



Could ray-tracing be used to develop graphics hardware benchmarking tools which provide the user with more useful insight?

by

James Craddock-Jones

Supervised by: Andrew Green

Second Marker: John Williams

Project submitted as part of the requirements for
the award of BSc (Hons) Computing (Computer
Networks and Cybersecurity)

March 2022

Table of Contents

Abstract	4
1.0 Introduction	4
2.0 Literature Review	5
2.1 What is benchmarking, and who uses it?	5
2.2 What benchmarking tools exist?	5
2.3 Technologies used by existing benchmarks and games	11
2.4 What Graphics APIs Exist?	13
2.5 What is Image Reconstruction/Upscaling, and What Image Reconstruction/Upscaling APIs Exist?	15
2.6 What Game Engines Provide These APIs?	19
2.7 Further Research into Existing Tools	20
2.8 Conclusion	21
3.0 Aims and Objectives	22
3.1 Aim 1: Discover existing computer benchmarking tools and conclude where they could be improved	22
3.1.1 Objective 1:	22
3.1.2 Objective 2:	22
3.1.3 Objective 3:	22
3.2 Aim 2: Discover existing tools and technologies used by games and new and emerging technologies gaining traction	22
3.2.1 Objective 1:	22
3.2.2 Objective 2:	22
3.2.3 Objective 3:	22
3.3 Aim 3: Design and develop a new hardware benchmarking tool based on research carried out in the first two aims	22
3.3.1 Objective 1:	22
3.3.2 Objective 2:	23
3.3.3 Objective 3:	23
3.3.4 Objective 4:	23
3.4 Aim 4: Review and survey of proposed tool to gather feedback	23
3.4.1 Objective 1:	23
3.4.2 Objective 2:	23
3.4.3 Objective 3:	23

4.0 Research Methods	23
5.0 Resources Needed	24
5.1 Hardware	24
5.2 Software	24
6.0 Project Plan	26
7.0 Development Methodology	27
8.0 Design & Implementation	27
8.1 System Designs	27
8.2 System Implementation	31
8.3 System Testing and End-user Feedback	33
Functionality Testing 8.3.1	33
End-user Testing and Feedback 8.3.2	33
9.0 Evaluation	39
9.1 Development	39
9.2 Primary Research	40
10.0 Conclusions and recommendations	40
10.1 Conclusions	40
10.2 Recommendations	41
11.0 Reflection	41
11.1 Development	41
11.2 Primary Research	42
References	43
Appendix	45
A: Benchmark complaints	45
B: Initial survey findings	46
C: User feedback survey findings	50
D: Functionality testing	58
E: Consumer GPUs that support Ray Tracing	61
E.1 Nvidia Desktop GPUs	61
E.2 AMD Radeon Desktop GPUs	62

Abstract

As technological advances continue to grow year-on-year in both hardware and software, existing benchmarking tools fall behind and are slow to adapt to support or take full advantage of the advances companies such as Intel, AMD, and Nvidia make.

This results in hardware or software no longer being supported by benchmarking tools resulting in, hardware instability with existing benchmarks or the benchmark no longer working properly in newer operating systems.

Over the past few years technologies such as real-time ray tracing and rendering techniques such as Deep Learning Super Sampling and Fidelity FX Super Resolution have become accessible to the consumer space through various software and hardware innovations. The downside is that benchmarking tools have been slow to adapt to these new technologies.

While game developers have been quick to include such technologies in their new and existing games, for example, Shadow of the Tomb Raider, Cyberpunk 2077, and many more games like Call of Duty now include some form of ray tracing or DLSS and FSR implementation. Epic and Unity have also been adapting their game engines to meet the demands of game developers.

Benchmark creators such as Unigine and UL have made benchmarks that use some form of ray tracing. The issue comes with them being unrealistic workloads or just using algorithm-based ray-tracing techniques. Those algorithm-based ray tracing try to mirror the same visual effect as proper real-time ray tracing with as little performance loss but without the true visual quality.

1.0 Introduction

This project is looking into how existing graphics benchmarks can be improved by taking what people dislike about the existing benchmarking tools and then building upon them.

This will be done by creating a new one using the most appropriate and up to date tools, for example, game engines, APIs, and rendering techniques.

This will be done to fulfil what people disliked about the existing benchmarking tools and then add any possible additional features within the given period.

Below will provide a detailed analysis of the topic areas that need to be researched and discussed to reach this project's aim and objectives.

For example, it will go over the different graphical APIs used in the industry, new and emerging rendering techniques such as ray tracing, game engines that need to be considered for this project and the different possible image reconstruction techniques.

This will include technologies like image reconstruction that are currently emerging, which can show the user any possible performance gains they could witness within games when using them on older/less capable hardware.

An interest has been developed in this topic due to many complaints seen on a day-to-day basis in many different discord communities with existing benchmarks found within suites like 3DMark. For example, performance bias towards hardware makers that is not present in other variants of the same

benchmark, including pointless scores from pointless benchmark tests and many more. See appendix A for some evidence of complaints.

After reading up online on people's issues with current benchmarking tools, a small survey was sent out to a limited number of people who use these tools to help gauge hardware performance when conducting tests on many different pieces of hardware.

What was found from the small survey was that respondents thought that the benchmarks needed to be updated more with newer technologies which were stated multiple times. One comment on this matter was that *"New graphical technologies are being introduced over time and the benchmarks are starting to fall behind"*.

The fully complete findings from the questionnaire can be found in Appendix B, with further analysis found in section 2.8.

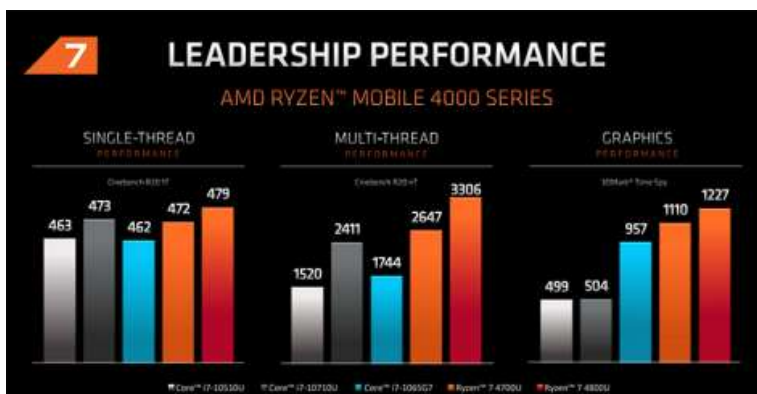
2.0 Literature Review

2.1 What is benchmarking, and who uses it?

A description of benchmarking is the comparison of one company's goods, services, and procedures to another. This is done to compare to one or more of those areas against those recognised as leaders. (13)

Benchmarking provides the data that is needed to see how one firm stacks up against their competitors, this can be done even if they are in a different industry, different product category, or serve a different group of customers. (13)

For example, in the hardware space, this would be seen as AMD, Intel and Nvidia comparing their products with each other's to show off in marketing slideshow much they are better than the others. An example of this is shown below.



(21)

2.2 What benchmarking tools exist?

2.2.1 3DMark

3DMark is a dedicated benchmarking tool that aims to measure the performance of devices over many different platforms such as PC and mobile. Each different benchmark is designed for those

platforms to scale over smartphones to high-performance PCs. These benchmarks aim to test the hardware to the best of their abilities by using graphical and computational workloads. The smoother these benchmarks run, the more powerful the device is. (9)

Each benchmark provides the user with a score based on how well the device ran the benchmark. The aim of this is to allow the user to compare one system to another easily. (9)

Below are all the current benchmarks provided with 3DMark and the target hardware for each.

3DMark benchmarks at a glance

3DMark includes many benchmarks, each designed for a specific class of hardware. You will get the most useful and relevant results by choosing the most appropriate test for your system.

BENCHMARK	TARGET HARDWARE	ENGINE	RENDERING RESOLUTION ¹
Time Spy Extreme	4K gaming with DirectX 12	DirectX 12 feature level 11	3840 × 2160 (4K UHD)
Time Spy	High-performance DirectX 12 gaming PCs	DirectX 12 feature level 11	2560 × 1440
Port Royal	Graphics cards with Microsoft DirectX Raytracing support	DirectX 12 feature level 12_1	2560 × 1440
Night Raid	PCs with integrated graphics	DirectX 12 feature level 11	1920 × 1080
Wild Life	Lightweight notebooks and tablets	Vulkan 1.1	2560 × 1440
Fire Strike Ultra	4K gaming with DirectX 11	DirectX 11 feature level 11	3840 × 2160 (4K UHD)
Fire Strike Extreme	High-performance DirectX 11 gaming PCs	DirectX 11 feature level 11	2560 × 1440
Fire Strike	DirectX 11 gaming PCs	DirectX 11 feature level 11	1920 × 1080

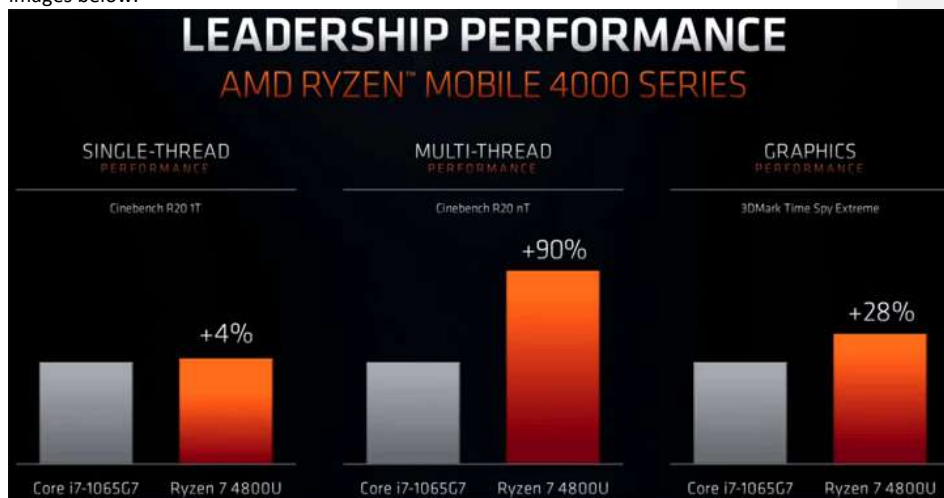
(9)

More will be explained in the next section on what DX12 and Vulkan are, what they provide, and the technologies these graphics APIs allow for. For example, DX12 Ultimate and Vulkan allow rendering technologies such as Ray Tracing, which 3DMark provides in their Port Royal benchmark.

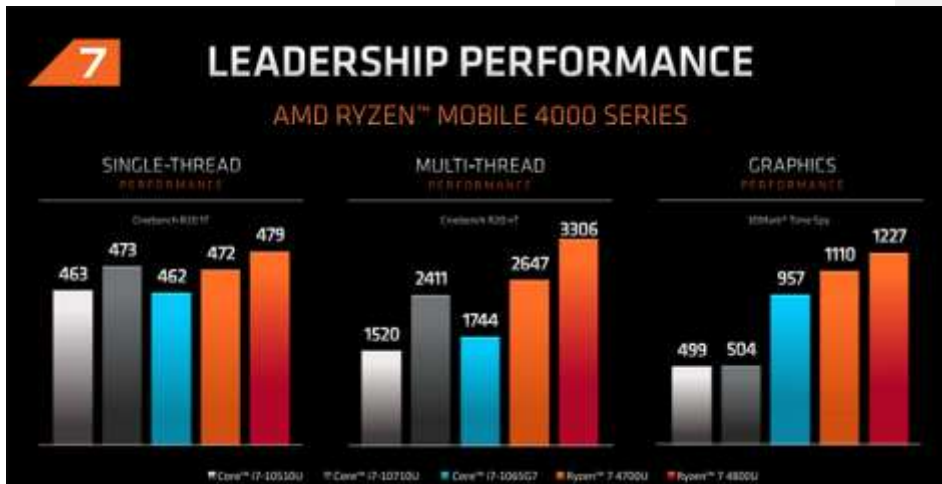
In addition, advanced images reconstruction techniques such as DLSS and FSR allow workloads such as games to run at lower internal resolutions. They do this while looking almost as good, if not as good, as the native resolution. 3DMark also provides some feature tests using these techniques, such as DLSS, for users to test them out for themselves.

