

# Обзор существующего программного обеспечения для создания шейдеров с точки зрения визуализации медицинских данных для обучения врачей.

Королева Полина Андреевна  
СПбГЭТУ «ЛЭТИ»  
Санкт-Петербург, Россия  
polina.koroleva05@mail.ru

Герасимова Тамара Владимировна  
СПбГЭТУ «ЛЭТИ»  
Санкт-Петербург, Россия  
graphics.tv@gmail.ru

*Аннотация — Через процесс обучения проходит каждый человек. Для упрощения обучения на наглядных интерактивных примерах применяется компьютерная графика. В компьютерной графике для ускорения вычислений и отрисовки сложных изображений используются шейдеры.*

*Данная статья посвящена проблеме визуализации медицинских данных с помощью шейдеров. В статье проведен обзор программного обеспечения, позволяющего разрабатывать шейдеры и выбрано наиболее подходящее программное обеспечение для разработки шейдеров визуализирующих медицинские данные для обучения врачей. Отобраны аналоги, среди которых: Unity, Unreal Engine, CryEngine, Godot Engine, Ogre3D. Сравнение проведено по следующим критериям: удобство настройки шейдеров, системные требования, графический API. Последний критерий значим для дальнейшего применения разработанных шейдеров в приложениях для обучения врачей, т.к. студенты должны иметь возможность запускать приложение на разных операционных системах.*

*В результате сравнения выбран движок Unity, этот движок предоставляет удобный интерфейс для создания шейдеров, что ускорит дальнейшую разработку, имеет низкие системные требования и поддерживает современное и кроссплатформенное графическое API Vulkan, благодаря чему разработанные шейдеры можно будет применять в приложении для любой операционной системы.*

*Ключевые слова—медицинские шейдеры, движки для создания шейдеров, обучение врачей.*

## I. ВВЕДЕНИЕ

Шейдеры - стремительно набирающая популярность сфера разработки, которая находит применение во многих сферах жизни человека [1], среди которых образование и медицина - одни из важнейших. Внедрение компьютерных технологий в обучение новых специалистов повысит эффективность процесса их профессиональной подготовки [2]. С учетом решений правительства о переходе на свободное программное обеспечение в сфере здравоохранения, обусловленных экономией финансовых средств и безопасностью [3-4], необходимо найти соответствующее программное обеспечение, которое позволит разработать шейдеры для визуализации медицинских данных с целью обучения врачей.

Объектом исследования являются шейдеры для визуализации медицинских данных, предметом

исследования - программное обеспечение для создания шейдеров. Целью является обзор программного обеспечения, позволяющего разработать шейдеры, применимые в визуализации медицинских данных для обучения врачей.

Для достижения поставленной цели были решены следующие задачи:

1. Обзор существующего программного обеспечения для создания шейдеров.
2. Определение критериев сравнения программного обеспечения для создания шейдеров для визуализации медицинских данных.
3. Сравнение выбранного программного обеспечения по заданным критериям.
4. Выбор программного обеспечения, удовлетворяющего заданным критериям.

## II. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ

### A. Принципы отбора аналогов

В роли сравнительных аналогов были рассмотрены графические движки, поддерживающие 3D графику.

Основным критерием служила возможность работы с шейдерами. Подбор аналогов осуществлялся с использованием ресурса Cyberleninka. Для поиска применялся следующий запрос: "движки шейдеры медицина", "движки шейдеры", "шейдеры медицина".

### B. Аналоги

#### a) Unity

Unity [5, 6] - это популярный мультиплатформенный игровой движок для разработки 2D и 3D приложений и игр.

Он поддерживает работу с шейдерами, предоставляя удобные инструменты для этого - шейдер-граф с встроенными функциями для работы с шейдерами, позволяющими экономить время на написании кода. Также, движок позволяет писать код на языке GLSL, расширяя функциональность шейдер-графа.

