



## ПРЕГЛЕД НА СОФТУЕР ЗА РАЗРАБОТВАНЕ НА 3D ИГРИ

Петя Петрова, Милена Стефанова

### 3D GAME DEVELOPMENT SOFTWARE REVIEW

Petya Petrova, Milena Stefanova

**Abstract:** The paper focuses on the significance of software in the development of 3D games. An overview of popular software environments in the selected field is made. It is emphasized that each has its strengths and weaknesses when it comes to developing 3D projects. Based on selected criteria, a comparison of the reviewed software is presented. The process of developing 3D games is described with a set of rules to follow. The guiding principle of the research is the opinion that the use of specialized software in the development of 3D games leads to an improvement of the user experience when interacting with them.

**Keywords:** Unity, Unreal Engine, Godot, Flutter, 3D game, Game development.

## ВЪВЕДЕНИЕ

Игрите и анимациите са неотменна част от битието на хората в наши дни. Децата израстват с анимации и игри, които имат важна роля не само за отдих, но и по време на обучението им. В статията са разгледани популярни среди за разработка на игри и анимации, като Unity, Unreal Engine, Godot и Flutter, като са описани техните положителни и отрицателни страни.

Изборът на софтуер, подходящ за разработването на игра е от решаващо решение, което значително би могло да повлияе върху крайния успех на играта. Темата е изследвана от Chia et al. (2020), като авторите застъпват тезата, че с разнообразието от опции, налични на пазара днес, е предизвикателство да бъде определена платформата, удовлетворяваща нуждите на екипа от разработчици [3]. Ключови фактори при избор на Engine за игра са описани в детайли от Ana Crudu & MoldStud Research Team. Като водещи такива са жанрът на играта и поддръжката на платформи, за които тя е предназначена [6].

Вниманието в настоящата статия е посветено на преглед и сравнение на четири водещи платформи за разработване на 3D игри: Unity, Unreal Engine, Godot и Flutter. Целта на проучването е да предостави обективна и подробна информация за всяка от тези платформи с акцент върху техни основни характеристики, предимства и недостатъци, както и на специфичните случаи, при които те са най-подходящи. В резултат от проведения анализ са обобщени и предложени препоръки за избор на подходяща платформа – в зависимост от нуждите и целите на разработчиците.

## Изложение

Изследването е посветено на детайлен преглед на платформи за разработване на 3D игри и потребителски интерфейси. Разгледани са техни отличителни черти, силни и слаби страни. Предложени са конкретни стъпки, важни в процеса на разработване на игра. В изследването на плат-

формите за разработка на игри е отделено специално внимание на Flutter, като софтуер, насочен към разработката на потребителски интерфейс.

## 1. Unity

Това е една от най-популярните и широко използвани платформи за разработване на игри в света. Създадена през 2005 г., тя предлага мощни инструменти и гъвкавост, които позволяват на разработчиците да създават висококачествени 2D и 3D игри [8]. Основните характеристики на Unity включват мощен редактор, поддръжка на множество платформи и богат магазин за активи (Asset Store), където разработчиците могат да намерят готови ресурси и плъгини [11].

### Предимства на Unity

Безплатна версия с основните функции, необходими за разработка на игри, защото компанията иска да осигури достъп до своя игрови двигател (Game Engine) за широк кръг разработчици, независимо от техния бюджет.