3DMark works off a tiered basis, a basic edition that is free and only has a few of the benchmarks provided. An advanced edition with all the benchmarks but missing a few extra features like private results. Finally, a professional edition with all the benchmarks like the advanced edition but with all the additional features provided. (9)

This benchmark suite is used by hardware companies such as AMD when marketing the performance improvement between their own and their competitors' products, as shown in the images below.



(22)



(21)

2.2.2 Unigine Superposition

Superposition is one of the newest benchmarks released by Unigine over the last few years. The benchmark was created to test and be more compatible with new and upcoming technologies such as 8k displays and VR headsets. In addition, evaluating the performance of a graphics cards performance. (19)

Unigine is well known for its benchmarking tools. For example, Heaven was released back in 2009. This aimed to test features like tessellation and light effects like real-time global illumination. They also released a benchmark called Valley back in 2013. This built upon Heaven by adding dynamic skyboxes, volumetric clouds and more advanced lighting effects like sun shafts and ambient occlusion. (19)

Superposition builds on top of Heaven and Valley once again by adding even more advanced rendering techniques such as physically based surface materials. Unlike 3DMark, Superposition is just one benchmark with multiple graphic options and settings. These are 720p low, 1080p medium, high, and extreme, 4K optimised, and 8K optimised. They also let you choose from a performance benchmark mode and others, such as a stress test mode. (19)



(20)

Superposition work on both 64bit Windows and Linux and supports VR headsets such as the Oculus Rift and HTC Vine. Superposition also provides a leader board system just like 3DMark does with benchmarks, allowing results to be shared with everyone. (20)

The new UNIGINE 2 engine brings the new visuals that the Unigine team has created. The engine also provides an SSRTGI rendering technique which brings algorithm-based ray tracing to the benchmark. (20) This is similar to what 3DMark provides with their Port Royal benchmark but not the same, which will be discussed in the next section when talking about Ray Tracing and software emulated and hardware-accelerated Ray tracing in games and various other applications like 3D rendering.

Like 3DMark, Superposition also has paid tiers to gain access to certain features within the benchmarking tool, such as stress testing, posting scores onto the leader board, loop testing, and many more, as shown below.

EDITIONS

	BASIC	ADVANCED	PROFESSIONAL
	Custom VPL license	Commercial VPL license	Commercial Company perpetual + Support Office Hardware Maintenance
Engine Settings	10	10	10
RTT Monitoring	10	10	10
Procedural Model	10	10	10
Perceptual Map	10	10	10
VPL Experience	10	10	10
Randomized Partitioning		10	10
Cooperated Stress Test		10	10
Customized Link Information			10
Ray-Plane Deep Analysis			10
CDR Reports			10
Commercial User			10
Technical Support			10
Price	Free	\$19.9K	\$49K
	DOWNLOAD	SELECT	SELECT

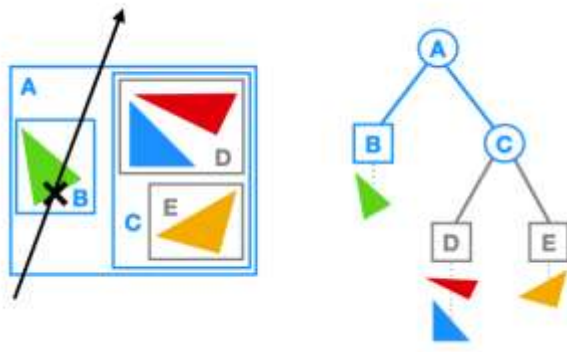
(25)

2.3 Technologies used by existing benchmarks and games

2.3.1 What is Ray Tracing?

2.3.1.1 Software Accelerated Ray Tracing

BVH stands for Bounding Volume Hierarchy, BVH is the most common choice for ray tracing alteration in data structures with graphics cards. (29, p1)



(29, p3)

The above shows an example of a BVH data structure that is containing four different primitives (for example, objects within a scene). Bounding volumes (these are called axis-aligned bounding boxes) are used within this data structure and are split up into BVH nodes with the four primitives being referenced

in leaves. The whole subtree of node C can be cut when traversing through the BVH. This is done to discover the ray intersections because the rays do not intercept the bounding box/boxes. (29, p3)

Software Emulation for BVH Search

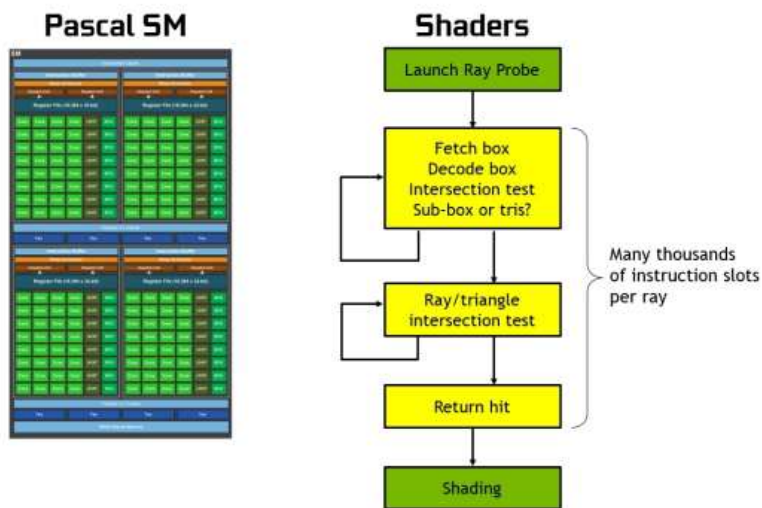


Figure 19. Ray Tracing Pre Turing

(3, p31)

The above shows how ray tracing is emulated within the BVH search of Nvidia GPUs, which do not include dedicated ray-tracing cores made for these specific tasks. This method is much slower and inefficient due to the processing power needed to draw rays onto the screen. This method is impossible to do in real-time with current GPUs without dedicated ray-tracing hardware. (3, p30)

2.3.1.2 Hardware Accelerated Ray Tracing

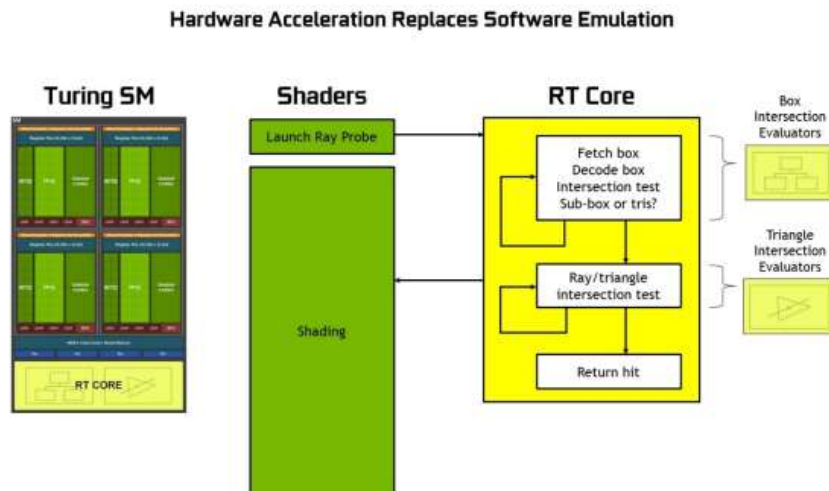


Figure 20. Turing Ray Tracing with RT Cores

(3, p31)

The above now shows how the dedicated RT cores (dedicated cores designed to accelerate and offload ray tracing workloads from the normal raster calculation pipeline) are used to accelerate the process of ray tracing within Nvidia RTX graphics cards. Nvidia notes that with this process, the BVH and RT cores can work together to provide a performance boost over software emulated ray tracing, with the ability to back and forth between the BVH and RT cores when performing calculations to render a scene. (3, 30)

Nvidia notes that a 1080Ti with no RT cores is 10x slower than a 2080Ti with dedicated RT cores when performing the same ray tracing workloads. (3, p32)

2.3.1.3 What consumer GPUs support Ray Tracing?

Any GPU that supports the DX12 Ultimate API (Application Programming Interface) will support ray tracing. The following has been taken from the TechPowerUp GPU database. Appendix E shows an example list for AMD and Nvidia GPUs with dedicated Ray Tracing hardware.

2.4 What Graphics APIs Exist?

2.4.1 DX12

DirectX12 is a complete redesign of the continuously improved DirectX APIs used in games to improve performance by delivering a lower-level API compared to previous DX APIs like DX11. DX12 aims to

provide more predictable and consistent behaviour by removing most if not all software layers within the API while also allowing for better compatibility across different Windows platforms. (10)

DX12 also removes the burden drivers were left within regarding queuing, scheduling, and synchronising tasks within hardware, such as CPU and GPU. DX12 aims to be a superset of the already widely used DX11 API on a feature level. (10)

DX12 Ultimate builds upon DX12, adding more features and setting a new standard of what is capable from new hardware such as GPUs. DX12 Ultimate brings DXR Ray tracing, which aims to bring a new level of graphical capabilities to games allowing for more realistic rendering of light and reflections. (11)

It also adds Variable Rate Shading, which allows the game developer to vary the shading of objects in-game to improve performance ever possible, as well as Mash Shading that allows the game developer to harness the full compute power of a GPU for the geometry pipeline allowing for more detail while keeping performance the same if not better. (11)

2.4.2 Vulkan

Vulkan is a graphics API developed to replace OpenGL due to the API's performance inefficiencies regarding both the GPU and CPU. Vulkan aims to improve on keeping the GPU busy, removing driver overhead on the CPU, allowing for more performance out of the system. (12)

Vulkan is developed by the Khronos Group, a member-funded not-for-profit industry consortium. Below are the members of Khronos who are helping with the development of Vulkan. (12)



(12)

Just like DX12, Vulkan is a low-level API meaning the developer has to provide more information instead of leaving it to the driver to sort out. It also has no “current state” that the driver maintains. Vulkan also removes the bloat of OpenGL by actually removing deprecated features and functions, for example, built-in pipeline transformations. (12)

Vulkan only half bakes and compiles the shaders. The rest have to be compiled within the runtime pipeline-building processes. (12)

As shown in the image below, Vulkan is supported and worked on by a large majority of the game engines open to the public, like Unity and UE. (12)



(12)

As discussed in the previous section, both graphics APIs just discussed allow for some advanced rendering techniques and technologies such as Ray Tracing. Still, in addition to that, they allow for advanced image processing techniques like Deep Learning Super Sampling (NVidia) and Fidelity FX Super Resolution (AMD). These technologies allow for 3D applications to run faster while retaining as much image quality as possible allowing more from weaker hardware which is welcomed in a time of shortages and price hikes of newer hardware. These two technologies will be discussed in the following section in detail.