#### b) Unreal Engine

Unreal Engine [6, 7, 8] - это популярный многофункциональный игровой движок, поддерживающий высокое качество графики и обладающий продвинутыми возможностями. Он

использует систему шейдеров, основанную на Material Editor, где шейдеры создаются посредством GUI, также Unreal Engine предоставляет возможность для написания шейдеров посредством кода на языке HLSL для более детальной настройки шейдера.

c) CryEngine

CryEngine [6, 9, 10] - это игровой 3D движок. Шейдеры в нем могут объединяться, накладываться или создавать анимированные текстуры. CryEngine 3 использует унифицированную шейдерную архитектуру, при которой блоки GPU могут выполнять шейдеры любого типа, при этом вычислительная нагрузка равномерно распределяется между блоками GPU, что позволяет достичь высокой производительности. Однако, CryEngine имеет высокий порог входа, из-за чего новым пользователям требуется время на освоение движка.

d) Godot Engine

Godot Engine [11, 12] - это открытый кроссплатформенный 2D и 3D игровой движок. Для написания шейдеров используются языки GDScript или Visual Shader Language. Также движок предлагает графовое представление для разработки шейдеров.

e) Ogre3D

Ogre3D [13, 14] - это свободный графический движок для рендеринга трехмерной графики с открытым исходным кодом. Ogre3D поддерживает языки шейдеров GLSL для OpenGL и HLSL для Direct3D. Шейдеры можно настраивать в системе материалов, позволяя разработчику задавать параметры текстур, освещения и эффектов, не углубляясь в код.

III. КРИТЕРИИ СРАВНЕНИЯ АНАЛОГОВ

Для сравнения были выбраны следующие критерии:

- Удобство настройки шейдеров
- Графический API
- Системные требования

Ниже представлено более подробное описание каждого критерия.

A. Удобство настройки шейдера

Данный критерий оценивает удобство настройки шейдеров и скорость разработки, в зависимости от инструментов, которые предлагает движок. В разработке шейдеров для визуализации медицинских данных могут потребоваться следующие инструменты: тесселяция, текстурирование на основе карты высот, нормали, маски и текстурная генерация.

- Если движок не имеет встроенных инструментов и разработчику необходимо целиком реализовывать шейдер, движок получает низкую оценку по этому критерию.
- Если движок имеет часть инструментов, нужных для разработки данных шейдеров, он получает среднюю оценку.
- Если движок предлагает все нужные инструменты для разработки данных шейдеров, он получает высокую оценку.

B. Графический API

Этот критерий описывает, какие графические API поддерживает движок. Выбор графического API в соответствии с операционной системой и видеокартой позволит эффективно использовать ресурсы GPU и достичь наибольшей производительности. Необходимо выбрать такие графические API, которые поддерживаются большинством платформ, чтобы разработанные шейдеры могли эффективно работать на различных устройствах.

C. Системные требования

Данный критерий оценивает минимальные системные требования для разработки.

IV. СРАВНЕНИЕ АНАЛОГОВ

Проведенное сравнение представлено в таблице 1.

ТАБЛИЦА 1 – СРАВНЕНИЕ АНАЛОГОВ

Аналог	Критерий сравнения		
	Удобство настройки шейдеров	Графический API	Системные требования
Unity	Высокое	OpenGL, DirectX, Vulkan	Низкие (4 GB RAM, Storage 4 GB)
Unreal Engine	Высокое	OpenGL, DirectX, Vulkan	Высокие (8 GB RAM, Storage 40 GB)
CryEngine	Высокое	OpenGL, DirectX, Vulkan	Средние (4 GB RAM, Storage 8 GB)
Godot Engine	Среднее	DirectX, Vulkan, Metal, WebGPU	Низкие (4 GB RAM, Storage 1 GB)
Ogre3D	Низкое	OpenGL, DirectX	Низкие (2 GB RAM, Storage 1 GB)

V. ВЫВОДЫ ПО ИТОГАМ СРАВНЕНИЯ

Среди графических API, поддерживаемых рассмотренными движками, представлены:

- OpenGL [15] - кроссплатформенное графическое API. Считается устаревшим, так как перестал обновляться, последняя версия вышла в 2022 году.
- DirectX [16] - графическое API, предназначенное для разработки под Microsoft Windows.
- Vulkan [17] - кроссплатформенное развивающееся современное графическое API, последняя версия вышла 3 декабря 2024 года.
- Metal [18] - графическое API для операционных систем IOS.
- WebGPU [19] – экспериментальное графическое API, позволяющее разработчикам использовать GPU для отрисовки изображений в браузере. На данный момент поддерживается только в Internet Explorer.

