[1] Вяткин, С. И. Прямой рендеринг трехмерных объектов на основе функций возмущения с использованием графических процессоров / С. И. Вяткин, Б. С. Долговесов // Программные системы и вычислительные методы. – 2023. – № 1. – С. 42-50. – DOI 10.7256/2454-0714.2023.1.38263. – EDN IWRNCU.

|  |  |
| --- | --- |
| АННОТАЦИЯ: | |
|  | Объектом исследования является метод прямого рендеринга сложных трехмерных объектов на основе функций возмущения с применением графических процессоров, c использованием множества потоковых мультипроцессоров. Прямой рендеринг означает, что визуализация функционально заданных моделей происходит без их предварительного преобразования в другие форматы, например, в сетки треугольников.Метод исследования базируется на аналитической геометрии в пространстве, дифференциальной геометрии, теории интерполяции и теории матриц, опирается на математическое моделирование и теорию вычислительных систем. Основными выводами проведенного исследования являются: возможность прямого рендеринга функционально заданных объектов, при рендеринге важно, чтобы вычислительные процессоры не простаивали. Первая проблема, которая была решена, заключалась в том, что у разных графических процессоров разное количество потоковых мультипроцессоров. Поэтому необходимо было выбрать во время исполнения оптимальную стадию, с которой начиналась работа... |

[2]Sanzharov, V. V. Vector Textures Implementation in Photorealistic Rendering System on GPU / V. V. Sanzharov, V. A. Frolov, V. A. Galaktionov // Proceedings of the International Conference on Computer Graphics and Vision “Graphicon”. – 2022. – No. 32. – P. 15-25. – EDN CAHPAG.

|  |  |
| --- | --- |
| АННОТАЦИЯ: | |
|  | Vector images provide a resolution independent representation that can be efficiently scaled to any resolution. However, there is no ready-to-use solution for supporting vector texturing in GPU based photorealistic rendering systems. In this work we present the experience of implementing support for vector textures in such rendering system using two different approaches - one based on signed distance fields and another based on rasterization. We describe implementation details and analyze the quality and performance of these methods. |

[3] Саловаров, А. Е. Использование паттерна ECS для решения проблем проектирования архитектуры игровых движков / А. Е. Саловаров, Р. С. Долгих // Кулагинские чтения: техника и технологии производственных процессов : сборник статей XIX Международной научно-практической конференции. В. 3 ч., Чита, 28–30 ноября 2019 года. Том Часть 1. – Чита: Забайкальский государственный университет, 2019. – С. 77-81. – EDN NBJOZU.  
  
АННОТАЦИЯ:

В данной работе рассматриваются проблемы, возникающие при проектировании архитектуры игровых движков с применением объектно-ориентированного подхода к разработке, а также способ их решения, основанный на использовании архитектурного паттерна ECS.

[4] Мельников, В. А. Процесс разработки движка для 2D игр и интерфейсов Sad Lion Engine / В. А. Мельников // Вестник Сыктывкарского университета. Серия 1: Математика. Механика. Информатика. – 2019. – № 4(33). – С. 21-37. – EDN UJTLXQ.

|  |  |
| --- | --- |
| АННОТАЦИЯ: | |
|  | Статья описывает процесс поиска решений проблем, возникающих при разработке игр, на основе опыта, полученного в Sad Lion Studio. Рассматриваются готовые движки и описан путь появления собственного ядра игрового движка Sad Lion Engine. |

[5] Свидетельство о государственной регистрации программы для ЭВМ № 2023680880 Российская Федерация. Программная оболочка (капсула) для встраивания Vulkan-визуализации в OpenGL-комплексы (ПО "Инкапсулятор Vulkan-визуализации") : № 2023669870 : заявл. 26.09.2023 : опубл. 06.10.2023 / М. В. Михайлюк, П. Ю. Тимохин ; заявитель Федеральное государственное учреждение «Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук». – EDN BHQDVE.

