

Game Engines and Architecture: A Review of Key Platforms and Comparative Analysis for Novice and Professional Developers

Manit Saxena
CSE Scholar
Chandigarh University
Mohali, India
ManitHuMain@gmail.com

Sidakpreet Singh
CSE Scholar
Chandigarh University
Mohali, India
sidakpreet007@gmail.com

Kunal Tanwar
CSE Scholar
Chandigarh University
Mohali, India
kunaltanwar1308@gmail.com

Mohit Jangir
CSE Scholar
Chandigarh University
Mohali, India
adityamohitjangir0@gmail.com

Disha Sharma
Assistant Professor, CSE
Chandigarh University
Mohali, India
e13241@cumail.in

Abstract — Game engines have evolved from simple 2D platforms to complex 3D and VR-based platforms, which have brought a remarkable evolution to the game development industry. Since the emergence of game engines, real-time rendering, physics simulation, artificial intelligence, and procedural content generation have been constantly improving, enabling game developers to create more immersive, visually appealing, and challenging gaming experiences. This paper introduces a beginner-friendly and easy-to-start review of the evolution of game engines that address the most prominent platforms, such as Unity, Unreal Engine, CryEngine, Frostbite, Godot, and other prominent game engines that have shaped the gaming industry over the years. By incorporating the results of various previous studies of game engine polls, this paper aims to provide a concise and comparative review, taking into account the most important factors of usability, scalability, cross-platform development and performance, learning curve, and availability for developers of all levels, as well as several other factors that are relevant for beginners. A lot of focus is given to the layered game engine architecture, providing a basic understanding of how different pieces fit together to deliver smooth gaming experiences. In addition to assessing the most important technological advancements, we critically analyze the pros and cons of each platform and identify the limitations of the existing comparative studies of these platforms. The potential, performance, and acceptance of each platform for different development needs are analyzed to provide a balanced review for both beginners and professional developers. By breaking down complex concepts, this paper aims to make the evolution of game engines easy to understand for beginners and provide valuable tips on technological advancements, challenges, and general implications of these platforms.

Keywords — Game engines, architecture, Unity, Unreal Engine, CryEngine, procedural content generation, real-time rendering, cross-platform development.

I. INTRODUCTION

Games have always been an important part of human culture. Games have come a long way from being just board games to being high-end computer games. Video games have become an industry by themselves, where art, story, and technology come together to create rich virtual worlds. The virtual world is created by game engines, which are SDKs to create games. SDKs are software frameworks that provide essential tools to the game engine.

Game engines have been the base of game development. They have come a long way from being platform-specific, hardcoded game engines to being highly modular game engines like Unity, Unreal Engine, and CryEngine. Earlier, each game was a scratch-made game, where developers would write their own rendering code, physics code, and logic code. But with the introduction of game engines in the mid-90s, game development has become a process-oriented field, where programmers could concentrate on building content rather than low-level programming [1]. The game engines are not only used to make video games but are also used to make movies, architectural simulations, medical simulations, and virtual reality training [3].

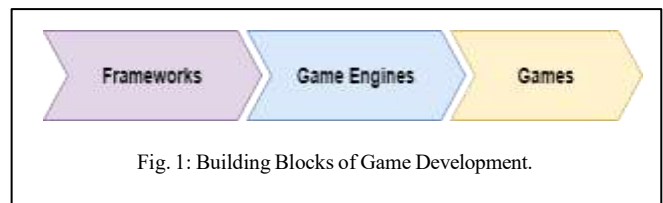


Fig. 1: Building Blocks of Game Development.

With all these advancements, game engines have continued to evolve with new features and functionalities that have improved their appearance, speed, rendering, and interaction. Even though these advancements have improved games significantly, they have also increased their complexity [6]. The increased use of real-time global illumination, AI-generated content, and physics-based rendering has changed the face of game development. However, with these changes, there have also been challenges such as increased computing requirements and increased complexity to learn.

The use of game engines in the gaming sector has now given way to their application in other sectors such as virtual production and simulation. Even though there are many research studies done on individual aspects, there are few that explain their evolution from their origins in 2D to the 3D and VR game engines that are seen today [5]. Currently, there are few accurate research studies done on a relevant and easy-to-start review of these.

This research aims to explore the issues involved and present a concise overview of the evolution of game engines over the years. It aims to make the data available to new stakeholders as well. The acquisition of new significant concepts and what this means to game design and beyond means that this research aims to fill in the gaps in our

knowledge with new insights into the technological evolution that has seen the game engines that are seen today.

A. What are Digital Games & Game Engines ?

In short, Games can be defined as a pattern of play with rules, objectives, and interactions. Today, games have become a source of fun, competition, and narration. The latest form of game development is video games, which utilize computer-based digital technology to create interactive digital worlds with the aid of advanced digital graphics, physics, and intelligence [11][13].

On the other hand, Game Engines can be defined as game development platforms. They provide the basic features of game development, which include REs, PEs, etc. They allow game developers to create interactive experiences without programming them from scratch. Today, game engines like Unity and Unreal Engine have brought game development to the mainstream, allowing anyone to develop games [1][11].

II. ARCHITECTURE OF GAME ENGINES

A game engine is a collection of systems and tools that are used collectively to develop and run a computer game. A game engine provides core features like graphics, physics, and AI so that game developers can focus on the game design rather than coding all of this from the ground up. The study of the following concepts is adopted from [2][4]. Below are the key components of a game engine:

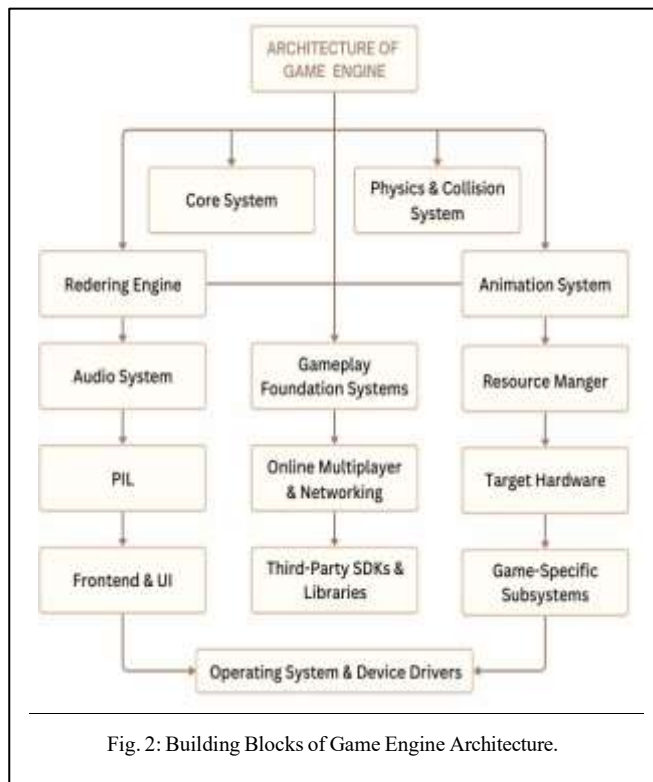


Fig. 2: Building Blocks of Game Engine Architecture.