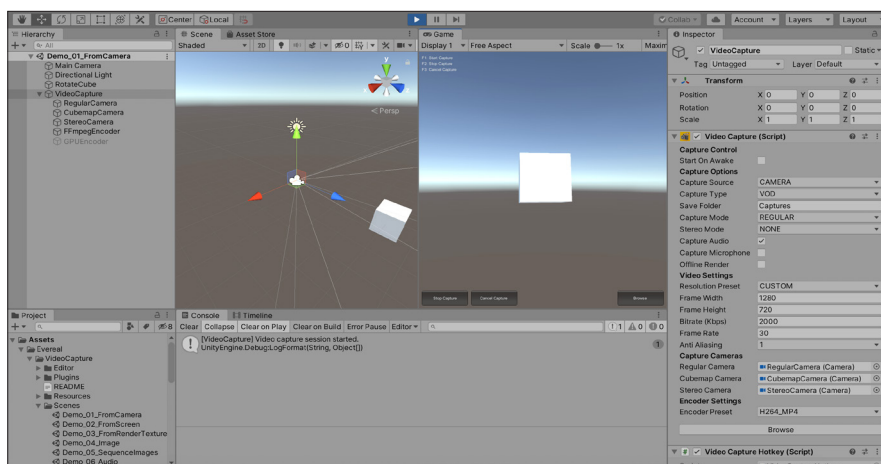
Лекота на използване. Програмата има интуитивен интерфейс и добре структурирана система, която позволява бърз старт на работа по проектите.

Лесен за научаване. Unity предлага интуитивен интерфейс и обширна документация, което го прави достъпен за начинаещи разработчици. Проектиран е с мисъл за лесна навигация и използване, което помага на новите потребители бързо да се запознаят с основните функции и инструменти.

Голямо и активно общество. Има значителна и активна общност, която предоставя поддръжка и споделя ресурси и уроци. Много разработчици по света използват Unity и споделят своите знания и опит.

Редица готови ресурси и плъгини. Магазинът на Unity предлага богат избор от предварително създадени активи, които могат да ускорят разработката на игри. Разработчиците могат да закупят или изтеглят безплатни ресурси, вместо да ги създават от нулата, което спестява време и усилия [1].

На Фигура 1. е представен екран с разнообразни панели с инструменти, достъпни при разработването на проект в Unity 3D.



Фигура 1. Работен екран на Unity

### Недостатъци на Unity

По-ниска производителност. Използва C#, което може да е по-бавно в сравнение с други скриптов езици от ниско ниво.

Платени функции в Pro версията. Някои от по-мощните функции на Unity, като по-добри инструменти за анализ и разширени възможности за рендеринг са налични в платената версия.

Ограничения при работа с много големи проекти. При сложни проекти може да има проблеми с производителността поради по-бавно време за зареждане на големи сцени и множество обекти, както и по-ниска обща производителност [1].

Не поддържа препратки към външни библиотеки. Всички използвани библиотеки трябва да бъдат копирани ръчно в проекта. Това може да затрудни управлението и актуализацията на външни зависимости и да усложни процеса на интеграция на външни библиотеки [2].

Проектите не се мащабират добре. Unity е отличен за малки до средни проекти, но може да се нуждае от допълнителна оптимизация и внимание при по-големи и сложни разработки.

## 2. Unreal Engine

Създадена от Epic Games, Unreal Engine е водеща платформа за разработка на игри, известна с впечатляващи графики и визуални ефекти. Тя предлага мощни инструменти и гъвкавост, които я правят предпочитан избор за разработка на големи и сложни проекти [5], включително AAA игри. "Triple A" е термин, често срещан в индустрията на игрите. С „AAA игри“ се обозначават високо бюджетни игри, както и компаниите с най-високи приходи в игралната индустрия. В рамките на игрите, "Triple A" навлиза в края на 90-те години, когато те се превръщат в големи "blockbuster" заглавия. При игрите, AAA се използва за обозначаване на заглавия и компании, които представяват най-сигурните възможности и е най-вероятно да постигнат високи приходни цели [13].

