23.03.2022

**SWE 585 In Class Practice #2**

**(Testing Unity Game Loop Methods)**

**Name, Surname:** Ceyda Düzgeç

**Group #:** 3

**Group Member Names, Surnames:**

Maria Asif

Mehmet Samet Taştı

Deniz Dikbıyık

Ceyda Düzgeç

**STEP 1:** Provide your system summary below.

|  |  |
| --- | --- |
| System Summary | |
| Operating System | MacOS |
| CPU / GPU | Apple M1 Pro |
| RAM | 32 GB |
| Unity Version | 2020.3.30f1 |

**STEP 2:** Open an empty project in Unity. Find the script ExampleScript.cs among ICP files on Moodle. Run this script in your project. (You can assign the script to some dummy object). Write values for all members

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Observed Values** | **Member 1**  **Ceyda** | **Member 2**  **Maria Asif** | **Member 3**  **Deniz D.** | **Member 4**  **Samet** | **Member 5** |
| Update / sec | 250 | 45 | 230 | 630 |  |
| FixedUpdate/Sec | 50 | 50 | 50 | 50 |  |

Short comment on the results of Step 2 (should be written collaboratively)

|  |
| --- |
| Because we have different hardware capacities and different computational power, we get different results on the update row. The highest result is coming from the more computational power computer.  In the fixed update row, we get the same result because it is fixed for all different hardware capacities. |

**STEP 3:** Create a synthetic computational load inside the update function. (Execute some meaningless computation just for the sake of creating load. The code should not create any audio/visual effect on the game). Make it parametric so that you are able to change the computational intensity as you need.

Write the code for synthetic load in the below area. Indicate in comments how the load is varied.

|  |
| --- |
| **Code:**  **Define the loadModifier:**  [SerializeField] private int loadModifier;  **Write function:**  private void ComputationalLoad(int l)  {  int t = 0;  for (int i = 0; i < l; i++)  {  for (int j = 0; j < l; j++)  {  t += j;  }  }  }  **Call the function in Update:**  ComputationalLoad(loadModifier);  **Then, assign loadModifier value from the editor.**  **Comment:**  The load decreased the result of the update. We needed to make a higher computational load to decrease the update to 20. Here again the higher capacity computer needed to make a high computational load to make the update lower.  The fixed update average is the same here for all computers, for low capacity machines it resulted with negligible changes. |

**STEP 4:** Try to find the parameter value that lowers FPS (i.e. update() frequency) to ~20. This value may be different for all group members.Write down the corresponding parameter in your code for all members.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Member 1**  **Ceyda** | **Member 2**  **Maria** | **Member 3**  **Deniz D.** | **Member 4**  **Samet** | **Member 5** |
| **Synthetic load parameter value** | 10750 | 11000 | 12000 | 16000 |  |
| **Observed FPS** | 19 | 20 | 20 | 18 |  |
| **FixedUpdate() frequency** | 52 | 51 | 52 | 50 |  |

**STEP 5:** Now put your synthetic load in FixedUpdate() as well. (Every member keep the values provided in STEP 4.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Member 1**  **Ceyda** | **Member 2**  **Maria** | **Member 3**  **Deniz D.** | **Member 4**  **Samet** | **Member 5** |
| **Observed FPS** | 3 | 3 | 3 | 6 |  |
| **FixedUpdate() frequency** | 50 | 50 | 50 | 50 |  |

Short comment on the results of Step 5 (should be written collaboratively)

|  |
| --- |
| Fixed update does not change regardless of the computational load added. The fps however decreases with the increase in computational load. Again the higher capacity computer got the higher fps. |

**What to submit:**

This file. (Every member should submit)