1) *RE*: The RE is a subsystem that shows graphics on the screen in the form of 2D or 3D graphics. It performs operations such as image rendering, shading, and frame optimization to provide graphical output.

2) *CS*: CS is the foundation of a game engine. It provides basic functionality like memory management, math calculations, and string manipulation, which are necessary in order to efficiently manage complex game logic.

3) *PE & CE*: PE is responsible for simulating real-world physics like gravity, force, and collision between objects. CE detection ensures that objects do not act irrationally in the game environment so that there would be no irrational penetrations or overlaps in the real game.

4) *AS*: Character and world existence is the outcome of animation. 2D games use sprite animation, and 3D games use skeletal animation, in which the animation of a character is described by a mesh or a skeleton of bones.

5) *AuS*: Sound design is a critical part of the game immersion experience. The AuS is responsible for managing background music, character voice-overs, dialogues, and general environmental sounds that are a part of the player interaction.

6) *GFS*: This layer regulates the overall game mechanics, such as character controls, interaction, player goals, and AI behaviors. It also comprises scripting engines to simplify gameplay and programming.

7) *RM*: RM is a single interface for managing all game assets of the engine like textures, fonts, models, and animations. It optimizes the loading of assets, avoids lag, and reduces memory usage.

8) *TH*: Games can be developed to support various platforms like PCs, consoles, and mobiles. TH is responsible for ensuring compatibility with various OSs and hardware.

9) *PIL*: PIL provides abstraction from platform-specific details, allowing a game engine to run uniformly on various hardware platforms. It provides smooth adaptation to various platforms.

10) *Online Multiplayer & Networking*: Networking features enable real-time multiplayer gaming. It provides various multiplayer options like split-screen game play, client-server architecture, and the most popular MMOGs.

11) *Third-Party SDKs and Libraries*: Third-party SDKs and libraries provide existing functionalities like PE, AI modules, and UI libraries. They reduce development time and enhance the quality of the game.

12) *VE*: VEs improve the visual appeal of a game, incorporating features like particle systems, dynamic lighting, and shadow rendering to create realistic or artistic environments.

13) *Frontend & UI*: Frontend comprises all graphical components of the game like in-game menus, HUDs, and cinematics. It provides smooth interaction between the player and the game.

14) *OS & Device Drivers*: OS is responsible for controlling the execution of game processes, while device drivers enable interaction between hardware components like GPUs, input devices, and audio interfaces.

15) *GSS*: These are game-specific components, such as AI-controlled NPCs, advanced camera systems, customizable physics engines, and game-specific mechanics such as weapons and vehicles.

III. GAME ENGINES

Game engines have been studied by many researchers to understand how they work and how they have developed over the years. For this review, we have chosen research studies that define game engines in simple and beginner-friendly terms. The articles highlight important aspects such

as graphics, usability, platforms, and how effective they are in helping developers develop games [1].

The following sections will extract insights from different studies and provide a clear & concise overview on various game engines. This will enable beginners to be aware of their key features, pros, and cons so that it is simple to choose the correct engine to suit their needs.

A. Unity Engine

Unity is one of the most popular and easiest game engines in the world. It was first showcased in 2005 at Apple's Worldwide Developers Conference. Since its showcase, Unity has become a versatile platform for creating 2D as well as 3D games. One of its best features is that it is compatible with many platforms. Developers can create a game once and run it on more than 15 platforms, including Windows, macOS, Android, iOS, PlayStation, Xbox, Linux, and even big web browsers. This "build once, deploy everywhere" feature saves a lot of time and effort while making games [2].

Unity also possesses a good and intuitive development environment. It possesses a visual scene editor, which allows developers to build scenes using the drag-and-drop method, and an IDE, which allows developers to write code using C#. The engine also possesses a component-based framework, which means that all objects in the scene are a GameObject, and their functionality is controlled by add-on components [6]. This makes the code clean and easy to read, especially for new developers.

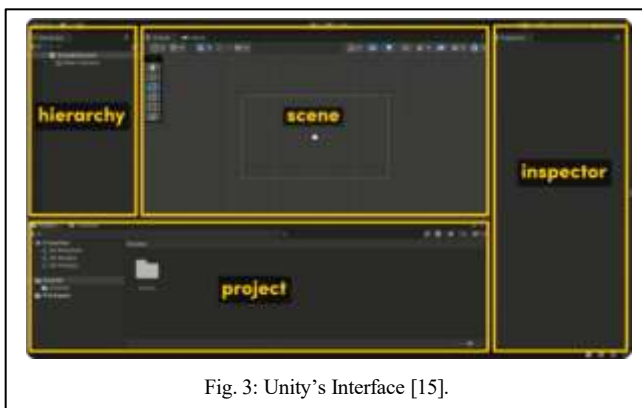


Fig. 3: Unity's Interface [15].

Unity is also very friendly to new developers because it possesses good documentation on its 2D development mode [2], good learning assets, and a good community for new developers. It also possesses sprite management, 2D tile mapping, and basic physics, making it easier to kick-start the development process for new developers. At the core, it uses various technologies such as NVIDIA PhysX to carry out physics simulations and also uses Mecanim to carry out animation blending and skeletal rigging. The real-time compilation of code using MonoDevelop or Visual Studio also provides instant feedback when developers test the code. The MLAPI is also another good feature that supports the implementation of basic online multiplayer features [4].

The next big advantage of Unity is its Asset Store. It offers thousands of assets such as scripts, models, templates, etc., most of which are free, while others require payment. These assets cut down the workload needed to create fully featured games significantly. Also, the community of Unity

users is huge, vibrant, and friendly, making debugging easy. The feature of HDRP rendering is one of the most recent additions to the Unity platform. It offers quality visualization in real-time through ray tracing and lighting. The URP adds solidity to the cross-platform rendering of images. Unity introduced the feature of DOTS, which relies on the principle of multithreading. It is now possible for small developers to access advanced technologies, which can enhance their productivity without having to learn these technologies themselves [15].

However, Unity also has some drawbacks. Although Unity can create high-end graphics, it lags behind Unreal Engine or CryEngine in the graphics details that can be achieved [2]. Also, Unity is not very suitable for ultra-realistic games or open-world games because of the intermittent performance of Unity in graphics-related scenes. Although Unity supports multiplayer games, it is not very good in terms of networking capabilities compared to other game engines, which might create problems in creating humongous multiplayer games [6]. Also, most of the extensions in the Asset Store require payment, which might create problems for less budgeted game developers [2].