2.5 What is Image Reconstruction/Upscaling, and What Image Reconstruction/Upscaling APIs Exist?

The most straightforward and basic upscaling technique is to stretch a lower quality image onto a bigger monitor. To make up for the extra pixels leftover on a higher resolution display, pixels from the lower resolution image are copied and repeated. (14)

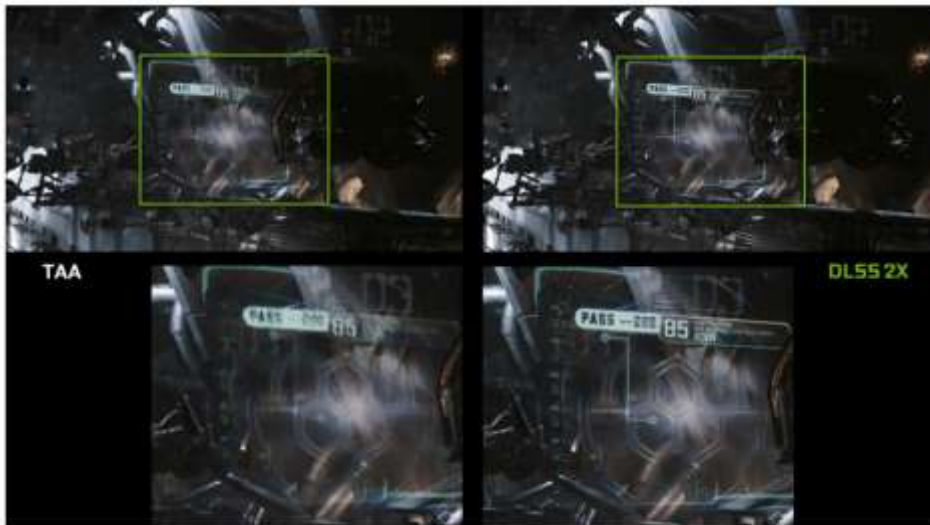
Filtering is used to smooth the appearance and round out any jagged edges that may have appeared as a result of stretching the images to fit the display's high resolution. As a result, the image fits, for example, on a 4K television, although it is generally muted or fuzzy. (14)

2.5.1 NVidia Deep Learning Super Sampling

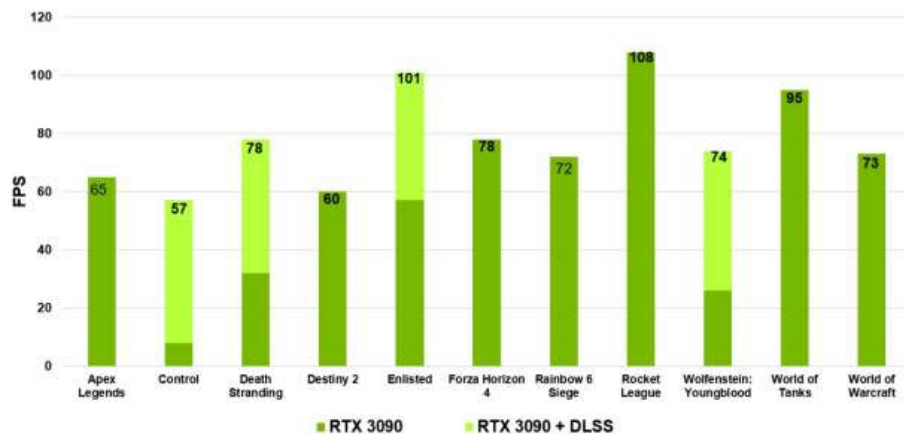
DLSS, also known as Deep Learning Super Sampling, is an AI (Artificial Intelligent) upscaler that has been developed by Nvidia to be used within applications such as games to help improve native resolution and performance by rendering from a lower resolution. This is done by using AI to analyse frame data like TAA and upscale those frames to the native resolution while keeping as much detail as possible and adding more detail than before while also adding a noticeable performance increase. (3) (4)

The downside to this is that it uses tensor cores which can only be found within graphics cards made by Nvidia, meaning that graphics cards without will not be able to utilise this method. (3) (4)

Below shows this method compared to normal TAA and the performance increase that can be seen in games.



(3, p36)



(4, p26)

2.5.2 AMD Fidelity FX Super Resolution

AMD Fidelity FX Super Resolution, also known as FSR, is an open-source compute shader that aims to provide the same experience as DLSS by rendering graphically intensive scenes at lower resolutions and upscaling them back to native resolution within as little quality loss as possible. (7) Unlike DLSS, this upscaling method can be used across many platforms and devices without the need for any specific kind of hardware ASIC, for example, Tensor cores, which allows it to be much more versatile and easier to integrate. (7)

FSR also uses many different quality options to choose from, such as ultra-quality, quality, balanced, and performance. The only difference is the render resolution scale, with them respectively being 77%, 67%, 59%, and 50% native resolution. (7)

FSR also works on many popular graphics APIs such as DX12 and Vulkan, with support for more legacy graphics APIs like DX11 when recompiling the shader to support it. (7)

Below shows in more detail what FSR is and how it works:

WHY USE AMD FIDELITYFX SUPER RESOLUTION 1.0?



Super Resolution

Generates "super resolution" images each input frame.

Support for any area scale factor between 1x and 4x.

Large performance improvements over native rendering.



Cross-platform

Runs on a wide variety of GPUs, including older architectures.

Can be ported onto multiple platforms without restriction.



Open source

Provided on GitHub under an MIT license.



Easy to integrate

Full source is provided.

Native samples and VBA patch provided.

Also available in Xbox GDK.

Requires no historical context.



Highly optimized

Hand optimized for great performance across a wide variety of GPUs.

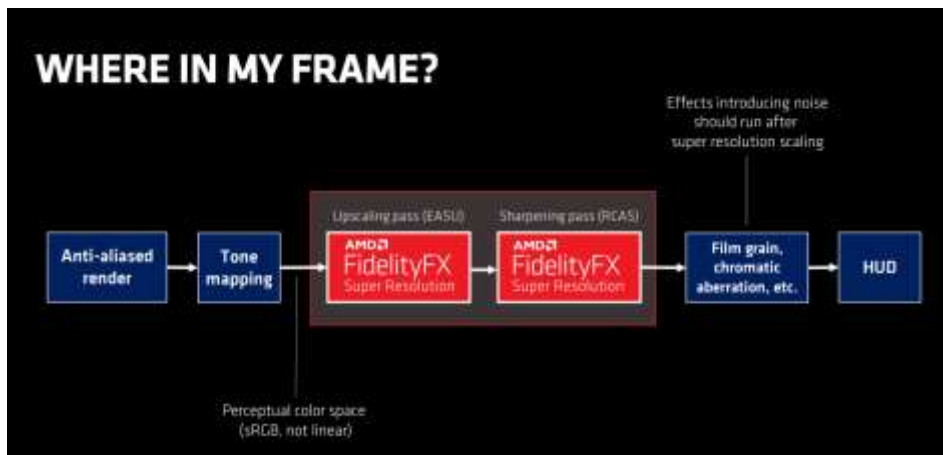
(7, p4)

FSR QUALITY MODES

FSR 1.0 quality mode	Description	Scale factor	Input resolution	Output resolution
Ultra Quality	Ultra Quality mode produces an image with quality visually indistinguishable from native rendering. It should be selected when the highest quality is desired.	1.3x per dimension (1.69x area scale) (77% screen resolution)	1477 x 831 1970 x 1108 2646 x 1508 2954 x 1662	1920 x 1080 2560 x 1440 3440 x 1440 3840 x 2160
Quality	Quality mode produces a super resolution image with quality representative of native rendering, with a sizeable performance gain.	1.5x per dimension (2.25x area scale) (67% screen resolution)	1280 x 720 1706 x 960 2293 x 960 2560 x 1440	1920 x 1080 2560 x 1440 3440 x 1440 3840 x 2160
Balanced	Balanced mode produces a super resolution image approximating native rendering quality, with a major performance gain compared to native.	1.7x per dimension (2.89x area scale) (59% screen resolution)	1129 x 635 1506 x 847 2024 x 847 2259 x 1270	1920 x 1080 2560 x 1440 3440 x 1440 3840 x 2160
Performance	Performance mode visibly impacts image quality and should only be selected in situations where needing additional performance is critical.	2.0x per dimension (4x area scale) (50% screen resolution)	960 x 540 1280 x 720 1720 x 720 1920 x 1080	1920 x 1080 2560 x 1440 3440 x 1440 3840 x 2160

For consistency AMD recommends using these quality presets when exposing FSR in your game!

(7, p10)



(7, p13)

The technologies discussed above, such as FSR, DLSS, and Ray Tracing, can all be found within modern game engines that aim for the best visual quality in the industry, two of the many engines on the market and of most popular that support these are Unity and Unreal Engine. The following section will go over the advantages and disadvantages of both Unity and Unreal Engine to formulate an idea on which is more suited to this project given the scope and limited period for it to be completed.

2.6 What Game Engines Provide These APIs?

2.6.1 Unity

2.6.1.1 Advantages

Unity is relatively easy to make small games vs other engines, for example, Unreal. This is because it is the more popular game engine and has more guides on websites like YouTube that help speed up development when you are stuck. (8)

It is also more popular and easier to use because it uses C#, which is more accessible than C++, unlike Unreal when not visual scripting. (8)

2.6.1.2 Disadvantages

The graphics side of the engine is not as good as other engines like Unreal, with some scenes looking less realistic. Unity also has slow rendering times, for example, light rendering within complex scenes. (8)

Unlike Unreal Engine, Unity's source code is only given out under certain conditions. It also comes with a cost meaning it is harder to tweak and add more features you need to the game engine. (8)

2.6.1 Unreal

2.6.1.1 Advantages:

The Unreal Engine has a built-in visual scripting solution. Visual scripting makes it easier for games to be developed as you do not need to know how to code well. This also makes development more straightforward as you do not have to code everything yourself (8)

Unreal Engine can be argued for producing some of the best quality graphics in the industry. (8)

Unreal Engine is open source, meaning that you can customise it to your needs if it does not have a feature that you would like or a feature that needs altering to meet your needs. (8)

2.6.1.2 Disadvantages:

Visual scripting cannot do everything and will require C++ knowledge, which may hinder development if C++ knowledge is not up to scratch. (8)

Once you start making over \$1 million with your Unreal projects, you must begin paying a 5% royalty fee which may not be the best for small businesses or game studios using the engine. (8)

The Unreal Engine also has a smaller asset store when compared to Unity meaning that you will have to create more assets yourself which could increase spending and require more time to complete. (8)

2.7 Further Research into Existing Tools

Further research was carried out into this topic area. This included a questionnaire and searching through online forums to find complaints from users of these benchmarking tools. What was found from searching online forums was that some people experienced issues with scores being inconsistent, leading them to believe that they had defective hardware. Source 26 shows an example of this where a user thought that their i9 9900K was defective when running 3DMark TimeSpy.

In addition to that, source 27 shows further issues with the 3DMarks benchmark suite with Firestrike being incompatible and unstable with a multi-GPU technology that was still being supported at the time, Crossfire. The user reports that Firestrike would not open when both Sapphire HD 7950 Vapor-Xs in crossfire are **overclocked** even though the overclock is stable. Even an extremely small overclock still causes Firestrike to not open.

Further results from the initial primary research also show that on two occasions, participants stated that they would like to see a more lightweight Ray Tracing benchmark rather than heavy ones provided in current benchmark suites, *“would also be nice to have a lightweight Ray Tracing benchmark instead of a heavy one like Port Royal”* and *“Something interesting would be low-load synthetic raytracing tests compared to the heavy raytracing tests available now.”*. The participant who talked about a lighter benchmark provided more information regarding what they meant after submitting their questionnaire results. Regarding lightweight, they meant more realistic use cases for ray tracing that are used in games rather than over the top. For example, every surface has high-quality, high-resolution reflections.

An example of heavyweight ray tracing workload:



(28)

Almost all participants stated that they liked current benchmarking suites as, for example, they allow them to make quick and easy comparisons between changes in hardware such as clock speeds. *"They allow you to easily identify the performance differences between different hardware as well as OC vs non OC"*, and they also stated that they liked them as they allow for you to test many different technologies and scenarios *"Good coverage of a variety of graphics scenarios and technologies with consistent scoring"*.

The ones that did not like existing benchmarking tools stated that they do not like them due to issues such as performance bias towards certain hardware makers, as shown with the initial finding over different discord servers, *"Either have a performance bias towards certain hardware or haven't been updated in a long time"*.

The fully complete findings from the questionnaire can be found in Appendix B.

2.8 Conclusion

Hardware benchmarks are the same tools used by companies such as AMD, Intel and Nvidia to show the performance improvements gen on gen compared with competitors' products.

Technologies and APIs are continually progressing as time goes on, allowing for such tools to take advantage of rendering techniques new to the consumer space, such as ray tracing with hardware acceleration for RT workloads now being added in the latest graphics cards such as the RTX 20 and 30 series as well as the RX 6000 series.

Over the years, many benchmarks have been created to take full advantage of these new technologies such as 3DMark Port Royal and Unigine Superposition, but this has come with some downside such as

bias in scoring and scores making zero sense. They all use in house engines that may never become exposed outside of the benchmarks.

Therefore, this project proposes creating a new graphics benchmark using a game engine that is open to anyone to use for their games, this being Unity. Unity will allow for creating a graphics benchmark that can simulate actual game performance more realistically, thanks to it being available to anyone. It will also allow for the use of new technologies such as DXR Ray Tracing and new rendering technologies such as FSR.

Visual Studio will also be used to create a Windows Forms application so that the user has a GUI that they can select a benchmark and then view the results after. Existing benchmarks can have a portable sharing of results. The proposed benchmarking tool will do the same using a web page to display the results exported from the Unity benchmark once the benchmarking run has finished.

3.0 Aims and Objectives

3.1 Aim 1: Discover existing computer benchmarking tools and conclude where they could be improved

3.1.1 Objective 1:

Research online existing benchmarks to gain enough information to create surveys

3.1.2 Objective 2:

Conduct a survey asking computer enthusiasts what they like and dislike about existing solutions

3.1.3 Objective 3:

Analyse findings to conclude how they could be improved given the project's period.

