Introduction

Last update : February 2023

As many of you may already know, Udon can be pretty slow, but I was curious to know what really slows it down, so I made a world called "Udon Benchmark" to benchmark it, the world can be found here https://vrchat.com/home/world/wrld_174475ad-6f8e-444d-8b02-67cd13e13b74
In that world I benchmarked :

- The execution time between C# and U#
- As many C# features I could possibly think off, for instance for-loops, function calls, recursive functions etc. and compare their execution times.

Each script got executed 50 times, and the results got averaged.

All results I'll share are the results I got from my world, you'll probably get different results on your hardware.

Since VRChat keeps improving Udon, many results I'll share will probably be out of date in the future, and I cannot promise that I'll keep those values up-to-date.

The values bellow shows a summary of all execution times, I'll explain each line in the next chapter, if you want you can paste those values into a .CSV file, or generate your own .CSV file in my "Udon Benchmark" world that features a CSV exporter.

- B1 : execution time of the Benchmark1 method
- B2 : execution time of the Benchmark2 method
- SD : Standard deviation, which shows the amount of variation or dispersion of a set of values https://en.wikipedia.org/wiki/Standard_deviation

Version 1.1 - Number o iterations : 50	B1 (ms)	B1 min (ms)	B1 max (ms)	B1 SD	B2 (ms)	B2 min (ms)	B2 max (ms)	B2 SD	B1/B2
C# vs U# (Part 1)	684.576946	661.567	804.8891	21.4634273630957					

C# vs U# (Part 2)	972.71161	928.6402	1119.3484	35.245644913913 8					
For-loop overhead test	79.903544	76.9296	94.951	3.70946646218348	35.388596	33.3341	53.8828	3.81011410065158	2.25788963201592
Recursive vs iterative	532.552632	519.5953	577.7497	12.9280788058929	79.859174	77.0833	93.3914	3.69170508577324	6.6686468858293
Builtin functions vs calculating something manually (part 1)	166.699154	161.6328	189.9542	5.54937905157721	92.678434	88.3964	125.4238	6.55232813505581	1.79868332691077
Builtin functions vs calculating something manually (part 2)	60.58319	57.8675	78.6651	3.79625209517229	63.034632	60.8127	76.2538	2.8186812640978	0.961109600830223
Function overhead test	57.03638	54.4904	65.0939	2.92755877119487	66.20722	63.0205	81.5224	3.63724205056524	0.861482780880997
GetComponent<>()	60.0576	58.0447	69.7074	2.32634479559673	111.367394	106.5221	122.6288	3.81778258895972	0.539274538470389
Calling methods from a separate script	236.120418	228.2506	261.7778	7.89126462852919	305.513994	297.6651	333.578	7.45977174212214	0.772862856161017
Caching Networking.LocalPlayer	15.097448	14.5741	18.5321	0.783616316634614	18.251728	17.7745	26.3932	1.23896447132918	0.827179103260798
The "ref" keyword	14.859624	13.9105	22.6906	1.79010585715594	14.171158	13.4693	18.1117	1.23886350436035	1.04858219772865

Revision #8
Created 14 February 2023 19:16:31 by MyroP
Updated 16 February 2023 20:35:17 by MyroP