Fig. 4: Rust [24].



Fig. 5: Fall Guys [25].

In conclusion, Unity can be said to provide a good balance between simplicity and power, which makes it highly suitable for game development. Although it might not be very suitable for creating ultra-realistic games or humongous multiplayer games, it can still be used for most game development scenarios.

Unity has been used to develop many successful games. Famous Unity-based video games include *Cities: Skylines*, *Temple Run*, *Rust*, *Among Us*, *Fall Guys*, and *Call of Duty: Mobile*. The official access link of Unity can be found at [14], along with the official Cloud link of Unity at [15].

B. Unreal Engine

Unreal Engine, developed by Epic Games, is the most powerful and most stunning game engine available today. Since its first release in 1998, the engine has received many major updates. Unreal Engine 4 (UE4) is a major milestone in the democratization of professional tools [6]. Epic Games has recently launched Unreal Engine 5 (UE5) in 2022, which has further improved the capabilities of the engine using game-changing technologies such as Nanite and Lumen. Nanite is a virtualized geometry solution that enables artists to build film-quality assets, and Lumen enables full dynamic global illumination and reflections.

From different studies, it has been found that Unreal Engine is most famous for its stunning graphics and the availability of the engine on a wide variety of platforms, including PC, consoles, VR, AR, and mobile devices [2][6]. It supports Physically Based Rendering (PBR) where the materials and objects react to light as they would in the real world. It has the ability to leverage the power of GPUs on devices using the native graphics API DirectX 12, Vulkan, and Metal. Advanced rendering features such as ray tracing, subsurface scattering, screen space global illumination, and virtual shadow maps allow cinematic visuals to be used in real-time environments [1][2].



Fig. 6: PUBG [26].



Fig. 7: Black Myth Wukong [27].

Another strength of Unreal Engine is the dual scripting system it provides. This allows developers to create complex systems using C++ if they want to optimize performance, or they can use Blueprints Visual Scripting, which is a node-based system that does not require writing code to create complex gameplay systems. This allows both programmers and designers to work effectively with the engine. The Sequencer Editor supports in-engine cinematics with timeline-based animation, lighting, and camera control [6].

Unreal Engine provides NVIDIA PhysX for real-time physics, collision detection, destructible environments, and dynamic objects. Unreal Engine also provides Behaviour Trees and EQS for AI development, enabling characters to make context-driven decisions [2]. The engine also supports networked features such as server functionality, replication, and lag compensation, making it an ideal choice for competitive and cooperative multiplayer games such as *Fortnite*.

Unreal Engine 5 also provides large-scale world-building features such as the World Partition system, which divides massive maps into grid sections that load dynamically depending on the player's location. The engine also provides the MetaHuman Creator, which enables the fast creation of photorealistic human characters.

However, despite Unreal Engine's many strengths, it is difficult for beginners to use because of the complexity of the interface. Although Blueprints make the engine easier to use, it requires C++ knowledge to master the engine fully, and the level of detail in the scenes requires optimisation to run smoothly.

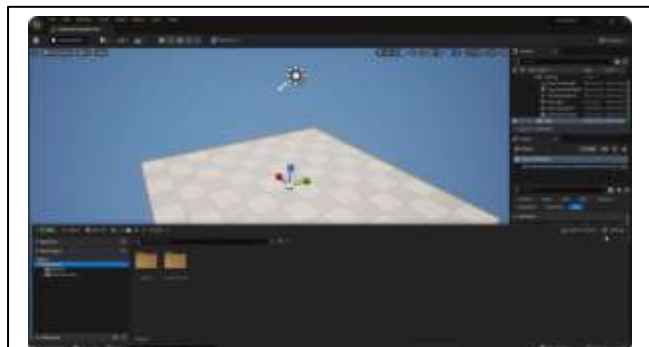


Fig. 8: Unreal's Interface [16].

In conclusion, Unreal Engine offers a rich and advanced environment to create high-quality, interactive content. The cutting-edge features of this game engine, combined with its flexibility, make it an advanced tool for creating realistic content. Even though the game engine is difficult to learn, its advanced features make it a powerful tool for anyone looking to create immersive, realistic content.

Unreal Engine is the go-to game engine for developers to create stunning visuals, advanced gameplay, and engaging stories. The game engine's application ranges from small independent games to realistic AAA title blockbusters. Some of the most popular games developed using Unreal Engine include *Fortnite*, *Batman Arkham Series*, *Borderlands 3*, *Final Fantasy Series*, *Ark: Survival Evolved*, *Black Myth Wukong*, *The Matrix Awakens Demo*, which demonstrates the true potential of Unreal Engine 5 in creating realistic cities in real-time. The official access link for Unreal Engine can be found at [16].

C. CryEngine

CryEngine, a game engine, was developed by Crytek. It was highly praised for its potential to produce stunning graphics. It was initially used to create games in the Far Cry series. It is now used as a tool to create quality FPS games and cinematic games. It is one of the most popular game engines for businesses that specialize in creating games with high levels of detail in terms of graphics and environments.

It was researched that CryEngine supports the development of games on Windows, PlayStation, Xbox, Linux, and VR platforms such as Oculus Rift and PSVR [4]. It also supports the writing of code in C++, Lua, or C#. In addition to this, Flow Graph, a visual language, can also be used to create game logic through nodes. Though this feature is useful, it is not as comprehensive as the Blueprint system in Unreal Engine and usually requires some programming knowledge [2].

What makes CryEngine stand out is its highly developed graphics rendering system. Its Environment Editor and Terrain Tools allow for the development of large, dynamic worlds with wind and weather effects. The game engine also boasts Voxel-Based Global Illumination for precise real-time lighting, real-time Volumetric Fog for increased realism, and Tessellation for more detail in 3D models when the camera gets closer.



Fig. 9: CryEngine's Interface [17].

CryPhysics is the PE of the CryEngine and is designed to assist the developer in the support of advanced physics-based simulations such as soft body physics, destructible terrain physics, fluid physics, and gravity-based physics for simulating objects. Physical expressions and character manners can be emulated using the Advanced Animation and Facial Rigging System of the CryEngine. The AI of the CryEngine using AI Territories & Waves, Behavior Trees, and EQS is very helpful in emulating the generation of intelligent non-player characters that will dynamically behave in accordance with the players and worlds they are in.

In the context of multiplayer engine development, the engine is server-authoritative and, although not as simple to use as the network API of the Unreal Engine, the addition of CryEngine Visual Scripting Enhancements and the ability to be used in conjunction with other industry-standard tools such as Autodesk Maya and Adobe Photoshop makes the engine a very efficient one.