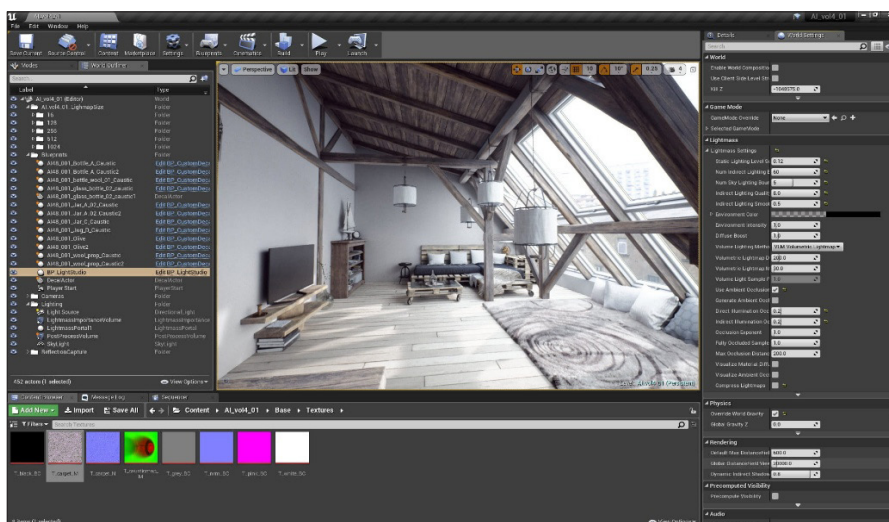
Unreal Engine предлага широка гама от инструменти за разработка на сложни и визуално впечатляващи игри, Фиг. 2. [5]. Подходящ софтуер е за работа по мащабни проекти.

### *Предимства на Unreal Engine*

Безплатен за училище и университети. Unreal Engine предоставя безплатен достъп за образователни институции като училища и университети, което насърчава обучението и развитието на студентите в областта на разработката на игри и визуализацията [5].

Бързо генериране на графика. С помощта на мощния си графичен двигател, Unreal Engine може бързо да генерира реалистични и красиви графики [2].

AI система в Unreal Engine. В авторските проекти на Unreal Engine 4 (UE4) е възможно създаването с изкуствен интелект (AI) за герои или други обекти. По този начин множество системи могат да работят заедно – от начин на поведение на героя, до отстраняване на грешки.



Фигура 2. Работен екран на Unreal Engine

### *Недостатъци на Unreal Engine*

По-сложен за научаване и използване. Изучаването на програмен език е трудно, Unreal е мащабна платформа, предоставя сложен набор от инструменти, които могат да се използват за всичко – от игри до филми до статични рендове. Изисква се време и усилия за научаването му.

По-малко ресурси и общност в сравнение с Unity. Въпреки че има значителна общност, тя не е толкова голяма и активна като тази на Unity, което може да ограничи достъпа до полезна информация и подкрепа за разработчиците [1].

### 3. Godot

Godot е сравнително нов софтуер за разработка на игри, създаден през 2014 г. Предлага изключителна гъвкавост и е подходящ както за 2D, така и за 3D игри. Godot е безплатен и с отворен код, което го прави достъпен за всички разработчици [4]. Екранна снимка на програмата е представена на Фиг. 3. [4].

#### Предимства на Godot

Безплатен и с отворен код. Godot е напълно безплатен и с отворен код, което осигурява голяма гъвкавост и възможности за персонализиране, без ограничения по лицензиране или такси за използване.

