**Profiling**

[Ultimate guide to profiling Unity games](https://resources.unity.com/games/ultimate-guide-to-profiling-unity-games?ungated=true)

**Profiling 101**

* Understanding profliling in Unity
  + Unity’s profiling tools are available in the Editor and via the Pacakge Manager.

**Sample-based vs Instrumentation Profiling**

* + **Sample-based Profiling**
    - when statiscal data about the work that is being done in the application is colledcted and then analyzed.
    - Sample-based profilers probe the call stack evey “n” nano seconds and use call stack information to figure out when functions were called.
  + **Instrumentation-based Profiler**
    - Include **instrumenting** the code by adding profiler markers.
    - this profiler captures a stream of begin and end events for each marker
    - this method doesn’t lose any information
    - Unity Profiler is instrumentation based
    - comes with an associated overhead that can inflate the reported timing data basaed on how many calls are within the captured profiling scopes..
    - one caveat is the Unity API code in question needs to have instrumentation Profiler markers itself

**Increase profiling detail with Profiler markers & Profiler Modules**

* Child samples of managed scripting code that call Unity API can be seen in the Profiler.
* The Unity API code must include instrumentation Profiler markers.
* Most Unity APIs with performance overheads are instrumented, e.g., using **Camera.main** triggers a **FindMainCamera** marker.
* Understanding the meanings of different Profiler markers is helpful when reviewing profile captures.
* Profiler markers increase profiling detail.
* By default, Unity Profiler profiles code timings wrapped in Profiler markers.
* Inserting Profiler Markers into key functions improves profiling detail efficiently.
* Profiler modules capture per-frame performance metrics to identify bottlenecks.
* The Profiler window displays details captured with the selected module in the bottom panel.
* Unity Profiler assesses application performance and targets specific areas and issues.
* Be aware of performance differences when profiling in the Editor versus a standalone build.

**Profiling Workflow**