However, the engine does have some issues for the novice developer. The interface is extremely daunting, and the documentation is not as comprehensive or updated as the Unity and Unreal engines [2]. In addition, the lack of tools and asset store items may be a drawback for the freelance developer. Despite the engine being royalty-based and free to use, the lack of a large community of users may be a drawback for the developer who wishes to utilize the collaborative and learning aspects of the engine.

In conclusion, CryEngine is a very powerful game engine that can be used to develop games that are visually stunning and have high-performance capabilities. Although it may be difficult for new users to understand and there may be a lack

of community resources, it can be fully utilized by experienced users to develop games that can be very immersive.



Fig. 10: Far Cry 6 [28].



Fig. 11: Crysis 3 [29].

Some of the visually stunning games that have been developed in the last few decades include Crysis Series, Prey, Ryse: Son of Rome, Far Cry Series, and Kingdom Come: Deliverance. The link to CryEngine's access can be found at [17].

D. Godot

Godot is an open-source and free game engine. Initially under the MIT license released in 2014, Godot is a relatively newer player compared to the Unity and Unreal engines. Nevertheless, the engine has quickly established a powerful reputation, especially among the indie gaming world, due to its ease of use, flexibility, and low resource utilization. It has especially been praised for its ability to run on low-end hardware, making it the first choice for individual developers and educational purposes [9].

One of the most beginner-friendly features of Godot is its node-based scene system. The node-based scene system allows a developer to create a tree-like structure for the different components of a video game, where each component serves a different purpose, such as rendering an image, handling input, and playing sounds, etc. The scripting language used in Godot is also very beginner friendly [8]. The scripting language used in Godot is GDScript, which is almost identical to Python and thus very easy to read and understand for a beginner programmer. Godot also supports C#, C++, and VisualScript, which is a node-based scripting language like Unreal's Blueprints. Godot also supports GDNative, which allows a programmer to link other programming languages like Rust, Kotlin, D, making it a great option for programmers of all skill levels [9].

Godot also provides both 2D and 3D PEs, enabling the user to choose between GodotPhysics and Bullet Physics, making it easier to optimize the performance of the project depending on the chosen platform. The latest updates also provide GPU-based light baking, real-time global illumination, and animation tools, making it comparable to commercial visual quality standards. Godot 4.0 is a significant achievement, with the RE being rewritten using the Vulkan API, making it faster and better in performance and lighting effects, especially in 3D scenes.



Fig. 12: Godot's Interface [18].

Another advantage of using Godot is that it supports all platforms, including Windows, macOS, Linux, Android, iOS, and HTML5, and even consoles with the right modules. Godot also supports multiplayer features using the client-server and peer-to-peer architectures, making it easier for the creation of online features, especially for small- and medium-scale games [8].

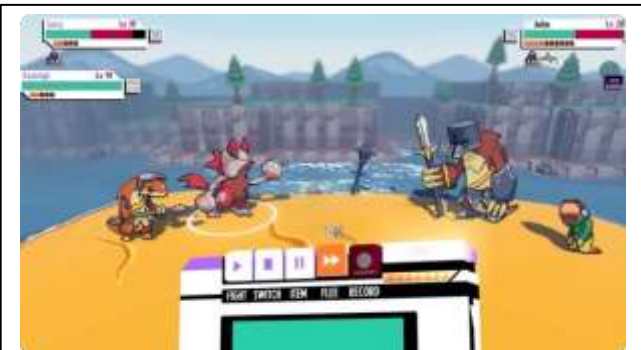


Fig. 13: Cassette Beasts [30].



Fig. 14: Dodgeball Academia [31].

Godot is also gaining popularity and is ranked second on itch.io [39]. The engine is also widely used in global jams such as the GMTK Game Jam, especially for 2D games. Although there is a slight drop in the popularity of Unity and

GameMaker Studio, Godot is gaining popularity among indie developers [9]. The engine also received funding from Microsoft for C#, Epic Games with a \$250,000 MegaGrant, and also from Meta for VR development.

Although the 3D engine in Godot has improved significantly, especially with the latest version 4.0, it is still not comparable to Unreal or CryEngine in the production of photorealistic graphics. The documentation and learning curve are also not comparable to the top-rated game engines such as Unity, but it is still an open-source and flexible platform and therefore cannot be ignored.

In conclusion, Godot can be said to stand out as a highly accessible game engine with considerable adaptability, especially among indie game developers and educators. Although it may not measure up to the high-performance graphics capabilities of game engines like Unreal, Godot is still a game engine with considerable promise. As a game development engine with considerable adaptability, Godot can be said to provide a reliable solution for game developers looking for considerable creative freedom with platform versatility.

Some of the prominent games developed with Godot that speak of the power of Godot in 2D game development, as well as its growing capabilities in 3D game development, especially among game developers looking for considerable creative freedom with platform versatility, include *Cruelty Squad*, *Lumencraft*, *Cassette Beats*, *Of Life And Land*, and *The Garden Path*. The official access link for Godot can be accessed at [18].

E. Amazon Lumberyard

Amazon Lumberyard is a free, proprietary 3D game engine developed by Amazon and released in 2016. Built from Crytek's CryEngine, it benefits from strong real-time graphics and rendering capabilities. Amazon later enhanced it with AWS and Twitch integration, making it well suited for online, multiplayer, and streaming-focused games [10].

Lumberyard retains CryEngine's advanced rendering and physics features, including high-quality visual effects, real-time environmental interaction, and cinematic lighting. Developers have full access to the source code for customization. It also incorporates Amazon Polly, a cloud-based text-to-speech service that can generate in-game voice lines dynamically, supporting storytelling, accessibility, and user-generated content [10].

For development tools, Lumberyard provides visual scripting, animation and particle editors, UI tools, and asset pipelines. With built-in AWS GameLift support, developers can host and scale dedicated multiplayer servers more easily. Additionally, Twitch ChatPlay enables viewers to interact with live games through chat, creating real-time engagement between streamers and audiences [10].

In the latest versions, this engine has moved a step further. In 2021, Amazon made an announcement regarding a shift to O3DE, which means a community-driven, open-source extension of the Lumberyard architecture, facilitated and governed by the Linux Foundation. This change comes after Amazon understood the benefits that come with an open development model, which include increased iterations and access to the engine. The new engine includes major advancements like modularity, photorealistic rendering

facilitated by the inclusion of the Atom Renderer, terrain tools, and the inclusion of contemporary workflows like version control, Python and Lua scripting, and DCC tools. [19].

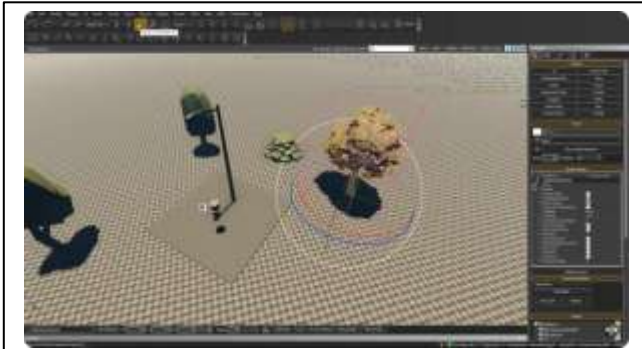


Fig. 15: Amazon Lumberyard's Interface [19].

