



COMP280: Specialisms in Creative Computing  
**9: Optimisation Process**



# Learning outcomes

- ▶ **Explain** the optimisation process
- ▶ **Explain** how a profiler is used
- ▶ **Understand** some of the key tools used for optimisation

# Intro

Programmers waste enormous amounts of time thinking about or worrying about, the speed of noncritical parts of their programs, and these attempts at efficiency have a strong negative impact when debugging and maintenance are considered. We should forget about small efficiencies, say about 97% of the time: premature optimisation is the root of all evil. Yet we should not pass up our opportunities in that critical 3%. –*Donald Knuth*

# Optimisation Process



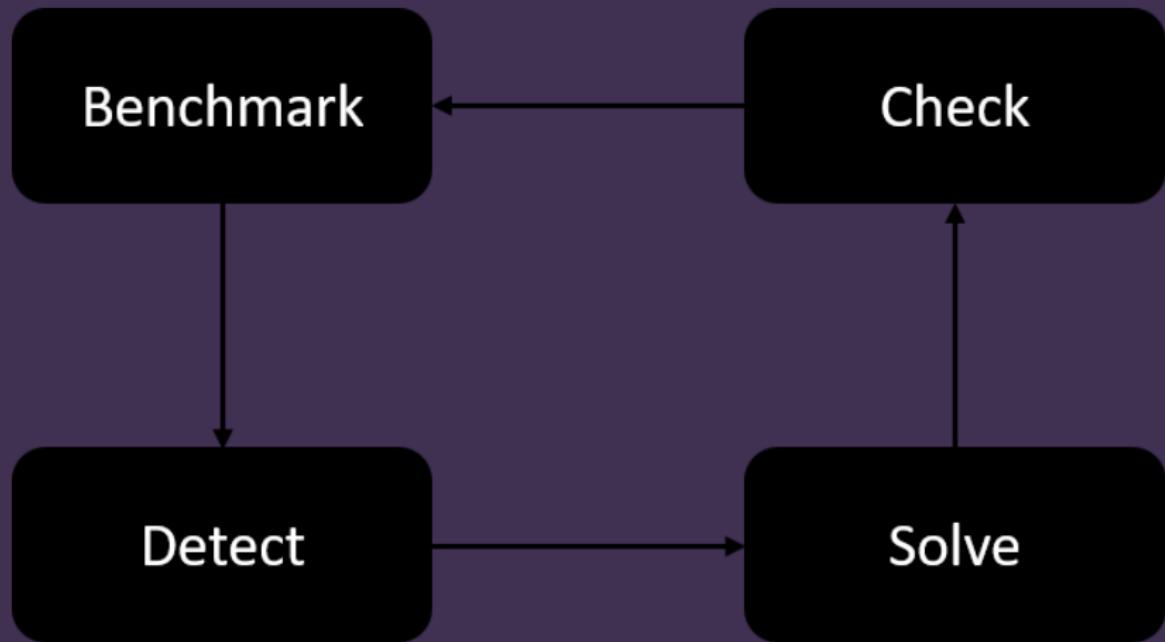
# Understand

- ▶ A few things to note when optimising
  - ▶ Limitations of the target hardware
  - ▶ Understand how the CPU/GPU works
  - ▶ Understand how your chosen engine works
  - ▶ Understand how your chosen language compiles
- ▶ For these workshops we are going to focus on a couple of these

# Optimisation Process

- ▶ Think of the quote in the intro, you should not worry about premature optimisation
- ▶ Get your core systems working and start optimising
- ▶ Make it a habit and part of your development process, not just when performance becomes a problem
- ▶ Every programmer in the team should optimise