3.2 Aim 2: Discover existing tools and technologies used by games and new and emerging technologies gaining traction

3.2.1 Objective 1:

Research online existing technologies and research online existing tools

3.2.2 Objective 2:

Research online new and emerging technologies.

3.2.3 Objective 3:

Analyse findings to draw conclusions on how they could be integrated into the project given the period.

3.3 Aim 3: Design and develop a new hardware benchmarking tool based on research carried out in the first two aims

3.3.1 Objective 1:

Design GUI for the main application and the portable HTML file to show results, incorporating what was found out in the last two aims.

3.3.2 Objective 2:

Develop the main application, portable **HTML web page** and small game scene within Unity with all the basics needed.

3.3.3 Objective 3:

Test the solution on several different pieces of hardware to see if the main application, portable HTML document and the unity demo all work as intended.

3.3.4 Objective 4:

Make any improvements, if necessary, based on what was found from the testing stage.

3.4 Aim 4: Review and survey of proposed tool to gather feedback

3.4.1 Objective 1:

Design and create a survey with the final solution attached.

3.4.2 Objective 2:

Release survey out to the previous participants and wait for results to come back for analysis.

3.4.3 Objective 3:

Once feedback has returned, analyse to draw a conclusion on the effectiveness of the new solution too, for example, discuss advantages and disadvantages of the new solution and any further possible improvements that could be made to the final solution.

4.0 Research Methods

The below has been decided based on research **from sources 15, 16, 17 and 18**. This information has been used to find what best suits the circumstances at this current time for this project with the current COVID pandemic and the type of outcome the project is realistically looking at.

- **Interpretivism** – The project will ask people their opinion on current graphics benchmarks used within the industry to see if any improvements could be made based on current experience using such tools. Interpretivism in this context is suitable because the project asks for people's ideas and thoughts on a topic.
- **Inductive** – inductive research is being used here as the project asks the wider community for feedback to generate ideas and theories on how existing benchmarking tools can be improved. Deductive research is suitable here as the project builds up a theory based on research from the wider community.

- **Action Research and Grounded Theory** – action research and grounded theory are being used here as the project will see if there are any strengths and weaknesses to existing graphics benchmarks that could be built upon.

This is done by asking people who use these tools. The grounded theory comes in when using this data generated to create a new graphics benchmarking tool based on what was found. These are suitable here as the project investigates existing graphics benchmarking tools to create a new, better one.

- **Cross-Sectional** – Cross-sectional is being used here as the project is asking for feedback from many different people who are hardware enthusiasts over a brief time span.

This feedback will be both about existing benchmarking tools and the solution produced by this project. Cross-sectional is suitable for this project because as stated before, the brief time span this project is being carried out for.

- **Questionnaire** – questionnaires have been chosen for this project as they allow me to gather the data from the participants safely given the current situation with COVID-19.

The project will do this using online tools like Microsoft Forms and ask for feedback on existing and the possible new tool created at the end.

Questionnaires are suitable in this situation due to the ever-changing COVID-19 pandemic, and it allows more people to respond and respond in their own time.

Further this point, the people who will be testing this are found within many different time zones around the world with, for example, many different work schedules and other arrangements, making other activities such as video conferencing difficult and in cases impossible to conduct, thus making this the better and more accessible option for both the participants and the project.

5.0 Resources Needed

5.1 Hardware

A computer equipped with a DX12 Ultimate/DXR compatible graphics card will be needed for this project to be carried out, as shown in the above RT section. The CPU will also need to support the X64 architecture with the addition of SSE2 instruction set support as stated by Unity on their website here: <https://docs.unity3d.com/Manual/system-requirements.html>

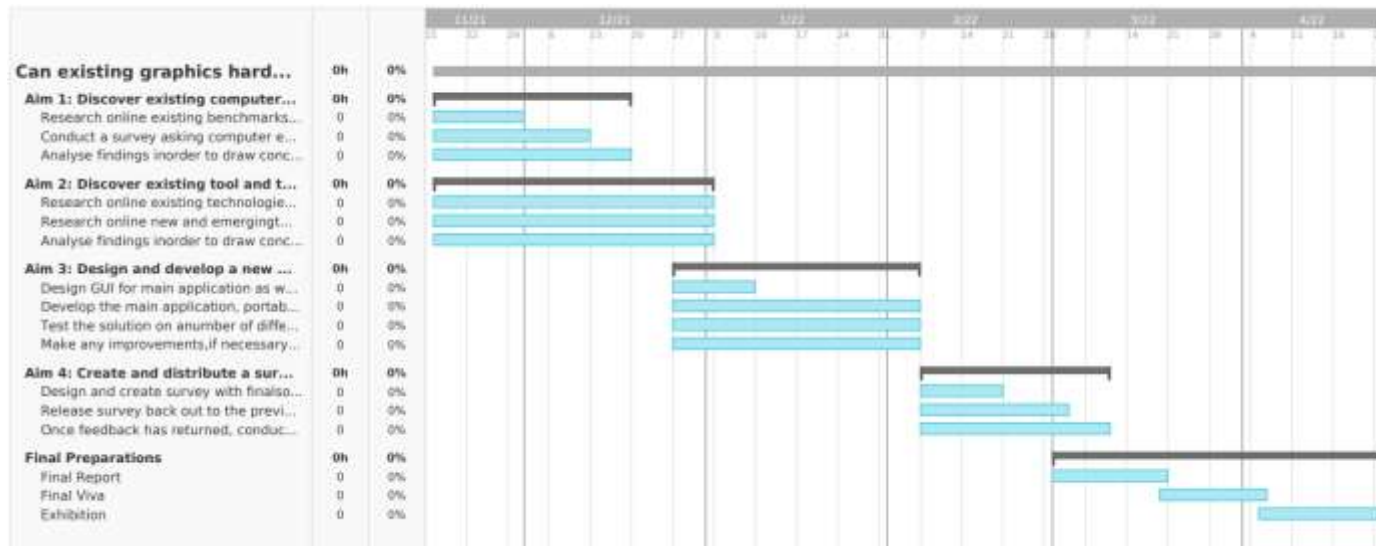
5.2 Software

The project will require the latest version of Unity to create the actual 3D ray-tracing demo to test the graphics hardware. It will also require APIs such as OpenHardwareMonitor to gather sensor data from

the graphics hardware and other hardware within the system such as the CPU to find possible bottlenecks elsewhere.

In addition to that, to create the GUI for the user to select the benchmark they wish to run, Visual Studio will be required to develop and build the windows application in C#. C# will be used throughout this project due to the vast number of online resources.

6.0 Project Plan



Above is the Gantt chart used to plan that is being used to organise and time manage this project. Every aim has been split up into its group with its corresponding objectives allowing for the utmost vigilance when keeping track of all the tasks that are needed to complete the project.

The Gantt chart is also keeping track of other preparations needing to be done outside of this final report that may get forgotten about overtime, for example, the last via and exhibition.

As well as helping to split the workload out over time, this is also helping to keep track of progress while also mitigating the possibility of going overtime by keeping a record of what is done and what needs to be done by a specific deadline, which is essential when time is limited in the way that it is for this project.

7.0 Development Methodology

The spiral development methodology has been chosen to develop the project's end goal. This has been chosen as spiral allows for constant review of the project's current state, allowing for judgments to be made based on what has been found on the latest prototypes allowing for more refinements to be completed and the result to be as refined as possible while also meeting the aims that were set out. (23)

Other advantages of this approach are that constant risk assessment takes place every time a spiral is completed allowing for checkups to see if the project is still on track to meet, for example, the time deadline and or what is required of the final solution. The downside to this is that time management is not the upmost priority of the model as the development typically has no end or set end date and can waste time worrying about risks with the project at the same time. (24)

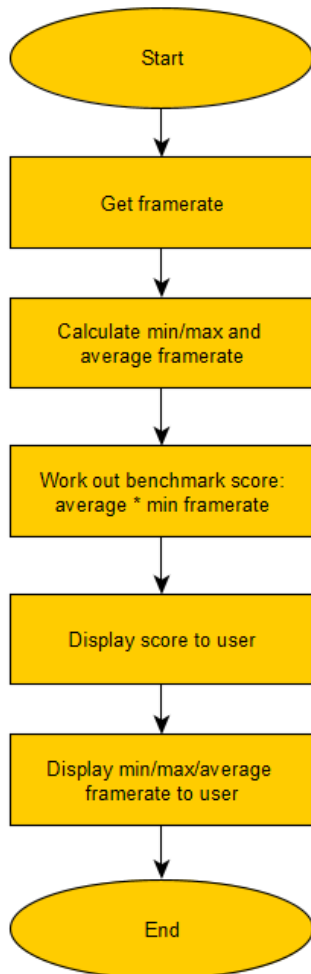
Commented [AG24]: Is this likely? Are you gaining feedback as you go?

8.0 Design & Implementation

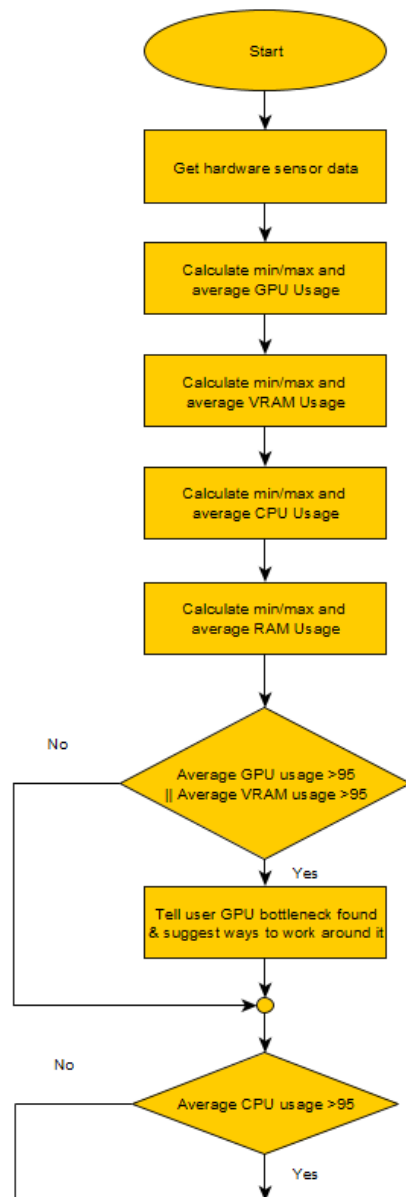
8.1 System Designs

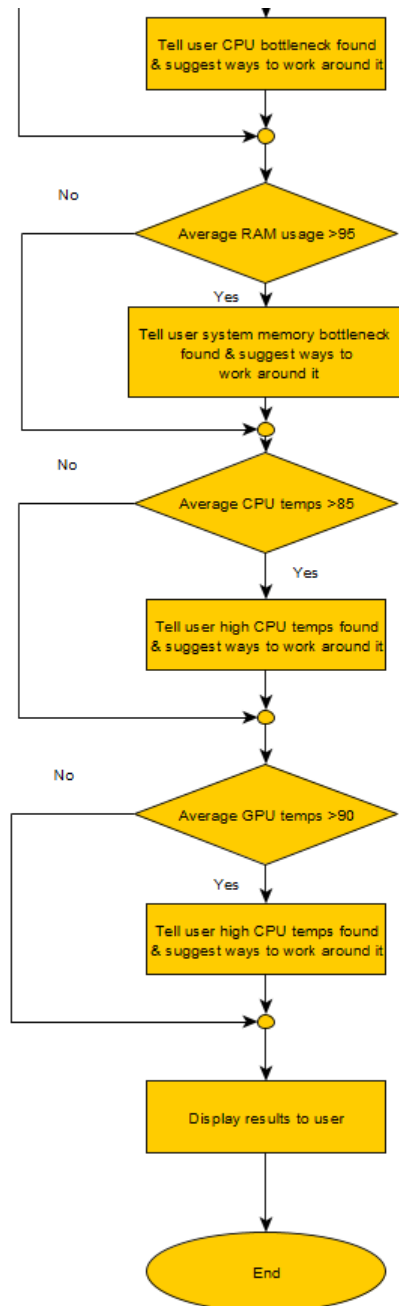


The above shows a screen flow diagram of the proposed system. The new system will use a C# application to launch the Unity benchmark. Unity will then go full screen to conduct the benchmark and will finally lead back to the C# application to show the results to the user and provide information such as any possible bottlenecks found during the benchmark.