Even though this engine has moved a step further, it still does not come without its limitations and disadvantages. The engine still remains difficult to learn due to a lack of tutorials and community support and due to the documentation being behind other engines like Unity and Unreal. The engine also does not come with solid support for other platforms like mobile and web, which can be a limitation for a developer looking to develop games for these platforms and reach a wider audience in the casual gaming market.



Fig. 16: The Grand Tour Game [32].

In conclusion, the Amazon Lumberyard, which is currently being replaced by O3DE, offers CryEngine-based rendering, AWS GameLift, and Twitch interactivity for the creation of games that can be played online, offer cloud gaming, and include procedurally generated content.

Some examples of games that have been made with this engine include *The Grand Tour Game* and *New World*, which is Amazon's best MMORPG. Though it may not be very popular compared to other game engines like Unity and Unreal, it can be said that it can be a very good option for those who need to include gamification, online engagement, and cloud gaming in their games. Amazon Lumberyard's official Cloud link can be found at [19], along with O3DE's official access link at [20].

F. GameMaker Studio

One of the best things about YoYo Games' GameMaker Studio, which is also characteristic of the game engine, is that it allows developers to create 2D games with an open and easy-to-use game engine. It was originally released as "Animo" back in 1991, but the game engine has since

become one of the easiest products available in the market for hobbyists, independent developers, and teachers. It is easy to learn how to make games with GameMaker Studio, but at the same time, one can make professional-level games with the game engine. The most characteristic aspect of GameMaker Studio is that the game engine has a dual-mode game development system. Developers can make games with DnD, which allows them to drop visual action blocks, which can be used to make game logic. This system is suitable for developers who know little or nothing about programming. The actions are translated quietly into GML, which is a C-based language, allowing novice developers to develop into programming. Experienced developers can use GML to program more complicated and dynamic game logic [1][3].

GameMaker is mostly used for developing 2D games and there are also many other features, including Room Editor, which can be used to design levels, Sprite Editor, which can be used to design sprite animation, and Timeline and Path systems, which can be used to time events and paths of objects. Simple 3D functionality is also available, but it is not suitable for complicated 3D projects. Most programmers discourage other programmers from making 3D games with GameMaker Studio because of the difficulties faced with depth, light, and camera [3].



Fig. 19: GameMaker's Interface [21].

YoYo Games published GameMaker Studio 2.3+, a series of releases, in 2022 and then, which enhanced its resource management, function-based GML scripting support, and nested prefabs (referred to as "layers and instances"). These are more closely structured projects and take away from the modularity of more complicated engines. Runtime support for more audio effects and filters, enabling developers to incorporate spatial sound and more in-game audio experiences, is another significant addition.

GameMaker has cross-platform export to Windows, macOS, Linux, HTML5, Android, iOS, PlayStation, Xbox, and Nintendo Switch, but export to consoles is more advanced license-based. The engine also supports direct publishing pipelines and has some support for external databases, so theoretically developers can utilize save systems, leaderboards, or other online persistent features [3][4].

All of which notwithstanding, GameMaker is not necessarily perfect. It has no support for more advanced graphical features, such as PBR, real-time lighting, or post-processing. It also has no support for modern 3D pipelines or standard 3D model formats. It has no support for extensive DCC tool integration. It has no support for plugins, although the Marketplace has reusable assets and scripts, and the

plugin system is quite restrictive compared to Unity or Unreal [6].

In Conclusion: GameMaker Studio can be considered an ideal gateway for new developers, as it remains one of the top game engines for rapid prototyping, 2D game creation, and education. It has very few entry points yet allows for the creation of commercial-quality games with no compromises. It balances game mechanics with coding by simplifying them to blocks and then nudging students towards coding. This is precisely in line with our purpose of informing new learners about the features of game engines through simple language.



Fig. 17: Katana Zero [33].



Fig. 18: Hotline Miami [34].

Some of the most legendary indie hits were created with GameMaker, including *Undertale*, *Hyper Light Drifter*, *Hotline Miami*, and *Katana Zero*. This demonstrates that with 2D-focused software, storytelling and creativity can be placed front and center. GameMaker's official access link can be found at [21].

G. Frostbite

Frostbite is an internal proprietary game engine developed by DICE for Electronic Arts (EA). It is used exclusively by EA and its associated studios. Frostbite was initially launched as part of *Battlefield: Bad Company* in 2008. It has now become a mature game engine for developing high-end and large-scale video games. Frostbite initially consisted of HDR Audio and Destruction 1.0, which was later replaced by Destruction 2.0, allowing for building destruction. Frostbite 1.5 was also launched, which featured building destruction. The engine has come a long way and has become mature enough to be used for developing high-end video games [11].

Frostbite 2 was launched in 2011. DirectX 11, 64-bit support, and real-time global illumination using Enlighten technology were incorporated. The graphics-intensive game *Battlefield 3* was also developed using Frostbite 2. Frostbite

3 was launched, which featured tile-based deferred shading, real-time subsurface scattering, bokeh depth of field, and better SSAO. The engine provided cinematic quality graphics while optimizing performance [11]. The game engine has also been used to launch other versions of *Battlefield*, such as *Battlefield V* and *Star Wars: Battlefront II* [22].

The game engine has faced challenges while being used to launch genres other than FPS. The challenges arose because of its initial launch as an FPS game engine. The limited resources of dialogue, UI, and branching narratives were also a challenge. The launch of RPGs such as *FIFA 17* was a challenge because of its initial launch as an FPS game engine [11].

Frostbite is a highly code-based engine, which means that there is no visual scripting or public support through documentation or forums. It also requires internal learning and very intimate knowledge of C++. This means that indie developers cannot use the Frostbite engine. It, however, excels over the other engines with regards to hardware-level optimization, scalability, and audio-visual quality, especially for AAA titles.

In conclusion, the Frostbite game engine has shown us that, despite the fact that a game engine can be highly proprietary, it can also be fine-tuned to provide visually stunning and highly optimized experiences, especially for different genres of games. It has also shown us that internal adaptability, especially for EA, is one of the benefits of having a highly customized game engine. For those who want to learn more about the design of game engines, Frostbite has shown us that specialization and control can also be just as good as flexibility, especially with the right creative direction.



Fig. 20: Need For Speed Unbound [35].



Fig. 21: FIFA 24 [36].