В результате сравнения графических движков можно сказать, что нет универсальных движков, каждый имеет

преимущества и недостатки. Ниже представлены краткие выводы по каждому движку:

- Unity имеет низкие системные требования и предоставляет нужные инструменты для реализации шейдеров в рамках объекта исследования. Поддерживает современное API Vulkan, поэтому подойдет для разработки шейдеров под любую платформу, и специализированное API DirectX для разработки под Microsoft Windows.
- Unreal Engine имеет большой набор встроенных функций, что позволяет быстро достичь желаемого результата при разработке шейдеров. Как и Unity поддерживает API Vulkan и DirectX, что позволяет разрабатывать кроссплатформенные шейдеры. Однако движок имеет высокие системные требования.
- CryEngine позволяет создавать реалистичную графику и имеет средние системные требования, однако имеет высокий порог вхождения. Поддерживает API Vulkan и DirectX, обеспечивает разработку кроссплатформенных шейдеров.
- Godot Engine имеет низкие системные требования, однако среднее удобство настройки шейдеров. Поддерживает большое количество разных графических API, специализированных для Microsoft Windows, IOS и браузера, а также кроссплатформенное API – OpenGL для остальных операционных систем.
- Ogre3D имеет низкие системные требования, но не предоставляет встроенных инструментов для реализации шейдеров, что замедляет процесс разработки. Поддерживает графический API DirectX, предназначенный для Microsoft Windows и кроссплатформенный, но устаревший графический API – OpenGL.

## VI. ВЫБОР МЕТОДА РЕШЕНИЯ

Для текущей задачи создания шейдеров визуализирующих медицинские данные, был выбран движок Unity. Он обеспечивает необходимые инструменты для разработки - тесселяцию, текстурирование на основе карты высот, нормали, маски и текстурную генерацию. Unity не требователен к ресурсам системы и поддерживает кроссплатформенное и развивающееся графическое API - Vulkan, что позволит использовать шейдеры на любом устройстве.

Решение должно представлять шейдер, разработанный на движке Unity.

## ЗАКЛЮЧЕНИЕ

В ходе работы был проведен обзор существующего программного обеспечения для создания шейдеров и выбрано наиболее подходящее для визуализации медицинских данных при обучении врачей.

Для сравнения были выбраны движки Unity, Unreal Engine, CryEngine, Godot Engine, Ogre3D. Сравнение проводилось по следующим критериям: удобство настройки шейдеров, графический API, системные требования.

На основе результатов сравнения было принято решение использовать движок Unity, так как он предоставляет нужные инструменты для разработки шейдеров в рамках визуализации медицинских данных, имеет низкие системные требования и поддерживает кроссплатформенное графическое API - Vulkan. Таким образом поставленная цель статьи была достигнута.