The above shows the flow chart for how the Unity benchmark will calculate the graphics score form within the Unity benchmark that will be saved within a text file for the C# application to read.





The above also shows how the new system will detect a bottleneck within the user's device. For example, this will be done by looking at CPU and GPU usage, VRAM usage, CPU and GPU temperatures to see if there are thermal issues, and RAM usage. Suppose a component usage is high or the temperatures of the components are high, for example, more than or equal to say 90. In that case, the system will notify the user of an issue with that component in the device and suggest ways to improve it.

8.2 System Implementation

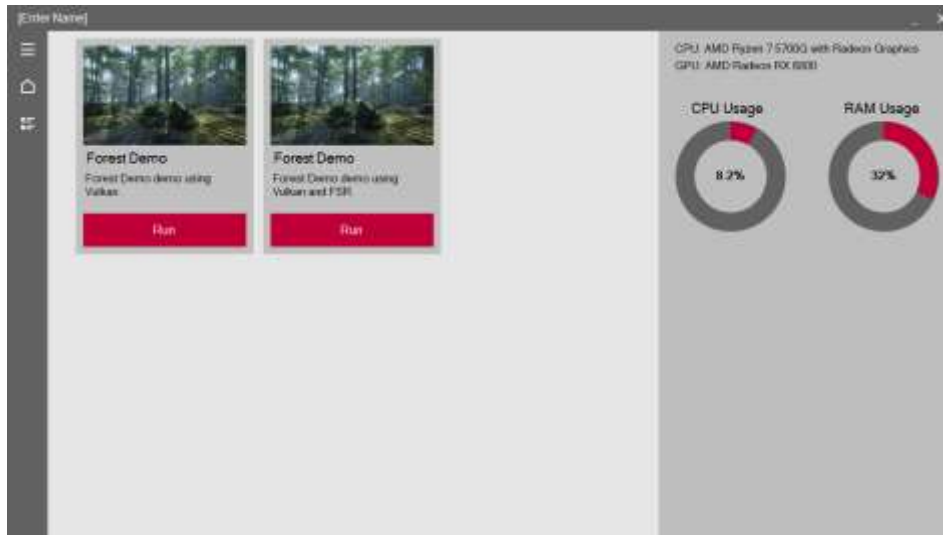
To create this new benchmarking system, tools such as Unity have been used to create the actual 3D demo. Unity was used as it provided all the tools required to develop this project, such as access to a DXR implementation for real-time ray tracing and new technologies such as AMD FSR.

Issues arose during the development of the system with Unity as the API OpenHardwareMonitor only partly works. For example, it can only detect the sensors for the GPU. One reason for this was that Unity could not open with admin rights, resulting in the API not having the necessary rights for the CPU sensors.

A workaround for this issue was to make the C# application open with admin rights and when launching a benchmark check to see if the benchmark is running and while doing so, record the CPUs sensor data using the same APIs until the benchmark had closed and then writing all the data to the text file created by the unity demo. This was also done to record RAM usage and provide the system specification and a timestamp for completion so the user can move the result text file elsewhere if they wish.

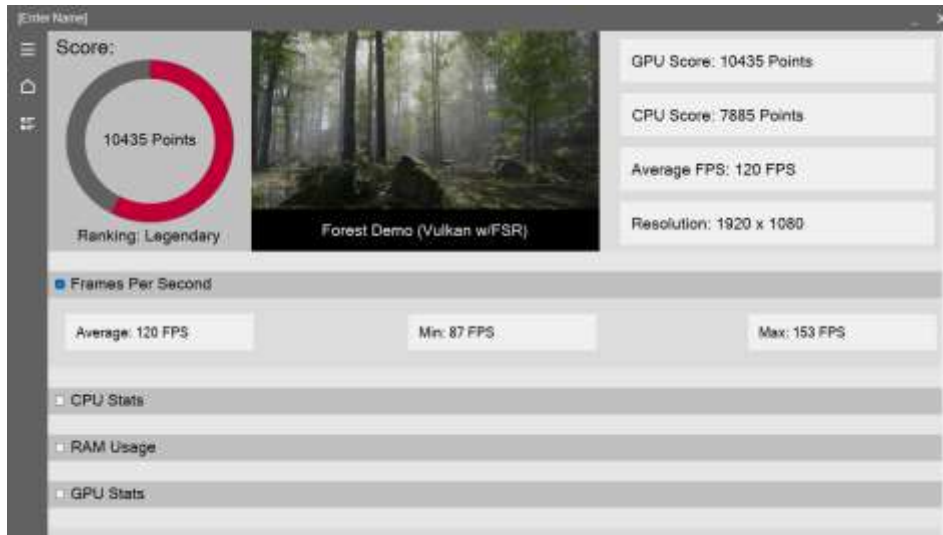
Other issues arose when displaying the results page. This was because the page to launch the benchmark was also a user control instance, meaning it did not have access to the main form to change between itself and the results user control. A new form was created to get around which the main menu user control could launch when the benchmark has been completed to show the results user control.

Main Menu:



The above shows the main menu of the application. This provides the users with the two different benchmarks that can be run on the user's computer and basic sensor data from their computer, for example, CPU and RAM usage. The main menu also shows the user the specification of their computer, such as the CPU and the GPUs installed within.

Results Menu:



The above shows the results page shown to the user once the application has completed the benchmark. This shows the user a CPU, GPU score, and the FPS, resolution that was run throughout the benchmark, the graphics API and whether FSR was used, and finally the various sensor data gathered from the computer's hardware with a recommendation of improvements and bottlenecks found.

8.3 System Testing and End-user Feedback

Functionality Testing 8.3.1

The system went through functionality testing before being sent out to the users to test, see appendix D.

End-user Testing and Feedback 8.3.2

For the first question, the users were asked how easy it was to use the system's GUI, in which out of 20 results, they responded with 4.25 stars on average.



After that, they were asked to explain why they thought it was easy to use or hard to use the system's main GUI. Most respondents responded that it was easy to understand thanks to it being like existing

benchmarks and/or that the learning curve was simple as it took design cues from existing tools and interfaces.

2. Please explain your answer to the last question

[More Details](#)

[Insights](#)

20

Responses

Latest Responses

"Nice and simple making it easy to use and learn"

"Easy to use and is clear"

"Doesn't look that pleasing to the eye, could use a new colour palet..."

15 respondents (75%) answered **easy** for this question.

simple **easy** tools
GUI design existing benchmarks

The users were asked how much they liked the new benchmarking tool. The 20 users responded with an average of 3.6 out of 5 stars as their rating.

3. Out of 5, how much do you like this new 3D benchmarking tool?

[More Details](#)

[Insights](#)

20

Responses



3.60 Average Rating

When asked why, they responded that it is a nice mix of both Unigine and 3DMark but does have a few issues that are hindering its overall success and that it needs refinement for them to prefer it more and to recommend it better.

4. Please explain your answer to the last question

[More Details](#)

[Insights](#)

20

Responses

Latest Responses

"It's a nice and much needed mix of Unigine and 3DMark"

"Would have liked something more unique"

"It's ok in its current state but could be improved to be more useful"

7 respondents (35%) answered **nice** for this question.

Word cloud content:

- mix of Unigine
- nice mix
- Interesting benchmark
- similar
- nice concept
- work
- hardware issues
- little refinement
- bug
- lot
- little
- nice
- mix of unigine
- benchmark
- Unigine Valley
- useful information
- tools
- nice to-use
- unique mix
- little benchmark

They were then asked how likely they would use this new benchmarking tool over existing benchmarks, in which they responded with an average of 3 out of 5 stars.

5. Out of 5, how likely are you to use this new benchmarking tool over existing tools?

[More Details](#)

[Insights](#)

20

Responses



3.00 Average Rating

Their reasons for this are similar to the previous section when asked how likely they are to use the new tool. For example, they would definitely use it if the bugs are urgently fixed and refinements are made to the system to make it a better and more fluid experience.

6. Please explain your answer to the last question

[More Details](#)

[Insights](#)

20

Responses

Latest Responses

"Once bug fixed, definitely "

"I'd rather use existing tools as this others very little unique"

"Would have been nice if the benchmark provided data over time lik...

12 respondents (60%) answered **bugs** for this question.

fixed most definitely
happily use
good
day to day
polish
urgent bug
tool
good benchmark
benchmark
certainly use
decent benchmark

bugs

When asked if they find the benchmark data helpful, 17 out of 20 stated that they found it helpful while only 3 out of 20 stated they did not find it help.

7. Did you find the benchmark data at the end helpful?

[More Details](#)

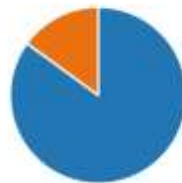
[Insights](#)

Yes

17

No

3



When asked for more information about why they stated either yes or no, they stated that it is always nice to have as much data as possible that can be looked upon to find issues with the hardware. However, some did state that it is not that different from other tools regarding the sensor data, while they did like the inclusion of the summary at the end that points out issues with suggestions.

Some also did state that they would have liked to have sensor data over time and the mins, maxes, and averages of each sensor.

8. Please explain your answer to the last question

[More Details](#)

[Insights](#)

20

Responses

Latest Responses

"Always nice to have data to go back and look on"

"Yes but isn't much different from existing tools"

"Always nice to have data in any capacity "

10 respondents (50%) answered **nice** for this question.

A word cloud visualization of responses for question 8. The word "nice" is the most prominent, appearing in the center in a large, bold, blue font. Other words include "information from the sensors", "easy", "information as possible", "graphs", "lot", "graph format", "PC hardware", "information about the hardware", "issues or shortcomings", "benchmarks", "summary", "sensor data", "no data", "data is very helpful", "hardware", "nice to have data", "hardware sensor", and "nice to have more information". The words are arranged in a circular pattern around the central "nice" word, with varying sizes and colors (mostly blue and green).

After that, the users were then asked if they found any issues while testing the new system, which all 20 did.

9. Did you encounter any issues with the new benchmarking tool?

[More Details](#)

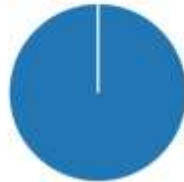
[Insights](#)

Yes

20

No

0



The issue, more so than issues found by the users, was that the system would crash in the first launch of the 3D demo. After conducting some testing and finding this out, the issue has come down to it being a pre-release of Unity that has broken something within its Vulkan implementation.

A workaround for this, until Unity fixes it, is to go back to DX12. The downside of going back to DX12 was that it was less efficient at Ray Tracing vs Vulkan, as it would lower performance while using a good amount more VRAM vs Vulkan.

10. If so, what issues did you encounter?

[More Details](#)

[Insights](#)

20

Responses

Latest Responses

"There is an issue where it crashes in the first launch"

"It crashes on the first run"

"There was a crash when when running it for the first time"

9 respondents (45%) answered **Crashes on first launch** for this question.

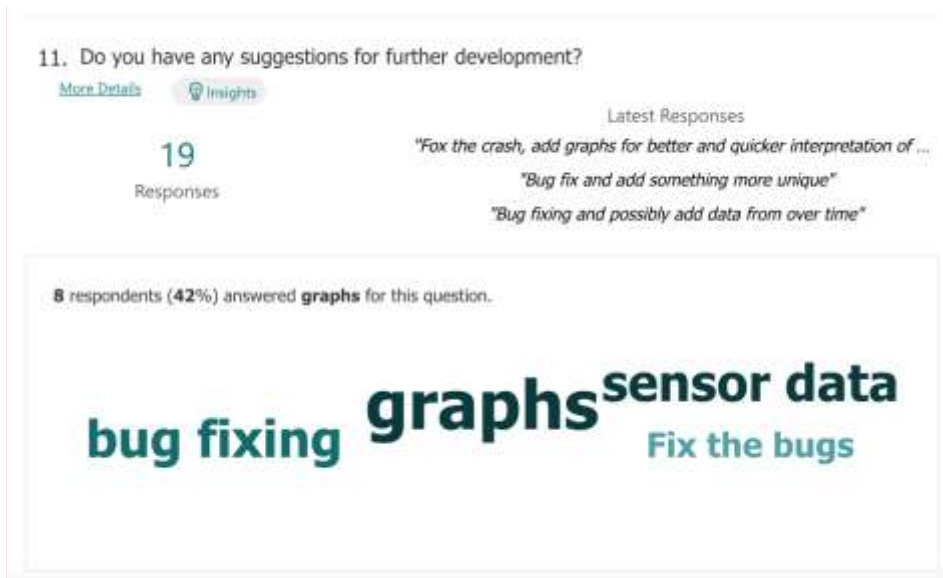
fine

3D demo

Crashes on first launch

crashes on the first run

When asked about future improvements and development to the system, the users responded that they would like the bugs with the system to be fixed, for example, the initial crashing on the first run. In addition, they would also like there to be sensor data over time presented to them and more use of graphs for quicker and more straightforward analysis of the data presented. One note was added to the end but was also a suggestion just covered.