Some of the prominent video games that have been developed using the Frostbite game engine include

Battlefield, *Need for Speed*, *FIFA 24*, and *Dragon Age: Inquisition*. This is because Frostbite is not a publicly available game engine. The official page link of Frostbite can be accessed at [22].

H. RAGE

The Rockstar San Diego has developed Rockstar's own in-house proprietary game engine, which is referred to as the Rockstar Advanced Game Engine (RAGE), to develop some of the most iconic open-world games. It was originally released back in 2006 with Rockstar Games Presents Table Tennis. RAGE has eventually replaced Rockstar's earlier use of third-party game engines, such as RenderWare, to allow more creative and technological autonomy. RAGE is based on Angel Game Engine, with the addition of other technologies such as Euphoria, which is used for procedural animation of characters, and Bullet Physics, which is used for dynamic simulations of the game environments [11][12].



Fig. 22: Grand Theft Auto V [37].



Fig. 23: Red Dead Redemption 2 [38].



Fig. 24: Grand Theft Auto VI (trailer footage) [23].

RAGE was used as the game engine with the launch of *Grand Theft Auto IV*, which featured dynamic weather, physics, and next-generation lighting. RAGE has over time been enhanced to allow for enormous open-world

environments, frictionless character migration, and artificial intelligence systems, as seen with the launch of *Grand Theft Auto V* and *Red Dead Redemption 2*. It has enabled the creation of the most realistic game environments. It has featured features such as volumetric lighting, day-night cycles, atmospheric simulations, and natural NPC behaviors. RAGE has also been enhanced with features such as DirectX 11, HDR, and Blending Animation technologies. RAGE has over time been enhanced with more features with each of the subsequent game releases by Rockstar. For instance, Rockstar Games' latest release, *Red Dead Redemption 2*, has featured very complex AI worlds, realistic NPC behaviors, and stunning natural environments, which are benchmarks for a compelling game experience. The subsequent releases, such as *GTA VI*, are bound to push RAGE to new limits with the use of technologies such as Real-Time Global Illumination, Procedural Content Generation, and World Creation.

Although RAGE is technically robust, it is not an open engine. There are no open documentation, modding, or plugin marketplaces like in Unity or Unreal. The development process requires in-house skills and training. RAGE is also designed to be able to scale Rockstar's specific design needs. That is the strength of RAGE; it can be highly optimized. The limitation of RAGE is that it does not allow third-party developers to get involved in the development of the game.

In conclusion, RAGE is the force behind a design-centric game engine that scales with the age of a game studio's story and technical vision. The capacity of RAGE to evolve to accommodate Rockstar's expansive game worlds and cinematic ambitions is an indication of how proprietary game engines can facilitate innovation within game development. While RAGE is not an open-source game engine, it is the epitome of how game engine design can contribute to crafting a classic game that rises above genre conventions.

RAGE powered games in different genres, from *GTA IV*, *GTA V*, *Max Payne 3*, *Midnight Club: Los Angeles*, *Red Dead Redemption*, *Red Dead Redemption 2*, to the FIFA series of *FIFA 17* and beyond. These games showcase RAGE's capabilities in racing games, sports games, and action games despite being a closed engine. RAGE, just like Frostbite, is also not open source. The official page link can be accessed at [23].

IV. COMPARATIVE ANALYSIS

With so many game engines to select from today, it is quite challenging for game developers, especially new ones, to choose the best game engine for their purposes given that there are numerous game engines to choose from. Each of the engines has its own strengths and weaknesses. For this purpose, we have conducted an analytical comparison of eight of the most popular game engines, namely Unity, Unreal Engine, CryEngine, Godot, Game Maker, Amazon Lumberyard, Frostbite, and RAGE. Rather than going into technical details, this analytical comparison is conducted on practical factors such as ease of usage, platform supporting, scripting language, graphical capabilities, documentation, and community support. These factors are of particular importance to new game developers, or anyone interested in learning how different game engines correlate to different purposes. The following Table 1 is an easy comparison of the selected game engines on factors such as License Type,

Open Source, Best Use Case, Primary Language, Visual Scripting, 2D Game Support, 3D Game Support, Performance Optimization, Graphics Quality, AI & NPC

Support, Multiplayer Support, P.E., Cross-platform Support, Learning Curve, Asset Marketplace, Community & Docs, Latest Advancements & Notable Games.

TABLE I. COMARATIVE ANALYSIS TABLE

Attributes	Game Engines							
	<i>Unity</i>	<i>Unreal</i>	<i>CryEngine</i>	<i>Godot</i>	<i>GameMaker</i>	<i>Lumberyard</i>	<i>Frostbite</i>	<i>RAGE</i>
License Type	Free/ Premium	Free w/ royalties	Free w/ royalties	MIT License (Open Source)	Paid (Tiered)	Free (Proprietary)	Proprietary (EA-only)	Proprietary (Rockstar-only)
Open Source	No	No	No	Yes	No	No	No	No
Best Use Case	Indie, Mobile, Cross-platform	AAA, Cinematic, Multiplayer	High-end FPS, Realism	2D Indie, Education	2D Arcade, Learning	Cloud- Connected Games	FPS, Sports, Cinematic Games	Open-World Sandbox
Primary Language	C#	C++, Blueprints	C++, Lua, C#	GDScript, C#, C++	GML, Drag & Drop	C++, Visual Scripting	C++ (Internal)	C++ (Internal)
Visual Scripting	Bolt (Add-on)	Blueprints	Flow Graph (Limited)	Visual Script	Drag & Drop	Script Canvas	NA	NA
2D Game Support	Excellent	Good	Basic	Excellent	Excellent	Minimal	Minimal	Minimal
3D Game Support	Good	Excellent	Excellent	Moderate	Limited	Advanced	Excellent	Excellent
Performance Optimization	Moderate	High	High	High (Efficient)	Moderate	High (Cloud- assisted)	Very High	Very High
Graphics Quality	Moderate to High	Very High (Photorealistic)	Very High (Photorealistic)	Moderate	Basic to Moderate	High	Very High	Very High
AI and NPC Support	Basic (via NavMesh, Scripts)	Advanced (EQS, BTs)	Advanced (Territories, BTs)	Moderate (Customizable)	Basic	Moderate	Advanced	Advanced
Multiplayer Support	MLAPI (Lightweight)	Strong (Built- in)	Server- Authoritative	Experimental	Limited	AWS-Native	Internal	Internal
P.E.	NVIDIA PhysX	PhysX + Chaos Physics	CryPhysics + Destruction	Bullet + GodotPhysics	Basic 2D	CryPhysics base	Frostbite Physics	Bullet + Custom
Cross- platform Support	25+ Platforms	Major Consoles & PC	PC, Consoles, VR	Desktop, Mobile, Web	Many Platforms	PC & Consoles only	EA Platforms	Rockstar Platforms
Learning Curve	Beginner Friendly	Steep (for C++), Easier via Blueprints	Steep	Beginner Friendly	Very Beginner Friendly	Moderate	Hard	Hard
Asset Marketplace	Extensive	Good	Limited	Growing	Basic	Minimal	None	None
Community & Docs	Large, Active	Large, Active	Small, Limited	Growing Rapidly	Moderate	Small	Internal Only	Internal Only
Latest Advancements	Unity DOTS, Shader Graph, URP	Nanite, Lumen, Chaos Physics	SVOGL, Advanced Terrain	Vulkan Renderer, XR Tools	GML Enhancements, New IDE	Amazon Polly, Twitch APIs	Real-time Destruction, Lighting	Volumetric Weather, AI Scripting
Notable Games	Among Us, Monument Valley	Fortnite, Gears of War	Crysis, Ryse	Cruelty Squad, Lumencraft	Undertale, Katana Zero	New World	Battlefield, FIFA, NFS	GTA V, RDR2, Max Payne 3