# Optimisation Cycle 1



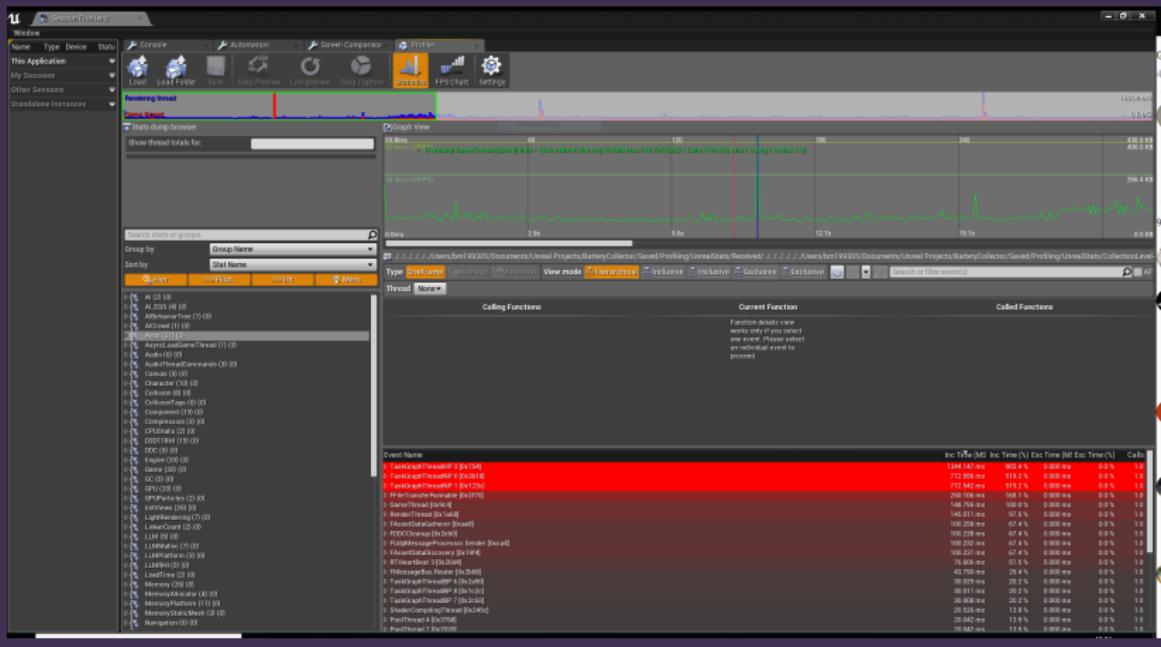
# Optimisation Cycle 2

- ▶ If you look at the previous slide, these steps feel like an **experiment**
- ▶ We just don't try to optimise **without** data
- ▶ We always **profile** and find out the bottle necks and hotspots
- ▶ We then make changes to our code base or content
- ▶ Then we return to start of the process using our benchmark as a comparison

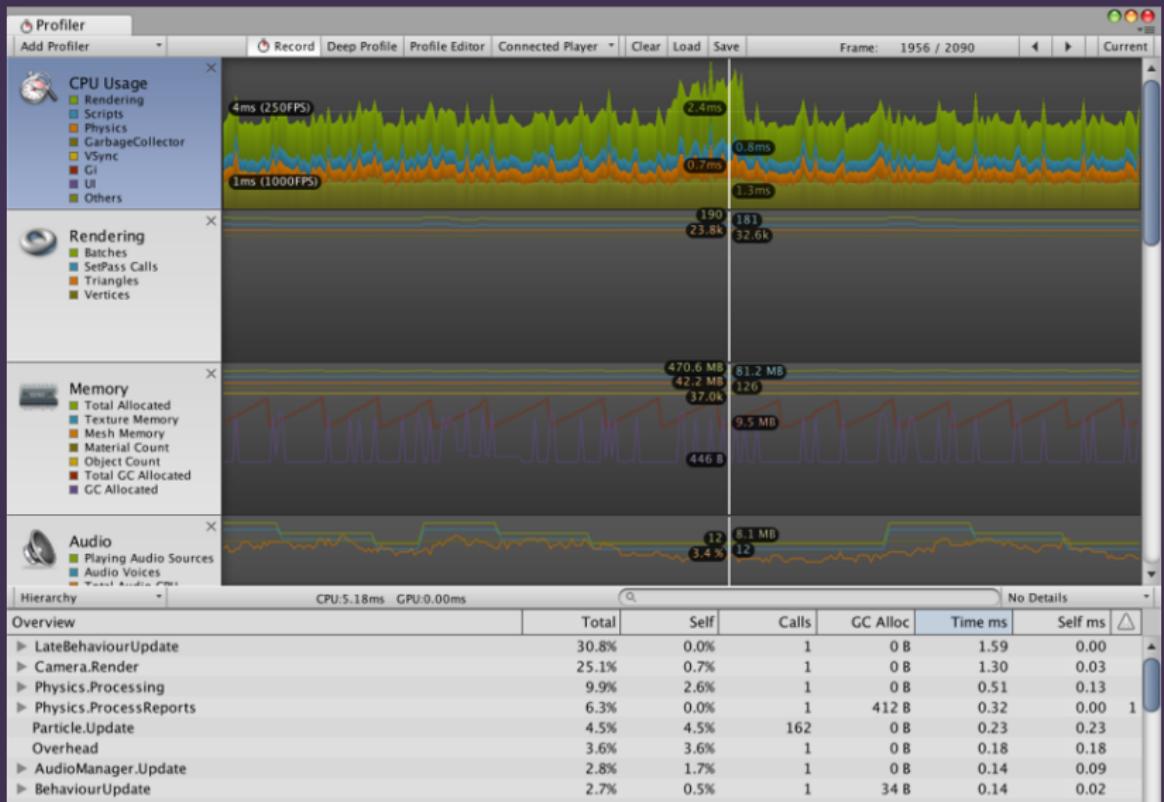
# Profiler



## UE4



# Unity



# Profiler

- ▶ You have already met the profilers for UE4 and Unity
- ▶ They have some commonalities
  - ▶ Visualise Performance via graphs
  - ▶ Performance measures (calls, time, frame time etc)
  - ▶ Code call graphs, usually linked to the visuals and perf measures
- ▶ This is the main tool you have to measure performance in your game

# Profiling Tips

- ▶ Run the profiler on a built version of your game
- ▶ Turn off features of the profiler you don't need

