list can be easily found in the watch list. You can stop monitoring this entity by clicking the minus sign in the watch list.



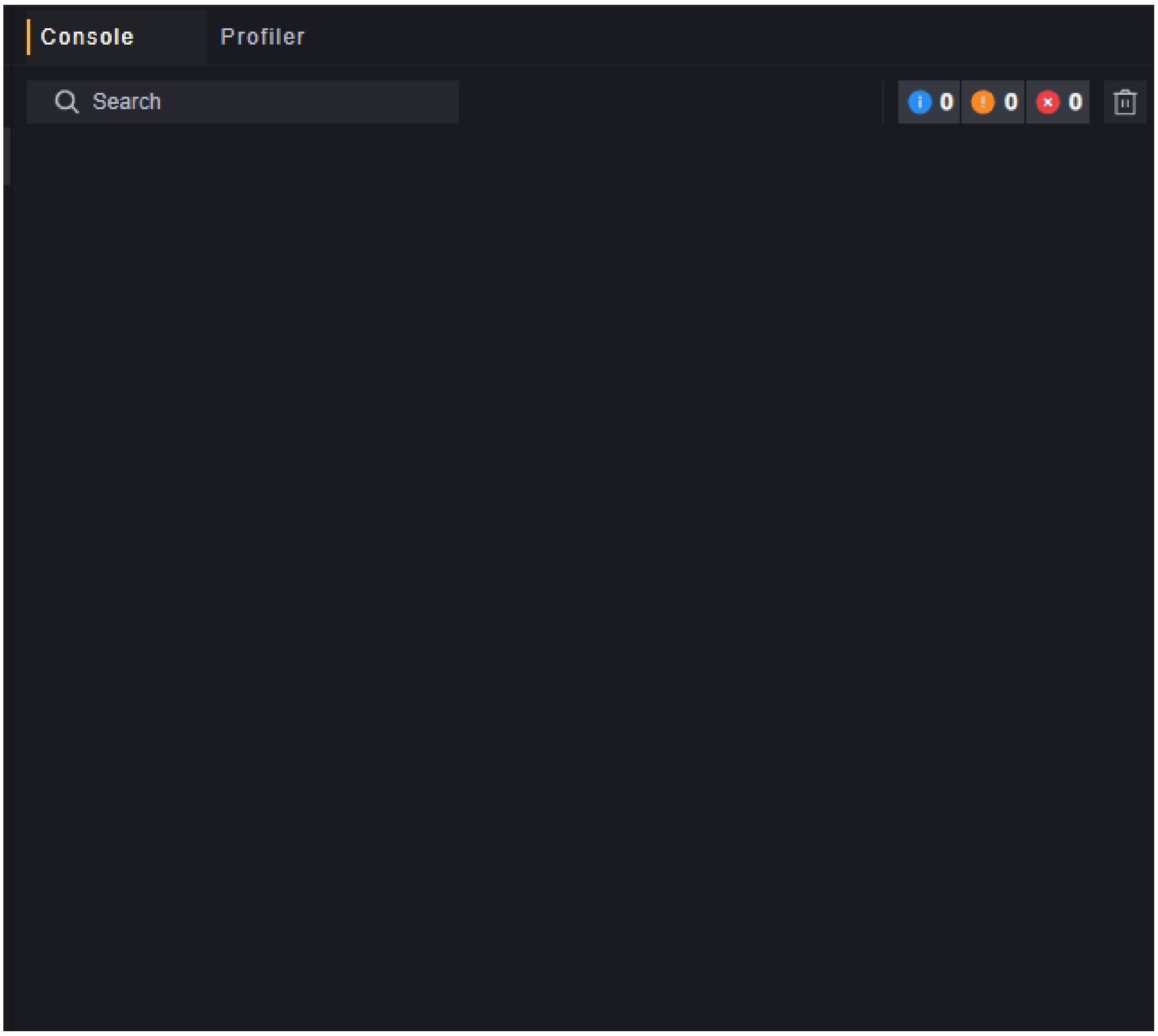




The watchlist only works for the current debug.

The game will restart by clicking the restart button in the upper right corner of the scene debugging. This is equivalent to closing debugging and then starting it again. This means that everything that acted on the current debug will be refreshed.