|  |  |
| --- | --- |
| АННОТАЦИЯ: | |
|  | Программа позволяет создавать из задачи визуализации, реализованной с помощью API Vulkan, отдельный, самодостаточный объект - капсулу, встраиваемую в систему визуализации, основанную на API OpenGL. Капсула изолирует низкоуровневую Vulkan-реализацию от OpenGL-комплекса и обеспечивает вывод синтезированных с помощью Vulkan изображений в буфер кадра OpenGL. Интеграция капсулы с OpenGL-комплексом осуществляется с помощью высокоуровневого малоинвазивного интерфейса. Целью создания ПО «Инкапсулятор Vulkan-визуализации» является реализация для широкого класса систем виртуального окружения, построенных на базе API OpenGL, возможности использования инновационных графических технологий, поддерживаемых API Vulkan (например, аппаратно-ускоренной трассировки лучей). Наличие такой возможности позволяет разрабатывать качественно новые видеотренажерные комплексы, системы научной визуализации, образовательные приложения и др. Тип ЭВМ: IBM РС-совмест. ПК. ОС: Windows 10/11. |

[6] Gil, I. Performance Improvement Methods for Hardware Accelerated Graphics Using Vulkan API / I. Gil // 2022 6th International Conference on Information Technologies in Engineering Education, Inforino 2022 - Proceedings : 6, Moscow, 12–15 апреля 2022 года. – Moscow, 2022. – DOI 10.1109/Inforino53888.2022.9782991. – EDN GGAWYS.

|  |  |
| --- | --- |
| АННОТАЦИЯ: | |
|  | In the present article, performance of two computer graphics application programming interfaces (APIs), OpenGL and Vulkan is compared using the application developed with C++ programming language. One of the two API's for scene rendering can be chosen before compilation of program. OpenGL Shading Language (GLSL) is used to create shader for OpenGL backend. Simple shader with implementation of Phong-style shading was created in GLSL then compiled to SPIR-V language for Vulkan support. Results indicate that usage of Vulkan may double the framerate compared to OpenGL. Performance boost was achieved using pre-recorded command buffers, minimization of descriptor set count, using one vertex buffer and one index buffer per scene and using mip- maps for rendering. Framerate was captured using RenderDoc software. |

[7] Арбузов, С. С. Тесселяция трехмерного ландшафта средствами OpenGL 4.0 / С. С. Арбузов, Н. А. Селянин // Актуальные вопросы преподавания математики, информатики и информационных технологий. – 2020. – № 5. – С. 144-151. – EDN GCKJWA.

|  |  |
| --- | --- |
| АННОТАЦИЯ: | |
|  | Данная статья посвящена изучению вопросов, связанных с тесселяцией трехмерного ландшафта средствами OpenGL. Рассматривается спецификация OpenGL, а в частности, реализации тесселяции в ней. Описаны основные принципы работы тесселяции в OpenGL. Рассмотрены подходы тесселяции трехмерного ландшафта и принципы применения динамических уровней тесселяции LoD. Делается заключение о необходимости в изучении тесселяции и спецификации OpenGL для студентов и профильных специалистов. |

[8] Del Gallego, N. P. Constructing a game engine: A proposed game engine architecture course for undergraduate students / N. P. Del Gallego // Entertainment Computing. – 2024. – Vol. 50. – P. 100657. – DOI 10.1016/j.entcom.2024.100657. – EDN WWHKAV.

|  |  |
| --- | --- |
| АННОТАЦИЯ: | |
|  | This paper presents a structured series of lessons for teaching undergraduate computer science or game development students how to construct a game engine using C++ and standard graphics APIs such as OpenGL or DirectX. Our proposed course content discusses rendering topics, game object management, scene management, and rigid body physics where learners are tasked to incorporate them into a fully functional prototype scene editor. By incorporating actual coding into the course, students can gain a better understanding of low-level engine features and also hone their programming skills. In addition to the learning content, we provide recommended assessment methods. We include discussions about our teaching experience in handling a game engine architecture course, titled GDENG03, which 36 students have passed. Our paper stands out from related works by providing a framework for interested educators on how to write a 3D game engine and how to teach the same in class... |