Направлением дальнейших исследований является обзор существующих библиотек шейдеров, использующих приведенные выше инструменты, требуемые для визуализации медицинских данных и разработка на их основе шейдеров на движке Unity для визуализации медицинских данных для обучения врачей.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Сыроежко А.А. Технология трехмерной визуализации // Современное состояние, проблемы и перспективы развития отраслевой науки - Москва, 2016 - с.466-474
- [2] Аванесова Татьяна Панайотовна Эффективность компьютерной технологии обучения // Вестник Северного (Арктического) федерального университета. Серия: Гуманитарные и социальные науки. 2013. №4. URL: <https://cyberleninka.ru/article/n/effektivnost-kompyuternoy-tehnologii-obucheniya>
- [3] Свободное программное обеспечение в государственном секторе. Сборник материалов. — М. INFO-FOSS.RU, 2007. — 112 с
- [4] Тихомирова А.А., Котиков П.Е. О перспективах перехода на свободное программное обеспечение в здравоохранении / А.А. Тихомирова, П.Е. Котиков // Детская медицина Северо-Запада. — 2018. Т. 7. № 1. С. 315-316.
- [5] Unity : офиц.сайт - URL: <https://unity3d.com> (дата обращения: 02.12.2024 )
- [6] Бровкина А. С., Сейидов Р. Х. ИГРОВЫЕ ДВИЖКИ // Форум молодых ученых. 2017. №1 (5). URL: <https://cyberleninka.ru/article/n/igrovye-dvizhki> (дата обращения: 02.12.2024).
- [7] Unreal Engine : офиц.сайт - URL: <https://www.unrealengine.com/en-US> (дата обращения: 02.12.2024 )
- [8] Пасько Д. Н. Современные игровые движки // Инновационная наука. 2016. №2-3 (14). URL: <https://cyberleninka.ru/article/n/sovremennyye-igrovye-dvizhki> (дата обращения: 02.12.2024).
- [9] CryEngine : офиц.сайт - URL: <https://www.cryengine.com/> (дата обращения: 02.12.2024 )
- [10] Усков Максим Александрович ОБЗОР ПРЕИМУЩЕСТВ И НЕДОСТАТКОВ ИГРОВЫХ ДВИЖКОВ. ОБОСНОВАНИЕ ВЫБОРА ИНСТРУМЕНТОВ И ТЕХНОЛОГИЙ РАЗРАБОТКИ КЛИЕНТСКОЙ ЧАСТИ ИГРОВЫХ ПРИЛОЖЕНИЙ // Глобус: технические науки. 2020. №5 (36). URL: <https://cyberleninka.ru/article/n/obzor-preimuschestv-i-nedostatkov-igrovyyh-dvizhkov-obosnovanie-vybora-instrumentov-i-tehnologiy-razrabotki-klientskoy-chasti> (дата обращения: 01.12.2024).
- [11] Godot Engine : офиц.сайт - URL: <https://godotengine.org/> (дата обращения: 02.12.2024 )
- [12] Ломайкин А. С. GODOT ENGINE В СОВРЕМЕННОЙ ИГРОВОЙ ИНДУСТРИИ // E-Scio. 2023. №6 (81). URL: <https://cyberleninka.ru/article/n/godot-engine-v-sovremennoy-igrovoy-industrii> (дата обращения: 02.12.2024).
- [13] Абалаков И. Н., Зотин А. Г. Обзор 3D -движков с возможностью визуализации ландшафтных изображений и природных эффектов // Решетневские чтения. 2012. №16. URL: <https://cyberleninka.ru/article/n/obzor-3d-dvizhkov-s-vozmozhnostyu-vizualizatsii-landshaftnyh-izobrazheniy-i-prirodnih-effektov> (дата обращения: 02.12.2024).
- [14] Ogre3D : офиц.сайт - URL: <https://www.ogre3d.org/> (дата обращения: 02.12.2024 )
- [15] OpenGL Registry: офиц.сайт - URL: [https://registry.khronos.org/OpenGL/index\\_gl.php](https://registry.khronos.org/OpenGL/index_gl.php) (дата обращения: 10.01.2025)
- [16] The Microsoft DirectX: офиц.сайт - URL: <https://www.microsoft.com/en-us/download/details.aspx?id=35> (дата обращения: 10.01.2025)

- [17] Vulkan : офиц.сайт - URL: <https://www.vulkan.org/#latest-vulkan-api-extensions-and-additions> (дата обращения: 10.01.2025)
- [18] Metal: офиц.сайт - URL: <https://developer.apple.com/metal/> (дата обращения: 11.01.2025)

- [19] WebGPU: URL: [https://developer.mozilla.org/en-US/docs/Web/API/WebGPU\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebGPU_API) (дата обращения: 11.01.2025)