The full results of the questionnaire can be seen in Appendix C.

9.0 Evaluation

9.1 Development

There are many areas of the development that went fully to plan, but many did not go fully to plan and had to be altered to get the right functionality from the system.

Firstly, the development was able to meet the initial requirements of benchmarking the user's hardware and showing results like other benchmarking tools such as Unigine and 3DMark. In addition to that, providing a summary of the computer's performance in terms of hardware, for example, showing bottlenecks with the GPU and suggesting ways to minimise the bottleneck.

One area in which the original plan changed was how the benchmark results were presented. Instead of being a portable HTML file getting the information from a text file, it has now been changed into a portable text file that the application can load instead. This was changed due to time constraints, as this would have been overly complex to get right. One reason for this was that it would work similar to what the system does right now without using an HTML webpage. Another reason would be to show the results. Time would have been taken to design and develop a webpage.

The 3D demo created within Unity had both its strengths and weaknesses in how it had been designed. When designed, it was not considered that Unity could not read the sensor data for the CPU resulting in the main application having to pick up where this failed. This added a little more complexity to and deviation from the proposed design but was simple to get up and to run. This made it slightly more

complex as the main program now needed to check if the 3D demo was still running and get the sensor data at the same time.

If this were to be carried out again, many changes would be made based on current knowledge and understanding gained from the development. Firstly, the main application would gather all the sensor information instead of both the 3D demo and the main application doing it simultaneously. This could be done for many reasons, such as making the system more efficient and less complex as the sensor API would only have to be called on one application instead of two.

In addition, another change would be made regarding how the data is presented to the end-user. The changes made here would also be to take advantage of more graphs, so the information is better presented to the user in an easier to understand format. This has come from both first-hand experiences when testing the application and feedback provided in the primary research.

9.2 Primary Research

The primary research methods served well throughout the process of the project. This was because many areas went as planned.

For example, the primary research effectively helped get the required information from many different people across many different time zones. Thus, allowing for more time to be spent on the secondary and final development while retaining the same level of knowledge. The same knowledge, if not more than what would have been gained through other methods such as interviews which may have caused time issues given the different time zones everyone was living in.

In addition, when providing the final solution to the end-users, the primary research gained from that allowed for weaknesses and potential to be found, allowing for action to be taken to address those issues. An example of this is the crashing when using Vulkan instead of DX12, with the pre-release version of Unity being used at the time.

If changes were to be made in the future, one to one interviews/observations would have also been done while testing the new benchmarking tool to witness any of the downfalls or struggles of the system while being used by someone else. This could have also been used to gain more helpful feedback at the cost of the time this would have taken to conduct.

10.0 Conclusions and recommendations

10.1 Conclusions

Many conclusions could be made from the findings of this report. One conclusion is that there is room for possible improvement within the benchmarking software industry, with either new tools or improvements to existing tools to take advantage of new and emerging technologies in the software/API space and the hardware space. This was found out from the first initial questionnaire asking for people's opinions on existing tools that started off the research into this endeavour.

Another conclusion that can be drawn is that the new system is a marginal success. This can be concluded from looking at the data provided in the primary research conducted on the new system with test users. It was found that the users like the system, but it does require some extra work to meet their usability requirements.

Furthering that point, conclusions can be made on the improvements, such as adding graphs to the results menu to improve the usability of the GUI, allowing for easier understanding and interpretation of the data. Another improvement would be to constantly collect and store hardware sensor data throughout the benchmark and display this information on the results menu, allowing the user to see the changes throughout the benchmark in hardware utilisation and other statistics such as hardware frequencies.

10.2 Recommendations

If the project were to be recreated or continued, several suggestions and recommendations could be made to improve this project's current state further.

A straightforward area would be to create a fully custom Unity scene rather than relying on assets from the asset store. This would allow the project to become open source as there would be no licences attached to the assets used within the unity scene, which would stop this. In addition, it would allow for better use and optimisation of real-time ray tracing within Unity.

Another recommendation/suggestion would be to use more primary research methods to gather more information, which allows for more analysis and feedback, which could help further improve upon what has already been found and areas in which the project could go next.

In addition, another area that should be pursued is making the final solution provide a portable HTML file that is time-stamped. The current solution is halfway there with the text file, but there was not enough time to create the portable HTML file. Doing this would allow the user to save the results in an easy-to-read format which can then be compared to other results from the benchmark from the same or a different PC.

Furthermore, adding more test runs and a dedicated CPU benchmark run to the solution would be better than currently provided. This would also allow the solution to be on the same feature level as what 3DMark currently provides.

11.0 Reflection

11.1 Development

Throughout the development, many elements went to plan. For example, the use of unity store assets allowed for quicker development of the Unity demo as only the GUI interface showing sensor data and the script needed to get that data and write it to a text file after were created manually.

In addition, the use of C# WinForms allows for a further speed increase in the development thanks to prior knowledge of the APIs and tools provided by Visual Studio. In addition, the plugin store that Visual Studio provides allowed for easy access to further APIs and tools used within the project, such as the graphs shown on the main menu and results menu.

OpenHardwareMonitor also allowed for greater compatibility across many different hardware vendors, CPU vendors and GPU vendors, while also speeding up the development as no API had to be created manually to gather sensor data.

OpenHardwareMonitor was used even though another API had been worked on during the development of this project and could have been used called RyzenAdj but would have only worked on

AMD systems that used AMD APUs. Although, RyzenAdj could be further modified into a custom version in the future to work on other hardware to detect sensor data fully.

Although many areas of the development went according to plan, many areas did not and caused a slight delay in the project's development.

One area was that OpenHardwareMonitor did not fully work within Unity resulting in research being carried out to get the C# project to detect the Unity demo to use itself to record the CPU sensor data to try and make up for where the Unity demo failed and then append that data to the text file the Unity demo creates once the demo had ended.

Another issue came when making the main menu user control communicate with the result menu user control. This was because when designing the new system, an oversight was made. That oversight is that a user control cannot effectively communicate with the main form to trigger a change to new user control.

This research was done to see if it would be possible to set up communication between the main form and the user controls that would be effective and would not cause any further complications, but no success was made. Instead, a new form was created that could be opened to display the result menu user control when the benchmark had ended.

In addition, further issues occurred importing all the Unity assets into the Unity project. The assets were not compatible with the Unity version being used, resulting in time spent fixing all the meshes, materials, and shaders individually to make them appear appropriately within the Unity scene.

There were also issues with asset placements within the Unity scene, with meant that time was also spent making sure that all the assets were positioned in a reasonable manner. For example, some of the water planes would not meet with the edge of the river/stream bank resulting in water randomly cutting off and floating in mid-air.

If the project were to be conducted again or further development time was to be spent on what already exists, it could be done differently and/or improved.

One way would be to do all the sensor data collection in the C# application rather than have both do it simultaneously. This would be to save on development time if done again and to make the system more efficient on the user's device, as currently the OpenHardwareMonitor API itself is being called twice with the same sensors and sensor data also being called at the same time when it could just be done once within the C# program.

In addition, if more time were allowed on the development, a whole new and entirely custom Unity demo scene would be created rather than relying on the ones included within owned Unity assets. Allowing the demo to be more customisable and unique to the new system would have also allowed the new system to be open source. The system would not depend on Unity store asset licenses.

11.2 Primary Research

Primary research was one critical area in the success of the project. Primary research allowed the project to find existing thoughts and feelings with existing benchmarking tools, allowing for areas of improvement to be found and acted upon with the development.

Primary research also helped get easy feedback when users reviewed the solution developed from this project. It allowed their thoughts on the solution to be expressed and allowed for issues to be found and acted upon. For example, fixing the crashing issue on the first launch.

The primary research method allowed this to be conducted as painlessly as possible with the many people across different time zones and locations, all of which had busy schedules that needed to be worked around, which questionnaires allowed with ease. Doing this allowed both parties to be suited as meetings did not have to be arranged, and the questionnaires could be done in their own time, making the information they provided less rushed.

There were downsides to using these research methods. For example, questionnaires are more locked down even when providing text fields to respond within. In addition, they lost the more personal side of, for example, interviews or one-on-one sessions, which would have allowed direct feedback to be observed and observing them using the system, which may have gained more information.

There would have been downsides to doing interviews and one to ones which questionnaires circumvented. For example, one-to-one interviews would have required more time to conduct. In addition, they may have put the user/s on edge, making their responses less genuine because they feel they may get, for example, judged.

References

1. AMD Radeon, "RDNA2" Instruction Set Architecture, https://developer.amd.com/wp-content/resources/RDNA2_Shader_ISA_November2020.pdf Date Accessed: 18/11/2021
2. NVIDIA, NVIDIA Tesla V100 GPU Architecture, <https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf> Date accessed: 18/11/2021
3. NVIDIA, NVIDIA Turing GPU Architecture, <https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf> Date Accessed: 22/11/2021
4. NVIDIA, NVIDIA Ampere GA102 GPU Architecture, <https://www.nvidia.com/content/dam/en-zz/Solutions/geforce/ampere/pdf/NVIDIA-ampere-GA102-GPU-Architecture-Whitepaper-V1.pdf> Date Accessed: 25/11/2021
5. R. Smith, Intel architecture day 2021: Intel unveils XeSS image upscaling technology, <https://www.anandtech.com/show/16896/intel-architecture-day-2021-intel-unveils-xeSS-image-upscaling-technology> Date Accessed: 30/11/2021
6. Intel, Intel Architecture Day 2021, <https://www.intel.com/content/www/us/en/newsroom/resources/press-kit-architecture-day-2021.html> Date Accessed: 30/11/2021
7. AMD Radeon, AMD FidelityFX Super Resolution, <https://github.com/GPUOpen-Effects/FidelityFX-FSR/raw/master/docs/FidelityFX-FSR-Overview-Integration.pdf> Date Accessed: 1/12/2021