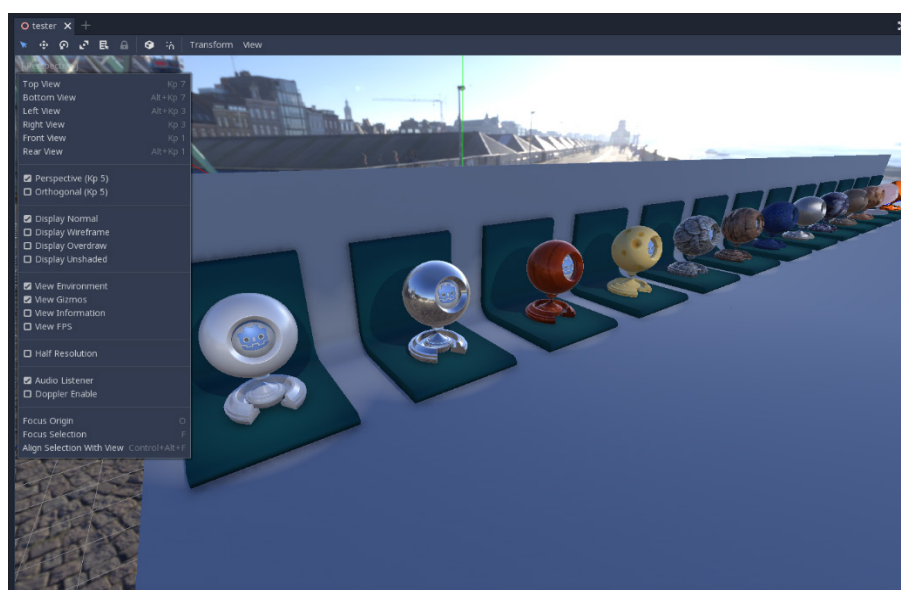
Лек и лесен за използване. Godot е лек и бърз, с интуитивен интерфейс и добре организирана структура, което го прави достъпен и удобен за работа дори за начинаещи разработчици.

Интуитивен потребителски интерфейс и мощен език за скриптове. По този начин е подходящ както за напреднали разработчици, така и за начинаещи разработчици, които искат да започнат да създават игри без предварителен опит в програмирането или дизайна на игри.

#### Недостатъци на Godot

По-малка общност и по-малко ресурси. Като по-нова платформа в сравнение с Unity и Unreal Engine, Godot има по-малка общност и ограничен достъп до помощ за разработчици.

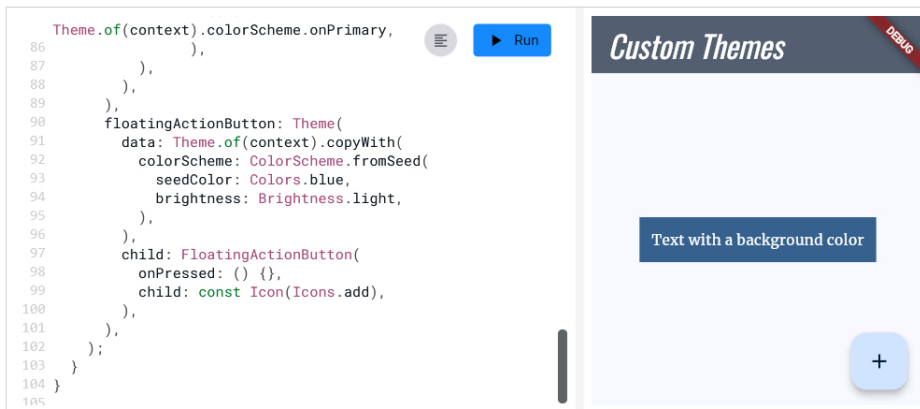
Ограничени графични възможности. В сравнение с Unity и Unreal Engine, Godot има по-ограничени графични възможности, което може да ограничи креативността и резултатите при създаване на визуално впечатляващи игри.



Фигура 3. Работен екран на Godot

### 4. Flutter

Разработена от Google, Flutter е програмна среда за разработка на потребителски интерфейси (UI). Въпреки че не е предназначена за разработка на игри, тя предоставя интересни възможности за междуplatformни приложения [7]. На Фиг. 4. [12] е представен фрагмент от кода за настройване на цветове и тема в приложение, разработено с Flutter.



Фигура 4. Настройка на цветове и теми във Flutter

### Предимства на Flutter

Лесна за научаване. Flutter е лесна за научаване и използване програмна среда, с добра документация и много ресурси за обучение [7].

Ефективна разработка. Flutter предоставя бързо и ефективно развитие на междуплатформенни мобилни приложения [2].

Интуитивен потребителски интерфейс. Flutter предоставя интуитивни инструменти и библиотеки за създаване на привлекателен и функционален интерфейс. По този начин подпомага създаването на приложения с високо ниво на потребителско удовлетворение [9].