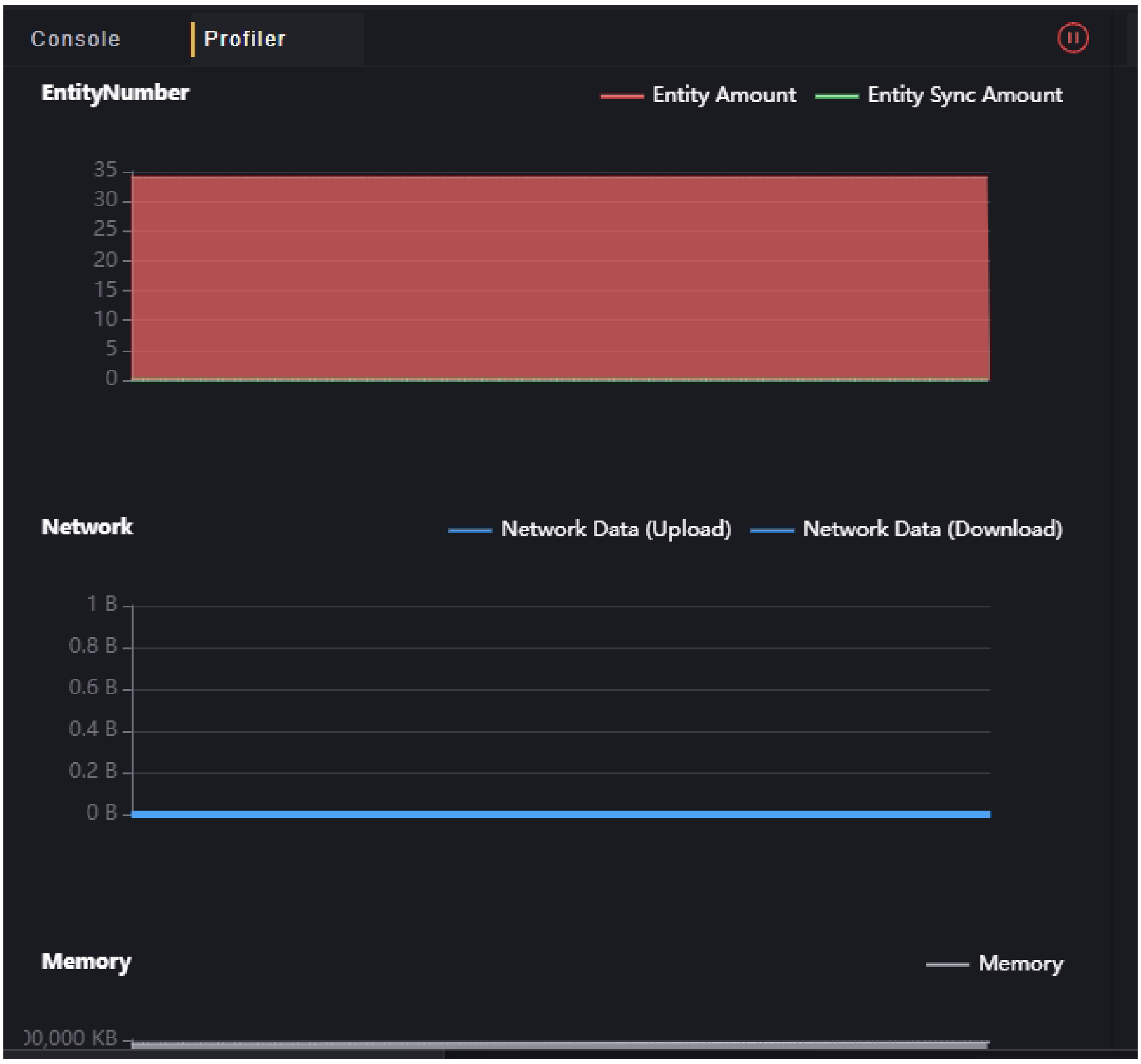
Console and performance:



Console and performance monitoring interface similar to the debug window.

Console and performance monitoring in scenario debugging is primarily for client-side information.

Start monitoring in Performance Monitoring allows you to observe the number of entities, network data throughput and memory usage in real time.


Console
Profiler
EntityNumber
Entity Amount
Entity Sync Amount
35
30
25
20
15
10
5
0
Network
Network Data (Upload)
Network Data (Download)
1 B
0.8 B
0.6 B
0.4 B
0.2 B
0 B
Memory
Memory
)0,000 KB