

УДК 004.457

## ИССЛЕДОВАНИЕ ПАТТЕРНОВ ПРОЕКТИРОВАНИЯ ВИДЕОИГР ДЛЯ ФОРМУЛИРОВАНИЯ ОБОБЩЕННОЙ МОДЕЛИ РАЗРАБОТКИ

*С. И. Богданов, Е. П. Костюкова, А. Ф. Лысенко*

Донской государственной технической университет (г. Ростов-на-Дону, Российская Федерация)

Рассмотрено текущее применение параллельных вычислений в видеоиграх. Указана значимость параллельных вычислений в соответствии с законом Мура. Проведен сравнительный анализ. Выявлена проблема, связанная с параллельными вычислениями. Отмечены основные инструменты, направленные на внедрение многопоточной обработки в игровые движки. Описаны требуемые параметры для проведения эксперимента. Выдвинута гипотеза об оптимизации кодовой части Unity D.O.T.S.

**Ключевые слова:** видеоигры, архитектурные паттерны, параллельные вычисления, игровые движки, разработка игр.

## STUDY OF VIDEO GAME DESIGN PATTERNS FOR THE FORMULATION OF A GENERALIZED DEVELOPMENT MODEL

*S. I. Bogdanov, E. P. Kostyukova, A. F. Lysenko*

Don State Technical University (Rostov-on-Don, Russian Federation)

The current application of parallel computing in video games is considered. The significance of parallel computing is indicated, referring to Moore's law. A comparative analysis of game engines was carried out. A problem related to parallel computing has been identified. The main tools aimed at implementing multithreaded processing in game engines are noted. The required parameters for the experiment are described. A hypothesis is put forward about optimizing the code part of Unity D.O.T.S.

**Keywords:** video games, architectural patterns, parallel computing, game engines, game development.

**Введение.** Разработка компьютерных игр в данный момент является быстроразвивающейся индустрией [1]. В соответствии с темпами создания новых игровых проектов совершенствуются текущие и выходят в свет новые технологии [2], позволяющие сделать процесс создания игр более комфортным и эффективным. Однако создатели программного обеспечения (ПО) иногда недостаточно углубляются в процесс оптимизации своих конечных продуктов и их ресурсопотребление. Поэтому параллельные вычисления в данный момент очень актуальны, ведь они предоставляют огромный потенциал для этих целей [3].

Согласно закону Мура, вычислительная мощность процессоров растет экспоненциально, каждые два года увеличиваясь вдвое. Примерно с середины 2000-х годов по разным причинам производители процессоров предпочитают многоядерные архитектуры. Соответственно, для получения всей выгоды от возросшей производительности процессоров, программы должны переписываться в соответствующей манере, с использованием параллельных вычислений. В противном случае, разработчики столкнутся с проблемой, при которой для созданного ПО не будет хватать вычислительных мощностей.

**Основная часть.** На данный момент все ныне существующие игровые движки используют параллельные вычисления, однако не во всех аспектах — так, например, многопоточной обработке подвергаются рендеринг изображения, обработка физических вычислений, обработка звуковых дорожек. Несмотря на это, логика игры, ее правила и механики обрабатываются в одном

потоке одного ядра ЦП, что, при большом масштабе проекта выливается в так называемые «фризы» — моменты, когда в очереди на выполнение оказывается множество команд и время обработки значительно вырастает.

Решить данную проблему призван архитектурный паттерн Entity — Component — System (ECS) [4], который заменяет традиционную для разработки игр парадигму объектно-ориентированного программирования на параллельное программирование и работу с данными.

В Unity3D применение данного архитектурного шаблона представлено в виде D.O.T.S. [5] — пакетов дополнений, устанавливаемых поверх основного Unity. Этот пакет, помимо реализации систем сущностей, содержит в себе планировщик задач (C# Jobs), открывающий доступ к написанию безопасного многопоточного кода.

Целью данной работы является оптимизация архитектурного паттерна ECS в рамках Unity D.O.T.S., для чего необходимо:

1. Изучить внутреннее устройство компонентов D.O.T.S.
2. Разработать игровой проект, задействующий загрузку ЦП, версию с использованием D.O.T.S. и без; провести сравнительный анализ роста производительности.
3. Найти и устранить критические ошибки в кодовой части D.O.T.S.
4. Поставить эксперимент — внедрить исправленную версию D.O.T.S. в игровой проект. По результатам эксперимента провести сравнительный анализ с оригинальным решением и подтвердить или опровергнуть гипотезу исследования.

Для проведения показательного эксперимента потребуется разработать игровой проект, в котором будет задействовано множество однотипных 3D-объектов. Необходимыми требованиями к игровому движку в данном случае являются:

1. Поддержка 3D-графики.
  2. Полномасштабная поддержка параллельных вычислений, задействование всех доступных ресурсов процессора для обработки.
  3. Опционально, поддержка DataFlow-парадигмы программирования для лучшей работы с параллельными вычислениями.
  4. Возможность портировать готовые проекты на мобильные устройства.
- Проведем сравнительный анализ наиболее популярных игровых движков.

### **Unity3D**

Unity является одним из популярных движков, использующий язык программирования C#. Данный продукт предоставляет множество платных и бесплатных ассетов, которые помогут разработчикам в создании игры. Также существует пробная версия дополнительных программных пакетов D.O.T.S., позволяющая применять параллельные вычисления ко всем игровым составляющим — логика, физика, видео- и звуковые потоки.