V. SUMMARY

The aim of the work was to provide an easy-to-understand review of eight of the most used game engines, including Unity, Unreal, CryEngine, Godot, GameMaker Studio, Amazon Lumberyard, Frostbite, and RAGE, which explored their strengths with their weaknesses with their practical uses. This study has also streamlined complicated technological information with the latest developments to provide precise game engine knowledge for new developers and students entering their careers.

This work has also detailed the technical features of each of the game engines through the explanation of their architecture, scripting, platform compatibility, and visual development features, along with their practical applications ranging from independent game development to elementary education, up to AAA video games and special effects. This has also compared the features of the game engines through their ease-of-use features, platform compatibility, user support networks, and performance enhancements. It has also distinguished itself from other studies through the connection between expert-level technical information and easy-to-understand information for new learners. More companies are using game engines for simulation, training, and other uses, which means that game engine knowledge has more professional value.

The choice of game engines must be based on the performance capabilities of the engineers, as well as the goals of the designers, their professional capabilities, and the dimensions of the projects. This review has also created a simple process for the reader to make smart decisions about the game engines they want to use to confidently get into this industry, as it has highlighted that each of these game engines excels in their respective fields and has their specific audience.

- Unity's flexibility and community support make it a great choice for independent and mobile game creators.
- Unreal Engine, with its advanced rendering and multiplayer capabilities, would be more suitable for projects that require more graphical intensity and cinematic experience.
- CryEngine, with its impressive graphical capabilities, would be more suitable for FPS games and similar settings, but it does come with a higher learning curve.
- Godot's ease of use and open-source nature would be a great choice for creating games that can adapt to different genres and styles, especially for 2D games and educational settings.
- GameMaker would be a great starting point for people who have never encountered game logic and programming before.
- Lumberyard would be a great choice for developers who would like to tap into AWS and Twitch capabilities.
- Frostbite and RAGE show us that proprietary engines can offer the best experiences in their respective

genres, even though they may not be available to the public.

It is recommended that new people in the field consider starting with engines that have lower barriers to entry, such as Godot and GameMaker, while others who wish to produce high-end games consider engines like Unreal and CryEngine once they have gained sufficient experience and skills in the field. As the gaming industry and the engines that power them continue to evolve, it would be interesting to see how these engines would be used to further explore other fields and opportunities, such as simulation, medical training, and virtual collaboration, and how they can be used to create new and interesting experiences in these fields.

ABBREVIATIONS AND ACRONYMS

The following *Table 2* provides all the major abbreviations used in this work.

TABLE II. ABBREVIATIONS

Abbreviations	Full Forms
"AI"	"Artificial Intelligence"
"AS"	"Animation System"
"AuS"	"Audio System"
"AWS"	"Amazon Web Service"
"CE"	"Collision Engine"
"CS"	"Core System"
"DCC"	"Digital Content Creation"
"DICE"	"Digital Illusions Creative Entertainment"
"DnD"	"Drag-and-Drop"
"DOTS"	"Data-Oriented Technology Stack"
"EQS"	"Environment Query Systems"
"FPS"	"First-Person Shooter"
"GFS"	"Gameplay Foundation Systems"
"GML"	"GameMaker Language"
"GPU"	"Graphics Processing Unit"
"GSS"	"Game-Specific Subsystems"
"HDR"	"High Dynamic Range"
"HDRP"	"High-Definition Render Pipeline"
"HUD"	"Heads-Up Display"
"IDE"	"Integrated Development Environment"
"MLAPI"	"Multiplayer Networking API"
"MMOG"	"Massively Multiplayer Online Games"
"MMORPG"	"Massively Multiplayer Online Role-Playing Game"
"NPC"	"Non-Playable Characters"
"O3DE"	"Open 3D Engine"
"OS"	"Operating System"
"PBR"	"Physics-based Rendering"
"PC"	"Personal Computer"
"PE"	"Physics Engine"
"PIL"	"Platform Independence Layer"

Abbreviations	Full Forms
"RE"	"Rendering Engine"
"RM"	"Resource Manager"
"RPC"	"Remote Procedure Calls"
"SDK"	"Software Development Kit"
"SSAO"	"Screen Space Ambient Occlusion"
"SVOGI"	"Sparse Voxel Octree Global Illumination"
"TH"	"Target Hardware"
"UI"	"User Interface"
"URP"	"Universal Render Pipeline"
"VE"	"Visual Effects"
"VR"	"Virtual Reality"
"XR"	"Extended Reality"

ACKNOWLEDGMENTS

"The authors would like to thank their supervisor who helped them throughout the course of development of this research and their 'Department of Computer Science & Engineering' of Chandigarh University."

