hetrodo.Utilities.MainThread

This is a wrapper class for executing heavy calculations without freezing the main thread but still using unity's api (like transform.position, Instantiate, Destroy, etc).

Reference

MainThread

- Exec(System.Action action) -> (method)(synchronous) Executes an action on the main thread, and freezes its caller thread until execution.

- ExecAsync(System.Action action) -> (method)(asynchronous) Executes an action on the main thread without freezing.

- IsRunning -> (field) Tells if you had already initialized the MainThread class.

- OnExceptionCaught(Exception ex) -> (event) If any exception occurs while executing the actions, this event will be fired.

MainThread.Timing

- DeltaTime -> (field) Time since last call for each thread.

Tips

Try to minimize MainThread.Exec usage, as every call will add at least 25ms to your code.
If you are using the MainThread.ExecAsync in a loop be sure to add a Thread.Sleep to not overflow the execution pool.

Syntax

```
void Start()
{
    //Initializing MainThread class
    _ = new MainThread();

    new Thread(() =>
    {
        MainThread.Exec(() => { print("Executed on unity"); });
        MainThread.Exec(Boo);
        print("Executed outside unity");
    }).Start();
}


void Boo()
{
    print("Executed on unity");
}
```