8. N-iX, *Unity vs. Unreal: How to choose the best game engine*, https://medium.com/@N_iX/unity-vs-unreal-how-to-choose-the-best-game-engine-d3dbb4add73c Date Accessed: 1/12/2021
9. UL 3DMark, *3DMark Technical Guide*, [3dmark-technical-guide.pdf](#) Date accessed: 5/12/2021
10. Chas. Boyd, *DirectCompute in DirectX12: Innovations from the Game Space*, [DirectX12: Low Level Graphics \(gputechconf.com\)](#) Date accessed: 8/12/2021
11. Shawn, *Announcing DirectX 12 Ultimate*, [Announcing DirectX 12 Ultimate - DirectX Developer Blog \(microsoft.com\)](#) Date accessed: 10/12/2021
12. M. Bailey, *Introduction to the Vulkan Computer Graphics API*, [Microsoft PowerPoint - 00ABRIDGED-16x9.pptx \(oregonstate.edu\)](#) Date accessed: 13/12/2021
13. ASQ, *WHAT IS BENCHMARKING?*, <https://asq.org/quality-resources/benchmarking> Date accessed: 26/12/2021
14. I. Salian, *What Is AI Upscaling?*, <https://blogs.nvidia.com/blog/2020/02/03/what-is-ai-upscaling/> Date accessed: 31/12/2021
15. 15writers, *Understanding the Research Onion*, <https://15writers.com/research-onion/> Date accessed: 01/01/22
16. Dr H. Stainton, *The research onion for beginners*, <https://tourismteacher.com/research-onion-beginners/> Date accessed: 01/01/22
17. UKEssay, *Research Onion - Explanation of the Concept*, <https://www.ukessays.com/essays/psychology/explanation-of-the-concept-of-research-onion-psychology-essay.php> Date accessed: 01/01/2021
18. Blue Ocean Outsource Limited, *Dissertation Chapter 3: Research Methodology using the Saunders Research Onion*, https://www.linkedin.com/pulse/dissertation-chapter-3-research-methodology-using-1f?trk=organization-update-content_share-article Date accessed: 01/01/2021
19. K. Carbotte, *Unigine Releases Superposition Graphics Benchmark With 8K, VR Tests* <https://www.tomshardware.com/uk/news/unigine-superposition-graphics-benchmark-test,34121.html> Date accessed 02/01/2022
20. WhyCry, *Unigine launches Superposition GPU benchmark*, <https://videocardz.com/68354/unigine-launches-superposition-gpu-benchmark> Date accessed 02/01/2022
21. T. Sandhu, *AMD Ryzen Mobile 4000-series (Renoir) examined*, [AMD Ryzen Mobile 4000-series \(Renoir\) examined - CPU - News - HEXUS.net](#) Date of access: 13/12/2021
22. H. Heagedoom, *AMD Renoir Ryzen 7 4800U Gets Geekbench*, [AMD Renoir Ryzen 7 4800U Gets Geekbench \(guru3d.com\)](#) Date of access: 13/12/2021
23. TechTarget, *Spiral Model*, <https://searchsoftwarequality.techtarget.com/definition/spiral-model> Date of access: 29/12/2021
24. S. Kumar Pal, *Software Engineering | Spiral Model*, <https://www.geeksforgeeks.org/software-engineering-spiral-model/> Date of access: 13/12/2021
25. Unigine, *Superposition*, <https://benchmark.unigine.com/superposition> Date accessed: 03/01/2022
26. u/PRSMesa182, *is something wrong with my 9900k (inconsistent 3dmark CPU results)*, https://www.reddit.com/r/intel/comments/qdr38a/is_something_wrong_with_my_9900k_inconsistent/ Date accessed: 17/01/2022

27. Ksomoos, *Problem with 3dmark firestrike*, <https://www.overclock.net/threads/problem-with-3dmark-firestrike.1545158/> Data accessed: 17/01/2022
28. UL Benchmarks, *3DMark Port Royal — Real-time ray tracing benchmark — Full Demo 1440p*, <https://www.youtube.com/watch?v=QX0VbzXs548> Data accessed: 18/01/2022
29. D. Meister, S. Ogaki, et al, *A survey on Bounding Volume Hierarchies for Ray Tracing*, https://meister.github.io/publications/bvh_star/paper.pdf Data accessed: 20/01/2022

Appendix

A: Benchmark complaints

**Schporgle** 03/08/2021
10100 beats 3300X clock for clock
in CPU Profile
3300X wins in everything else, sometimes by big margins
as it should, since zen2 has higher IPC than skylake

**Nexus** 03/08/2021
Even with same memory settings?

**Schporgle** 03/08/2021
ye
my 10100 can't do what my 3300X could but matching them nothing changes
I sold 3300X so I can't retest it anymore, but even dropping 10100 to 2666 it still wins
vs 3300X's 3800 scores
which shouldn't be possible in a well-constructed benchmark
so the conclusion, CPU Profile is not well-constructed

**Schporgle** Today at 15:39
1080 medium is a CPU benchmark, and in general superposition favors Nvidia, though not as badly as heaven and valley do

**Mr. Sask Noodles** Today at 15:39
It also punishes AMD cpus

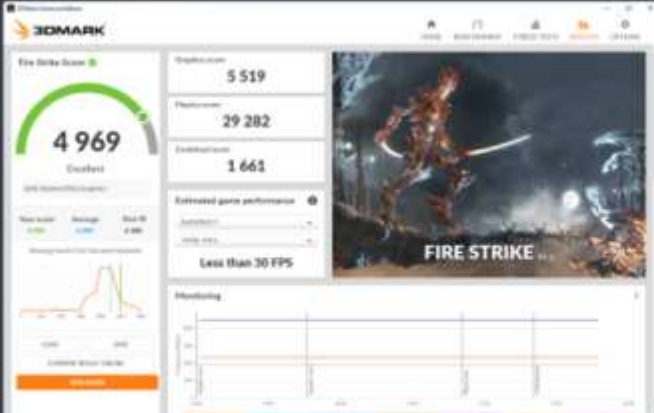
**Unoriginal Sin** Today at 15:40
Wild life prevents multi-gpu from working, even with force SLI/crossfire

**Unoriginal Sin** Today at 15:41
Night Raid CPU test maxes at 8 threads

 **Mr. Sask Noodles** Today at 16:01
The Superposition records on HWBot are all Intel

 **Unoriginal Sin** Today at 15:43
And time spy's CPU test (not tse that's fine) has insane run to run variance

 **Unoriginal Sin** 14/12/2021
3800 sure [redacted] works



The screenshot shows the 3DMark Fire Strike benchmark results. The main score is 4969. Sub-scores are: Graphics score 5519, Physics score 29282, and Combined score 1661. Estimated game performance is listed as 'Less than 30 FPS'. A 'FIRE STRIKE' game scene is visible in the background. Below the scores are graphs for 'Max FPS' and 'Min FPS'.

wow
as usual the combined score takes no cues from reality

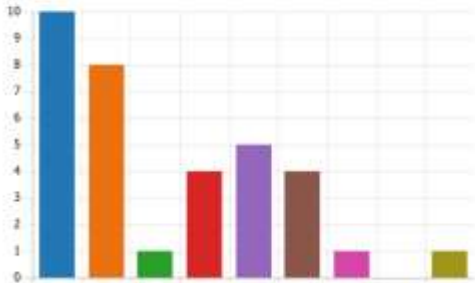
 **Absolute [redacted] Walnut** 14/12/2021
I did notice that combined scores was a metric they pulled from their [redacted]

B: Initial survey findings

1. What current 3D graphics benchmarks do you use to test your hardware? (Select all that apply)

[More Details](#)

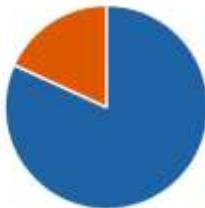
3DMark - Timespy	10
3DMark - Firestrike	8
3DMark - Wildlife	1
3DMark - Port Royal	4
Unigine - Superposition	5
Unigine Heaven	4
Unigine Valley	1
Other	0
Other	1



2. Do you like the benchmark/s you have chosen?

[More Details](#)

Yes	9
No	2



3. Please state why you like/dislike the benchmark/s chosen?

11 Responses

ID ↑	Name	Responses
1	anonymous	Good coverage of a variety of graphics scenarios and technologies with consistent scoring
2	anonymous	They're all quick and easy to compare stock vs OC'd results
3	anonymous	They're relatively unbiased, compared to other benchmarks, and have a large amount of online results available for system comparisons.
4	anonymous	They allow you to easily identify the performance differences between different hardware as well as OC vs non OC
5	anonymous	Unigine Heaven, I just used it ever since I got into tech, and Time Spy when I need something very intensive.
6	anonymous	They seem every limiting and don't have alot that can be changes about them that really matters

3. Please state why you like/dislike the benchmark/s chosen?

11 Responses

		between different hardware as well as OC vs non OC
5	anonymous	Unigine Heaven, I just used it ever since I got into tech, and Time Spy when I need something very intensive.
6	anonymous	They seem every limiting and don't have alot that can be changes about them that really matters
7	anonymous	Numbers go brr
8	anonymous	Either have a performance bias towards certain hardware or haven't been updated in a long time
9	anonymous	Widely used and widely compatible, easier to find results for comparison.
10	anonymous	They work and they are commonly used
11	anonymous	Timespy is neat for comparing performance between different GPUs/APUs cause its close-ish to the actual pecking order

4. Do you believe these benchmark/s could be developed further?

[More Details](#)

[Insights](#)

Yes	10
No	1



5. Please state why you believe that they do/do not need further development?

11 Responses

ID ↑	Name	Responses
1	anonymous	New graphical technologies are being introduced over time and the benchmarks are starting to fall behind
2	anonymous	There's always room to improve when it comes to scenarios tested, optimization, support for upcoming hw, etc.
3	anonymous	Something interesting would be low-load synthetic raytracing tests compared to the heavy raytracing tests available now.
4	anonymous	Timespy and Firestrike both have some degree of a performance bias towards hardware from AMD or NVidia depending on the one in question making it harder to compare hardware
5	anonymous	I think Heaven should utilize DX12.
6	anonymous	Afaik both are tools that aren't being updated anymore especially heaven

5. Please state why you believe that they do/do not need further development?

11 Responses

5	anonymous	I think Heaven should utilize DX12.
6	anonymous	Afaik both are tools that aren't being updated anymore especially heaven
7	anonymous	Everything can be improved
8	anonymous	Older benchmarks would be nice to see them be upgraded to DX12/Vulkan, would also be nice to have a light weight Ray Tracing benchmark instead of a heavy one like Port Royal
9	anonymous	They do need further development just like how game graphics are keep improving and new hardware being released which need optimization, it's a "benchmark for DX11 and DX12 games".
10	anonymous	They need to be closer to games in terms of the CPU and what the CPU stalls on
11	anonymous	Make it closer to the actual pecking order (its a bit NV biased)

C: User feedback survey findings

1. Out of 5, how easy to use is the GUI?

[More Details](#)

[Insights](#)

20
Responses

★★★★☆
4.25 Average Rating

ID ↑	Name	Responses
1	anonymous	Easy to run, easy to interpret the data
2	anonymous	Everything is straightforward and easy to understand
3	anonymous	GUI is straight forward and easy to use thanks to it being somewhat like exists benchmarks
4	anonymous	Easy to use, understand, and learn as it takes design cues from existing tools
5	anonymous	Easy to use as it has a similar design to other benchmarks
6	anonymous	Like other tools but could use a better colour pallet as colors currently do not match
7	anonymous	Good GUI design that is not over the top and uses design cues form existing benchmarks to make the transition more seamless

8	anonymous	Similar to other GUI designs for other tools making it easy to use
9	anonymous	simple and easy to understand
10	anonymous	nice and simple to use with a very small learning curve
11	anonymous	Easy to use and navigate
12	anonymous	Very easy to use and feels similar to existing tools
13	anonymous	Simple and versatile like existing benchmarks
14	anonymous	Easy to use and understand with basically no learning curve thanks to how alike it is to existing tools
15	anonymous	Easy to use like other tools I use on a daily basis when testing hardware
16	anonymous	Very simple and easy to use

17	anonymous	Versatile and easy to use
18	anonymous	Doesn't look that pleasing to the eye, could use a new colour palette
19	anonymous	Easy to use and is clear
20	anonymous	Nice and simple making it easy to use and learn

3. Out of 5, how much do you like this new 3D benchmarking tool?

[More Details](#)

[Insights](#)

20

Responses



3.60 Average Rating

ID ↑	Name	Responses
1	anonymous	Interesting benchmark, could use more tuning on the rendering front
2	anonymous	Needs a little refinement with some graphical elements such as water
3	anonymous	It provides a lot of useful information that may have been lost with other benchmarks
4	anonymous	It's like Unigine Valley and 3DMark TimeSpy/Firestrike had a baby
5	anonymous	Needs some bug fixing but is nice to-use otherwise with the information that it provides which cannot be found in most other tools
6	anonymous	It's nice to use and feels like a mix of Unigine and 3DMark
7	anonymous	I like the amount of detail this benchmark provides as well as how it feels like Unigine and 3DMark merged together
8	anonymous	almost as if Unigine and 3DMark mixed into one
9	anonymous	it's very similar to 3DMark and Unigine if they were mixed into one but needs some work to get it up to those standards
10	anonymous	Needs some work but is a nice concept of mixing Unigine and 3DMark
11	anonymous	Feels like a mix of 3DMark and Unigine
12	anonymous	It's a unique mix of 3DMark and Unigine
13	anonymous	It's a nice mix of unigine and 3dmark
14	anonymous	Needs urgent bug fixing so it does not crash so often

15	anonymous	It's a nice and simple mix of unigine and 3DMark
16	anonymous	It's a nice little benchmark that helps a lot when finding hardware issues
17	anonymous	It's very similar to existing tools and offers little that is unique
18	anonymous	It's ok in its current state but could be improved to be more useful
19	anonymous	Would have liked something more unique
20	anonymous	It's a nice and much needed mix of Ungine and 3DMark

