

Emergency Exit Documentation

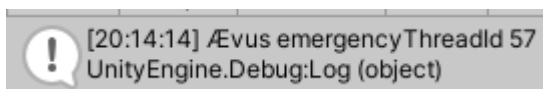
Hotkeys	1
Changing Hotkeys	2
Testing The Extension	2
Warning	2

Hotkeys

There are 3 Hotkeys used:

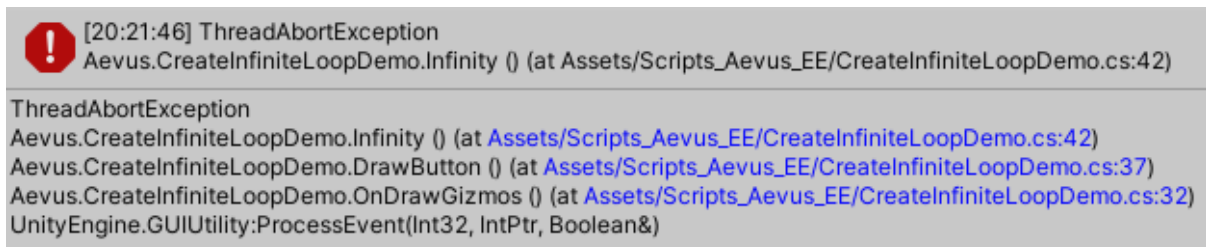
CTRL + SHIFT + H:

Makes the Emergency Exit thread log to the console. This is mainly used to check if this extension works correctly. It logs the thread Id of the Emergency Exit thread and looks like this:



CTRL + SHIFT + Q:

Throws the exception into the main thread exactly once, no matter how long the button combo is held down. Use this one to throw exceptions with precision. It is recommended to use this option over the next one for more safety.




CTRL + SHIFT + E:

Throws the exception about 10 times a second while these buttons are held down. Throws the same exception as the option above. It is recommended to use the option above, but depending on the situation this might be more useful if you are stuck in a loop of infinitely repeating infinite loops. Just be careful that you stop holding this once you stop the cycle of infinite loops, so that you don't accidentally throw an exception when the main thread is doing other things.

Changing Hotkeys

This script can't use unity's input system, because that freezes once you enter an infinite loop. You need to open the EmergencyExit.cs script and go to the bottom of it. Then change the hex number inside the GetAsyncKeyState function into the keys that you want. You can see the keycodes [here](#). (Note: The function keys don't work well, sometimes they get "stuck" and report being pressed down, even when they aren't)

Testing The Extension

1. To see if everything is running, press CTRL + SHIFT + H, which makes the Emergency Exit thread log to the console. The script is active and running if that succeeds.
2. Open the included demo scene
3. Click on the cube and check the inspector for the CreateInfiniteLoopDemo component 
4. Here you can play around with different infinite loops you might encounter during development. It's useful to learn how to deal with them and how to navigate the unity inspector if you are stuck in a loop of infinite loops.
 - 4.1. The first boolean activates an infinite loop that gets run inside of update, so you need to enter playmode and then check it. This one is easy to deal with because you can just press CTRL + SHIFT + Q to break out of it, which enters pause mode. Now you can do whatever to fix the issue, which in this case is to uncheck the bool or deactivate the gameobject or do whatever comes to mind. now you can resume playmode without issues.
 - 4.2. The second boolean creates an infinite loop inside of a GUI function. This one is much trickier to deal with, because this function runs multiple times a frame and runs inside of edit and pause mode too. It's useful to learn how to navigate the unity inspector when this happens. You can do one action with the mouse, like clicking on something, and then press CTRL + SHIFT + Q until it updates. This way, you can navigate the unity editor with repeated presses of the hotkeys. try to navigate around, select different objects in the hierarchy and then go back to the cube and uncheck the bool.

Warning

Do not throw an exception if you are currently not in an infinite loop. This exception can reach the main thread at any time, which might cause problems, especially if the main thread is currently doing things inside the unity internals. In my testing nothing bad has happened, unity doesn't crash, it's mainly Job buffers not getting properly cleared. But still, bad things could happen. Worst case would be that you interrupt unity saving something and then end up with corrupted data.