# Other Tools



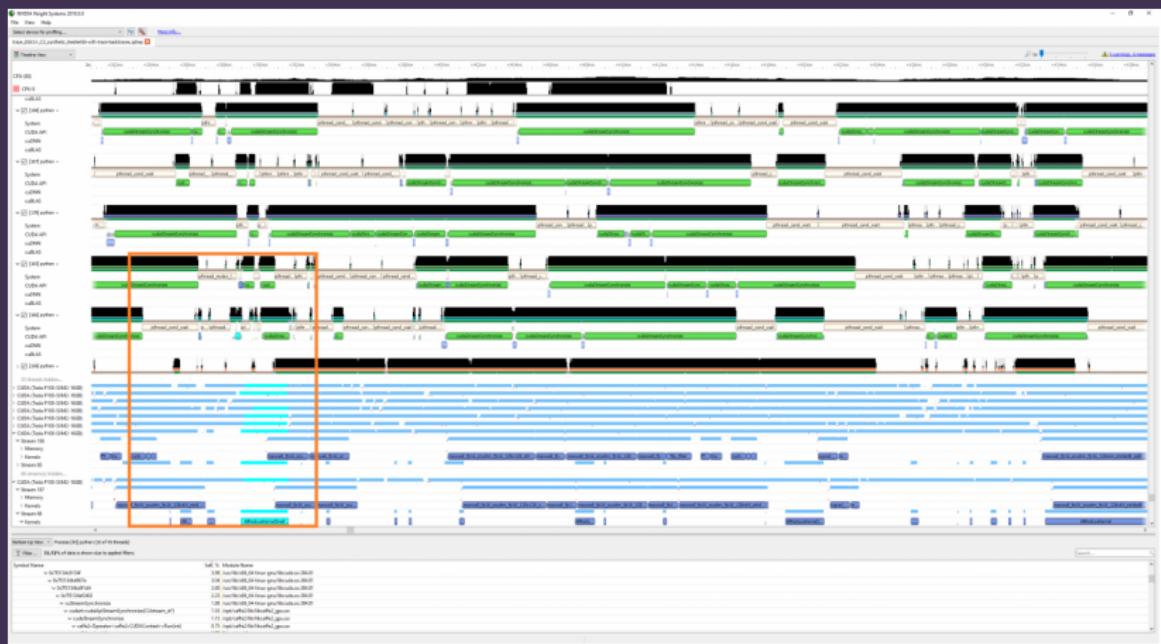
# Visual Studio

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Performance Explorer:** Displays a CPU usage timeline for a session of 13.42 seconds. The chart shows CPU usage in percent of all processors over time.
- Solution Explorer:** Shows the project structure for "TerminalHacking".
  - References
  - External Dependencies
  - Source Files
    - main.cpp
    - stlfile.cpp
    - WordList.cpp
- Properties:** pane showing file properties.
- Output:** pane showing build output and diagnostics information.

```
Show output from: DiagnosticsHub
Profiling of 'TerminalHacking' started.
TerminalHacking has exited.
Profiling of 'TerminalHacking' stopped.
```

# NVidia Nsight



# Render Doc

data2\_2018.03.01\_15.15.31\_frame@478.rdc - RenderDoc v1.0

File Window Tools Help

Timeline - Frame #8479

Usage for scratchrendertarget\_1118301577\_1600x900\_0\_1.vtex: Reads (▲), Writes (▼), Read/Write (△), Barriers (▲), and Clears (▲)

Event Browser

ID	Name	Duration (μs)
7670	vkCmdDrawIndexed(1020, ▲)	---
7676	vkCmdDrawIndexed(3678, ▲)	---
7682	vkCmdDrawIndexed(1715, ▲)	---
7688	vkCmdDrawIndexed(2688, ▲)	---
7694	vkCmdDrawIndexed(17847, ▲)	---
7698	vkCmdBindPipeline(12422, ▲)	---
7704	vkCmdDrawIndexed(11226, ▲)	---
7710	vkCmdDrawIndexed(3571, ▲)	---
7716	vkCmdDrawIndexed(1794, ▲)	---

API Inspector

7697

- commandBuffer: Command Buffer 12422
- pipelineBindPoint: VK\_PIPELINE\_BIND\_POINT\_GRAPHICS
- layout: Pipeline Layout 4058B78
- firstSet: 0
- setCount: 2
- pDescriptorSets:
  - [0]: Descriptor Set 5740440
  - [1]: Descriptor Set 2726082
- dynamicOffsetCount: 0
- pDynamicOffsets: uint32\_t[]

7698

- vkCmdDrawIndexed
- commandBuffer: Command Buffer 12422
- indexCount: 17847
- instanceCount: 1
- firstIndex: 0
- vertexOffset: 0
- firstInstance: 0

Callstack

Replay Context: Local data2\_2018.03.01\_15.15.31\_frame@478.rdc loaded. No problems detected.

Texture Viewer X Mesh Output X Pipeline State X Resource Inspector X

Cur Output 0 - scratchrendertarget\_1118301577\_1600x900\_0\_1.vtex

Inputs Outputs

PB 5 res3927 - tb\_colormap

PB 7 test5742 - east\_effect

Pixel Context

# Worksheet



# Optimisation Worksheet

<https://learningspace.falmouth.ac.uk/mod/resource/view.php?id=114234>

# Worksheet