### Недостатъци на Flutter

Ограничена поддръжка за 3D. Flutter не е оптимална програмна среда за разработка на 3D игри и приложения [10].

Мащабируемост. За Flutter има предизвикателства при мащабирането на приложения – за големи и сложни проекти, особено в областта на 3D графиката.

По-високи разходи за разработка. Използването на допълнителни библиотеки и плъгини, както и необходимостта от по-сложни решения за мащабируемост, може да доведе до увеличаване на разходите за разработка на приложения във Flutter [9].

## 5. Планиране при създаването на 3D игра

В процеса на проектиране и разработване на игра е добре да се спазва предварително работен план. Преминава се към изпълнение на замисъла в него, като преди това е важно да се познава и да се борави с лекота с интерфейса на софтуера, предвиден за създаването на играта.

Последователност от стъпки за създаване на игра

- Избор на платформа за разработка.
- Ясна идея и концепция за играта. Определяне на жанра и целевата аудитория, идеи за създаване на герои, дизайн.
- Дизайн на играта – създаване на 3D модели, текстури, анимации и звукови ефекти.
- Програмиране и скриптове. Това включва програмиране на основните механики, управление на героите, взаимодействие с обектите и други.
- Тестване и отстраняване на грешки.

## 6. Сравнение на платформите

В тази секция е представено обобщение и таблична съпоставка на изследвания софтуер.

Таблица 1. Сравнение на разгледаните игрови двигатели (Game Engine)

Engine	Unity 3D	Unreal	Godot	Flutter
Отворен код	-	+	+	+
Безплатен за използване	+	-	+	+
3D	+	-	+	+
2D	+	-	+	+
Подходящ за начинаещи	+	-	+	+

*Unity* може да се сблъска с предизвикателства при мащабирането на приложения, особено когато става въпрос за големи и сложни проекти. Софтуерът *Unity* не е с отворен код, но е безплатен за използване. Предоставя възможност за разработка на 3D приложения и игри. Счита се за подходящ за начинаещи, особено ако те имат начални познания в областта.

*Unreal Engine* също не е с отворен код, но предлага безплатен достъп за училища и университети. *Unreal Engine* е изключително мощен инструмент за разработка на 3D приложения и игри, като може да бъде по-труден за усвояване от начинаещите.

*Godot* е с отворен код и безплатен за използване. Предлага възможности за разработка на както 2D, така и 3D приложения, същевременно е подходящ и за начинаещи разработчици.

*Flutter* също е безплатен за използване, макар да не е с отворен код, Въпреки че не се фокусира специално върху 3D графика, той предлага възможности за разработка на красиви и високопроизводителни потребителски интерфейси за мобилни и уеб приложения. *Flutter* също е подходящ за начинаещи разработчици.

## ЗАКЛЮЧЕНИЕ

Може да се обобщи, че изборът на платформа за разработка на 3D игри зависи от множество фактори, включително от опита на създателя на игра, специфичните нужди на проекта и ресурсите, които са на лице. Въз основа на проведеня сравнителен анализ, следва синтезирано обобщение на причините за избор на всяка от разгледаните софтуерни среди.

**Unity** е отличен избор за тези, които търсят гъвкава и лесна за научаване платформа с голяма общност и богат набор от ресурси. Тя е подходяща за разработка на широк спектър от проекти, от малки мобилни игри – до по-сложни настолни приложения. *Unity* може да бъде точният избор за начинаещи или за напреднали, които се нуждаят от бърз старт.

**Unreal Engine** е софтуер, предпочитан за създаване на висококачествени игри с впечатляваща графика и сложни механики. Подходящ е за професионални разработчици и големи проекти, които изискват мощен игрови двигател (Engine) и възможности за създаване на AAA игри. *Unreal Engine* е подходящ за хора с опит в програмирането и за разработчиците на игри.

