Documentation

Required Packages

- Universal Render Pipeline
- TextMeshPro

How To Use

1. Import the package into your game.
2. In the scenes folder there should be a scene called FPS Counter. This is a example scene to show how this works.
3. To add it to your own scene, drag the Graph Camera prefab from the prefabs folder into your scene.
4. Then either drag the canvas prefab into your scene if you don't already have a canvas. If you do drag the Text and FPS Counter prefabs into your scene underneath the canvas gameObject you already have in your scene.
5. Add the script FPS Counter to the canvas if its not already there.
6. On your main camera, not the Graph Camera make sure that in the culling mask that it doesn't render the FPS UI layer. But the Graph Camera has the FPS UI layer selected. Then make sure Graph has FPS UI set as its layer. (If you don't have this layer, you will need to add a layer to the layers list)
7. If all is done correctly it should display a FPS graph on the screen with the FPS text.

Main Scripts

FPSCounter

This is what actually counts the FPS and stores it so it can be used by the other scripts.

pollingTime

This is how often it will update the FPS. The default for this is 1 meaning it will update the FPS every 1 seconds. If make it lower, it will update faster, but the fps counter might fluctuate more. If you make this higher, it will update slower but the FPS counter will be smoother.

_time

This is just a counter for how much time has passed since the last update.

_frameCount

How many frames have passed depending on the pollingTIme.

FPSGraphChanged

This is just a Event Action which is called depending on the pollingTIme and passes through a list of floats which is the history of FPS.

FPSTextChanged

This is just a Event Action which is called depending on the pollingTIme and passes through a float of the current FPS.

fpsHistory

This is a list of floats which stores the FPS history.

lastFrameIndex

This is an index into the frameDeltaTimeArray that indicates which element of the array should be updated with the current frame's delta time.

graphHistory

This stores how many FPS values it will keep in the history, which will be used to calculate the average FPS over the current pollingTIme.

frameDeltaTimeArray

This is an array that holds the delta time for the last graphHistory frames. Each time Update() is called, the current frame's delta time (Time.deltaTime) is stored in the frameDeltaTimeArray at the current lastFrameIndex.

FPSGraphDisplay

FPSLineHistory

This is a reference to a Line Renderer in the game scene.

frequency

This controls how close together the points are on the graph.

amplitude

This controls how high each point is. Higher values cause the graph to be more exaggerated. Smaller values make the points in the graph flatter.

offset

This is a vector2 which controls the offset of the graph. X being left and right and Y being the up and down.

_yOffset

This controls the y position of the graph taking into consideration the y of the offset.

_maxPeek

This is a value of the highest value so far in the graph. Its used to dynamically change the graphs y scale so that the graph scales accordingly.

OnEnable

This just subscribes Draw to the FPSGraphChanged event when this gameObject is enabled in the scene.

OnDisable

This just unsubscribes Draw to the FPSGraphChanged event when this gameObject is disabled in the scene. This is done so that there aren't any memory leaks or unexpected behaviours.

Draw(List<float> graph)

This takes in a List of floats called graph from the FPSGraphChanged event. It calculates the _yOffset of graph and then sets the FPSLineHistory position count to the count of the list thats passed in. Then for every point it updates its position and if the current point is higher then the _maxPeek it sets it to be the current. It then sets the position of the point in the graph based on its index, frequency, and amplitude.

FPSTextDisplay

This is in charge of updating the FPS text with the correct value.

FPSCounterText

This is just a reference to a TextMeshPro text gameObject in the scene.

OnEnable

This just subscribes UpdateText to the FPSTextChanged event when this gameObject is enabled in the scene.

OnDisable

This just unsubscribes UpdateText to the FPSTextChanged event when this gameObject is disabled in the scene. This is done so that there aren't any memory leaks or unexpected behaviours.

UpdateText(float newFPS)

This is a function which takes in a float called newFPS and then sets the text in FPSCounterText to a string which rounds the newFPS and adds it into the string.