REFERENCES

- [1] Antonio Andrade, "Game Engines: A Survey", *EAI Endorsed Transactions on Serious Games*, vol. 2, issue 6, e8, pp. 1-6, October 2015, doi: 10.4108/eai.5-11-2015.150615.
- [2] Barczak Andrzej Marian and Woźniak Hubert, "Comparative Study On Game Engines", (Studia Informatica - Systems and Information Technology) *Systemy i Technologie Informacyjne*, no. 1-2, pp. 5-24, 2015, doi: 10.34739/si.2019.23.01.
- [3] Sanja Pavkov, Ivona Franković and Nataša Hoić-Božić, *Comparison of Game Engines for Serious Games*. Department of Informatics, University of Rijeka, Rijeka, Croatia., pp. 728-733, May 2017.
- [4] C. Vohera, H. Chheda, D. Chouhan, A. Desai and V. Jain, "Game Engine Architecture and Comparative Study of Different Game Engines", *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 21-27, May 2021, doi: 10.1109/iccncnt51525.2021.9579618.
- [5] Aishwarya Shinde and Alister Quinny, *From Pixels to Progress: A Chronicle of Video Game History and Evolution*. St. Rock's College of Commerce and Science, Borivali (W), Mumbai, India (*IJARSC*), vol. 3, issue 6, pp. 214-217, Jan. 2023.
- [6] Eleftheria Christopoulou and Stelios Xinogalos, "Overview and Comparative Analysis of Game Engines for Desktop and Mobile Devices", *Department of Applied Informatics, University of Macedonia, Greece*, vol. 4, issue 4, pp. 21-36, Dec. 2017, doi: 10.17083/ijsg.v4i4.194.
- [7] Andreas Jungherr and Damien B. Schlarb, "The Extended Reach of Game Engine Companies: How Companies Like Epic Games and Unity Technologies Provide Platforms for Extended Reality Applications and the Metaverse", *Social Media + Society*, pp. 1-12, April-June 2022, doi: 10.1177/20563051221107641.
- [8] Mahesh Ranaweera and Qusay H. Mahmoud, "Deep Reinforcement Learning with Godot Game Engine", *MDPI Electronics*, 13, no. 5: 985, pp. 1-15, Jan-March 2024, doi: 10.3390/electronics13050985.
- [9] Holfeld, Julian. "On the Relevance of the Godot Engine in the Indie Game Development Industry." *ArXiv, Computer Science*. December 2023 - January 2024. <https://arxiv.org/abs/2401.01909>, doi: 10.48550/arXiv.2401.01909.
- [10] Freedman, Eric. "New Media Ecosystems: Amazon and the Advancing Game Economy." *Abstract Proceedings of DiGRA 2019 Conference: Game, Play and the Emerging Ludo-Mix*. 2019.
- [11] Ahmad, Faizi Noor. "An Overview Study of Game Engines." *Int. Journal of Engineering Research and Applications* 3, no. 5, pp. 1673-1693, October 2013. <https://www.ijera.com/>.
- [12] Sahay, Ayush, Tanupriya Choudhury, and Vivek Kumar. "Hierarchical Evolution of Digital Games.", *International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 381-386. IEEE, 2019.
- [13] Kirriemuir, John. "Video gaming, education and digital learning technologies." *D-lib Magazine* 8, no. 2 (2002): 7, doi: 10.1045/february2002-kirriemuir. Source: DBLP.
- [14] Unity Technologies, *Unity – Real-Time Development Platform*, Unity. [Online]. Available: <https://unity.com/>. [Accessed: Feb. 28, 2025].
- [15] Unity Technologies, *Unity Cloud Services*, Unity Cloud. [Online]. Available: <https://cloud.unity.com/>. [Accessed: Mar. 3, 2025].
- [16] Epic Games, *Unreal Engine – The most powerful real-time 3D creation tool*, Unreal Engine. [Online]. Available: <https://www.unrealengine.com/>. [Accessed: Mar. 7, 2025].
- [17] Crytek, *CryEngine – The complete solution for next generation game development*, CryEngine. [Online]. Available: <https://www.cryengine.com/>. [Accessed: Mar. 11, 2025].
- [18] Godot Engine, *Godot Engine – Free and open source 2D and 3D game engine*, Godot. [Online]. Available: <https://godotengine.org/>. [Accessed: Mar. 17, 2025].
- [19] Amazon Web Services, *Amazon Lumberyard Game Engine*, AWS. [Online]. Available: <https://aws.amazon.com/lumberyard/>. [Accessed: Mar. 22, 2025].
- [20] Open 3D Foundation, *Open 3D Engine (O3DE)*, O3DE. [Online]. Available: <https://www.o3de.org/>. [Accessed: Mar. 25, 2025].
- [21] YoYo Games, *GameMaker – 2D Game Development Engine*, YoYo Games. [Online]. Available: <https://www.yoyogames.com/studio>. [Accessed: Mar. 30, 2025].
- [22] Electronic Arts, *Frostbite Engine*, EA. [Online]. Available: <https://www.ea.com/frostbite>. [Accessed: Apr. 4, 2025].
- [23] Rockstar Games, *Official Rockstar Games Website*, Rockstar. [Online]. Available: <https://www.rockstargames.com/>. [Accessed: Apr. 8, 2025].
- [24] Facepunch Studios, *Rust* [Online]. Available: <https://rust.facepunch.com/>. [Accessed: Apr. 07, 2025].
- [25] Mediatonic, *Fall Guys* [Online]. Available: <https://www.fallguys.com/>. [Accessed: Apr. 08, 2025].
- [26] PUBG Corporation, *PUBG: Battlegrounds* [Online]. Available: <https://pubg.com/>. [Accessed: Apr. 09, 2025].
- [27] Game Science, *Black Myth: Wukong* [Online]. Available: <https://www.heishenhua.com/en/>. [Accessed: Apr. 10, 2025].
- [28] Ubisoft, *Far Cry 6* [Online]. Available: <https://www.ubisoft.com/en-us/game/far-cry/far-cry-6>. [Accessed: Apr. 11, 2025].
- [29] Crytek, *Crysis 3 Remastered* [Online]. Available: <https://www.crytek.com/games/crysis3remastered>. [Accessed: Apr. 12, 2025].
- [30] Bytten Studio, *Cassette Beasts* [Online]. Available: <https://www.cassettebeasts.com/>. [Accessed: Apr. 13, 2025].
- [31] Pocket Trap, *Dodgeball Academia* [Online]. Available: <https://www.dodgeballacademia.com/>. [Accessed: Apr. 14, 2025].
- [32] Amazon Game Studios, *The Grand Tour Game* [Online]. Available: <https://www.amazon.com/adlp/thegrandtourgame>. [Accessed: Apr. 15, 2025].
- [33] Askiisoft, *Katana ZERO* [Online]. Available: <https://katana-zero.com/>. [Accessed: Apr. 16, 2025].
- [34] Dennaton Games, *Hotline Miami* [Online]. Available: <https://www.devolverdigital.com/games/view/hotline-miami>. [Accessed: Apr. 17, 2025].
- [35] Electronic Arts, *Need for Speed Unbound* [Online]. Available: <https://www.ea.com/games/need-for-speed/need-for-speed-unbound>. [Accessed: Apr. 18, 2025].
- [36] EA Sports, *EA Sports FC 24* [Online]. Available: <https://www.ea.com/games/ea-sports-fc/fc-24>. [Accessed: Apr. 19, 2025].
- [37] Rockstar Games, *Grand Theft Auto V* [Online]. Available: <https://www.rockstargames.com/V>. [Accessed: Apr. 19, 2025].
- [38] Rockstar Games, *Red Dead Redemption 2* [Online]. Available: <https://www.rockstargames.com/reddeadredemption2>. [Accessed: Apr. 20, 2025].
- [39] itch.io, "Top Game Engines," itch.io. [Online]. Available: <https://itch.io/game-development/engines>. Accessed: Apr. 20, 2025.