5. Out of 5, how likely are you to use this new benchmarking tool over existing tools?

[More Details](#)

[Insights](#)

20

Responses



3.00 Average Rating

ID ↑	Name	Responses
1	anonymous	Seems like a decent benchmark, needs a bit more polish and bugfixing
2	anonymous	It is buggy and needs refinement but once fixed will happily use it
3	anonymous	It needs some polishing off to minimise the amount of bugs but it is a good first attempt
4	anonymous	A few bugs need to be fixed but i would certainly use it once fixed
5	anonymous	Will recommend it more once the bugs are fixed
6	anonymous	I would use this more when testing once bugs are fixed and the tool becomes more stable
7	anonymous	Crashes on the first go so can not recommend it or use it myself on a day to day basis

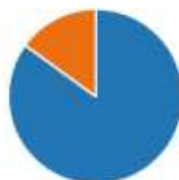
8	anonymous	There are a few bugs that need to be ironed out but would use after that has been done
9	anonymous	it needs some further development to get it to the point I would use it
10	anonymous	Like said, it needs some work but once issues are fixed I would happily use it
11	anonymous	It's a good benchmark in what it provides to the user but needs polishing off
12	anonymous	Needs urgent bug fixing otherwise good
13	anonymous	After bugs are fixed I would certainly use it in the future
14	anonymous	Once bugs are fixed I'd be happy using it
15	anonymous	Once bugs are fixed most definitely

16	anonymous	For the bugs and I would love to use it
17	anonymous	I might once bug fixed
18	anonymous	Would have been nice if the benchmark provided data over time like 3DMark
19	anonymous	I'd rather use existing tools as this others very little unique
20	anonymous	Once bug fixed, definitely

7. Did you find the benchmark data at the end helpful?

[More Details](#)

[Insights](#)



ID ↑	Name	Responses
1	anonymous	Well detailed, includes score for quick comparing
2	anonymous	Provided a lot of detail and liked the summary at the end
3	anonymous	It's easy to understand and display in a useful manner unlike other benchmarks
4	anonymous	Could use some graphs in addition to what is provided but other than that it help a lot showing the averages, mins and maxes as well as if there were any bottlenecks
5	anonymous	Would be nice to see graphs as well as just solid numbers
6	anonymous	More than what other benchmarks provide but would be nice if there were graphs as well

7	anonymous	The data is very helpful in identifying issues or shortcomings with PC hardware, would be nice if graphs were also used
8	anonymous	It's very detailed and I like how it analysis's the sensor data after to find issues
9	anonymous	Did not find it that helpful as it had no data showing full sensor data over the full time of the benchmarking run
10	anonymous	Not really easy to view, could have been better in a graph format perhaps
11	anonymous	Always nice to have as much information as possible
12	anonymous	Nice to have all the information from the sensors
13	anonymous	Always nice to have as much information as possible at your disposal when debugging hardware

14	anonymous	Nice to see so much information about the hardware
15	anonymous	Also nice to have more information with these tools
16	anonymous	Always good to see hardware sensor data and a little summary on the computers performance
17	anonymous	Not really that useful to me
18	anonymous	Always nice to have data in any capacity
19	anonymous	Yes but isn't much different from existing tools
20	anonymous	Always nice to have data to go back and look on

9. Did you encounter any issues with the new benchmarking tool?

[More Details](#)

[Insights](#)



ID ↑	Name	Responses
1	anonymous	A single crash on the first run, lots of pop-in, obvious asset streaming, screenspace artifacts, a few stutters on the non-fsr version.
2	anonymous	It crashed on the first run
3	anonymous	It did crash on the first launch but was fine after that
4	anonymous	Crashes on the first go but is fine after that
5	anonymous	Crashes on first launch but works fine after that
6	anonymous	Crashes on the first run of the benchmark but is otherwise fin after that from my testing
7	anonymous	Crashes on the first launch of the 3D demo but once that has happened it works fine

8	anonymous	I found that it crashes on the first run but otherwise its runs perfectly fine from what I observed
9	anonymous	I found that it crashes on the first run
10	anonymous	crashes on the first run
11	anonymous	Crashes in first launch of 3D demo
12	anonymous	Crashes when launching the forest demo for the first time
13	anonymous	Crashes on the first launch
14	anonymous	Has a crashing issue on first launch of the 3D demo
15	anonymous	Crashes on first launch
16	anonymous	It's a little crashy on first sun h if the benchmark
17	anonymous	Crashes on first launch
18	anonymous	There was a crash when when running it for the first time
19	anonymous	it crashes on the first run
20	anonymous	There is an issue where it crashes in the first launch

11. Do you have any suggestions for further development?

19 Responses

ID ↑	Name	Responses
1	anonymous	Fix the bugs to make it more refined and possibly make the benchmark a bit longer so more detail can be collected
2	anonymous	Add some graphs to show sensor data over time as well as the min, max and averages. In addition, fix the crashing issue.
3	anonymous	Graphs to show sensor data over time would be nice and general bug fixing to make it more useable
4	anonymous	Bug fixing + adding graphs to make the data even more easy to understand and interpret
5	anonymous	most important is bug fixing with the addition of graphs
6	anonymous	would be nice if bugs were fixed and maybe graphs introduced to better display the sensor data gathered by the tool

7	anonymous	Would be nice for an online database to store users scores and sensor data for comparison with other users
8	anonymous	fix the bug where it crashes and add what i stated was missing above
9	anonymous	Fix the crashing issue, add more graphs for easier understanding of data, and possibly create a results website so users can share their results
10	anonymous	Fix bugs!!!!
11	anonymous	Urgent bug fixing and possibly a better way to display sensor data
12	anonymous	Bug fixing is much needed
13	anonymous	Obviously bug fixing!!!! and possibly add more graphs where possible shen showing the user data

14	anonymous	Just general bug fixing and usability improvements from graphs
15	anonymous	Nothing other than bug fixing that crashing issue
16	anonymous	Fix the bugs with crashing
17	anonymous	Bug fixing and possibly add data from over time
18	anonymous	Bug fix and add something more unique
19	anonymous	Fix the crash, add graphs for better and quicker interpretation of data, and add sensor data over the course of the whole run

12. Any further notes or suggestions?

1 Responses

ID ↑	Name	Responses
1	anonymous	Once the above suggestions are done I would like to see it maybe have 2 graphical runs and then produce the report

D: Functionality testing

What is being tested?	Expected outcome	Actual Outcome	Action needed?
-----------------------	------------------	----------------	----------------

Testing to see if the main menu form opens without issue and objects within display properly	The main menu form opens with all objects displaying correctly	Worked as expected	N/A
--	--	--------------------	-----

Evidence:



What is being tested?	Expected outcome	Actual Outcome	Action needed?
Testing to see if pressing either run button opens the correct benchmark.	The correct benchmark opens when pressing either button.	Worked as expected	N/A

Evidence:



What is being tested?	Expected outcome	Actual Outcome	Action needed?
-----------------------	------------------	----------------	----------------

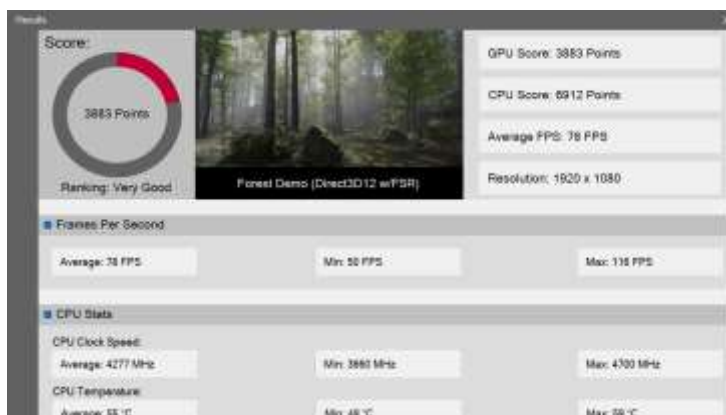
Testing to see if the benchmark saves results to a text file once finished.	The benchmark should save the results file once the benchmark is over.	Worked as expected	N/A
---	--	--------------------	-----

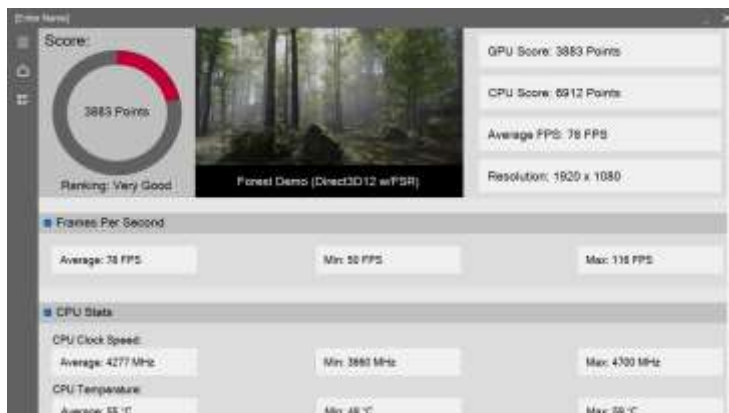
Evidence:

Managed	2/14/2022 5:18 PM	File folder	
Plugins	2/14/2022 5:19 PM	File folder	
Resources	2/14/2022 5:18 PM	File folder	
app.info	2/15/2022 10:08 PM	INFO File	1 KB
boot.config	2/27/2022 2:12 PM	CONFIG File	1 KB
data.unity3d	2/27/2022 5:09 PM	UNITY3D File	1,138,076 KB
results	2/28/2022 10:09 PM	Text Document	1 KB
RuntimeInitializeOnLoads	2/15/2022 10:08 PM	JSON Source File	3 KB
ScriptingAssemblies	2/15/2022 10:08 PM	JSON Source File	4 KB

What is being tested?	Expected outcome	Actual Outcome	Action needed?
Testing to see if the results menu opens and displays the results from the result text file	The results menu should open with the correct result displayed	Worked as expected	N/A

Evidence:





E: Consumer GPUs that support Ray Tracing

Any GPU that supports the DX12 Ultimate API will support ray tracing. The following has been taken from the TechPowerUp GPU database. DX12 Ultimate was explained in the graphics API's section.

E.1 Nvidia Desktop GPUs

GPU Name	Architecture	Source
RTX 2060	Turing	https://www.techpowerup.com/gpu-specs/geforce-rtx-2060.c3310
RTX 2060 Super	Turing	https://www.techpowerup.com/gpu-specs/geforce-rtx-2060-super.c3441
RTX 2070	Turing	https://www.techpowerup.com/gpu-specs/geforce-rtx-2070.c3252
RTX 2070 Super	Turing	https://www.techpowerup.com/gpu-specs/geforce-rtx-2070-super.c3440
RTX 2080	Turing	https://www.techpowerup.com/gpu-specs/geforce-rtx-2080.c3224
RTX 2080 Super	Turing	https://www.techpowerup.com/gpu-specs/geforce-rtx-2080-super.c3439
RTX 2080Ti	Turing	https://www.techpowerup.com/gpu-specs/geforce-rtx-2080-ti.c3305
RTX 3060	Ampere	https://www.techpowerup.com/gpu-specs/geforce-rtx-3060.c3682
RTX 3060Ti	Ampere	https://www.techpowerup.com/gpu-specs/geforce-rtx-3060-ti.c3681
RTX 3070	Ampere	https://www.techpowerup.com/gpu-specs/geforce-rtx-3070.c3674
RTX 3070Ti	Ampere	https://www.techpowerup.com/gpu-specs/geforce-rtx-3070-ti.c3675
RTX 3080	Ampere	https://www.techpowerup.com/gpu-specs/geforce-rtx-3080.c3621

RTX 3080Ti	Ampere	https://www.techpowerup.com/gpu-specs/geforce-rtx-3080-ti.c3735
RTX 3090	Ampere	https://www.techpowerup.com/gpu-specs/geforce-rtx-3090.c3622

E.2 AMD Radeon Desktop GPUs

GPU Name	Architecture	Source
RX 6400	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6400.c3813
RX 6500XT	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6500-xt.c3850
RX 6600	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6600.c3696
RX 6600XT	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6600-xt.c3774
RX 6700XT	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6700-xt.c3695
RX 6800	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6800.c3713
RX 6800XT	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6800-xt.c3694
RX 6900XT	Navi II (RDNA2)	https://www.techpowerup.com/gpu-specs/radeon-rx-6900-xt.c3481