**Godot** е безплатен и с отворен код, което го прави лесен за използване. Той е подходящ за по-малки проекти и тези, които искат да експериментират с 2D и 3D разработка. Софтуерът е безплатен лек и иновативен, което го прави добър избор.

**Flutter** е специфична програмна платформа (framework), насочена основно към разработка на потребителски интерфейси и междуплатформени мобилни приложения. Въпреки че не е оптимална за разработването на сложни 3D игри, тя предлага висока производителност и лесно използване за UI разработки. Ако проектът на разработчика включва основно потребителски интерфейси и мобилни приложения, *Flutter* може да бъде подходящия избор.

## ЛИТЕРАТУРА

[1.] Admin. (2024, April 3). *Unity vs Unreal Engine: Pros and Cons*. Kevuru Games. <https://kevurugames.com/blog/unity-vs-unreal-engine-pros-and-cons/>

[2.] Berezovskyi, O. (2024, July 3). *Unity vs Unreal: Choosing the Best Engine for Your Game in 2024*. Pingle Studio. <https://pinglestudio.com/blog/unity-vs-unrealchoosing-the-best-engine-for-your-game-in-2024>

- [3.] Chia, A., Keogh, B., Leorke, D., & Nicoll, B. (2020). Platformisation in game development. *Internet Policy Review*, 9(4). doi: 10.14763/2020.4.1515
- [4.] Engine, G. (n.d.). *Godot Engine – Free and open source 2D and 3D game engine*. Godot Engine. <https://godotengine.org/>
- [5.] Epic Developer Community Forums. (n.d.). Epic Developer Community Forums. <https://forums.unrealengine.com/>
- [6.] Exploring the world of game development software. MoldStud. (2024, March 24). <https://moldstud.com/articles/p-exploring-the-world-of-game-development-software>
- [7.] Flutter – Build apps for any screen. (n.d.). <https://flutter.dev/>
- [8.] Lanzinger, F. (2022). 3D Game Development with Unity. doi: 10.1201/9780429328725.
- [9.] LeanCode. (2024, January 15). Flutter Pros and Cons 2024 - Summary and recommendations. <https://leancode.co/blog/flutter-pros-and-cons-summary>
- [10.] Petrovskaya, V. (2023, November 27). When not to use Flutter: Complete list of cases. WTF Blog. <https://blog.flutter.wtf/when-not-to-use-flutter/>
- [11.] Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine. (n.d.). Unity. <https://unity.com/>
- [12.] Use themes to share colors and font styles. (n.d.). <https://docs.flutter.dev/cookbook/design/themes>
- [13.] Video game Dictionary – Why do we call them “AAA” games? | ECI Games. (n.d.). <https://www.ecigames.net/media/video-game-dictionary-why-do-we-call-them-aaa-games>

---

#### ИНФОРМАЦИЯ ЗА АВТОРИТЕ

**Петя Петрова**, студент, специалност „Компютърни науки“, Факултет „Математика и информатика“, Великотърновски университет „Св. св. Кирил и Методий“, e-mail: [s2109013196@sd.uni-vt.bg](mailto:s2109013196@sd.uni-vt.bg)

**Милена Стефанова**, главен асистент, доктор, Факултет „Математика и информатика“, Великотърновски университет „Св. св. Кирил и Методий“, e-mail: [m.stefanova@ts.uni-vt.bg](mailto:m.stefanova@ts.uni-vt.bg)

#### ABOUT THE AUTHORS

**Petya Petrova**, student in Computer Science, Faculty of Mathematics and Informatics “St. Cyril and St. Methodius” University of Veliko Tarnovo, e-mail: [s2109013196@sd.uni-vt.bg](mailto:s2109013196@sd.uni-vt.bg)

**Milena Stefanova**, Assistant Professor, PhD, Faculty of Mathematics and Informatics, “St. Cyril and St. Methodius” University of Veliko Tarnovo, e-mail: [m.stefanova@ts.uni-vt.bg](mailto:m.stefanova@ts.uni-vt.bg)