### **Unreal Engine**

Unreal Engine [6] универсальный и очень гибкий движок, использующий язык программирования C++. Он наполнен множеством бесплатных инструментов разработки. Начинаям разработчикам в работе поможет визуальный редактор Blueprints, позволяющий создать игру без единой строки кода. Его преимуществом является поддержка UltraHD разрешения графики и прямая работа с памятью с помощью встроенных библиотек. Параллельные вычисления можно осуществлять с помощью встроенного в C++ планировщика задач, однако его невозможно применить к чему-либо, кроме логической части игры и физическим вычислениям.

### CryEngine

CryEngine [7] на данный момент представляет разработчикам возможность делать в своих проектах фотореалистичную графику. Данный движок предоставляет встроенную поддержку мультиточечных вычислений, однако, так же, как и в Unreal Engine, они применимы лишь на логическую и физическую части игрового проекта.

### Godot

Godot [8] развивающийся движок, использующий собственный язык программирования. Присутствует поддержка C++ и C#, в скором времени и Python. Идеально подойдет для знакомства с разработкой игр, если у пользователя уже есть небольшой опыт в программировании. Также имеет встроенный функционал распараллеливания процессов с помощью планировщика задач.

### GameMaker Studio 2

GameMaker Studio 2 [9] является хорошим вариантом для начинающих разработчиков, которые хотят попробовать себя в сфере разработки игр — движок имеет похожую на Blueprints в Unreal систему программирования D&D, имеет встроенный редактор графики. Однако данный продукт не предназначен для реализации 3D-проектов. Встроенный язык GML не имеет возможности вести параллельные вычисления.

В следующей сравнительной таблице приведены требования к игровым движкам и соответствие этим требованиям самих движков.

Таблица 1

Требования к игровым движкам и соответствие этим требованиям самих движков

Критерий/ Движок	Unity3D с D.O.T.S.	Unreal Engine	CryEngine	Godot	GameMaker Studio 2
Поддержка 3D	+	+	+	+	-
Полномасштабные параллельные вычисления	+	-	-	-	-
DataFlow -парадигма	+	+	+	+	+
Портирование на мобильные устройства	+	+	-	+	+

Исходя из таблицы сравнений, наиболее подходящим по критериям движком является Unity3D с пакетами D.O.T.S.

Целью проведения эксперимента является целесообразность применения D.O.T.S. как инструмента для запуска параллельной обработки видеоигр на ЦП. В данном эксперименте авторы планируют расположить на одной игровой сцене множество однотипных объектов (более 10000), задать им определенное поведение, запустить и измерить интересующие нас показатели на различных платформах, а именно:

1. Частоту кадров в секунду.
2. Нагрузку на процессор.
3. Наличие кратковременных просадок в работе процессора.
4. Нагрузку на видеокарту.

5. Температуру и энергопотребление процессора.

**Вывод.** Основываясь на теоретических знаниях о выгоде применения параллельных вычислений при разработке ПО, была выдвинута гипотеза: «Оптимизация кодовой части Unity D.O.T.S., отвечающей за взаимодействие Сущностей и Систем, позволит пользователям не углубляться в строение движка, а сфокусироваться на написании многопоточных, модульных игр». Также предполагается, что данные модификации полностью раскроют потенциал архитектурного паттерна ECS и Data Driven парадигмы программирования. В следующей публикации будут описаны результаты эксперимента и предложены наилучшие пути оптимизации игровых проектов.

#### Библиографический список

1. Индустрия моды — Индустрия компьютерных игр / Дата-центр Высшей школы экономики : [сайт]. — URL: <https://dcenter.hse.ru/data/2020/07/27/1599127653/-Индустриякомпьютерных%20игр-2020.pdf> (дата обращения : 06.04.2022).
2. Топ – 4 технологии разработки компьютерных игр в 2020 году / GDJob : [сайт]. — URL: <https://gdjob.pro/stati/raznoe/top-4-tekhnologii-razrabotki-kompyuternykh-igr-v-2020-godu> (дата обращения : 06.04.2022).
3. Достоинства и недостатки параллельного программирования / Электронный научно-практический журнал «Современные научные исследования и инновации» : [сайт]. — URL: <https://web.snauka.ru/issues/2016/06/69538/> (дата обращения : 06.04.2022).
4. Шаблон проектирования Entity-Component-System — реализация и пример игры / Хабр : [сайт]. — URL: <https://habr.com/ru/post/343778> (дата обращения : 06.04.2022).
5. Unity D.O.T.S. — Производительность по умолчанию / Unity3D : [сайт]. — URL: <https://unity.com/ru/dots> (дата обращения : 06.04.2022).
6. Unreal Engine 5 / UnrealEngine : [сайт]. — URL: <https://www.unrealengine.com/en-US/unreal-engine-5> (дата обращения : 06.04.2022).
7. CryEngine / CryEngine: [сайт]. — URL: <https://www.cryengine.com> (дата обращения : 06.04.2022).
8. Godot Engine - Free and open source 2D and 3D game engine / Godot Engine: [сайт]. — URL: <https://godotengine.org/> (дата обращения: 06.04.2022).
9. GameMaker Studio 2 / YoYoGames : [сайт]. — URL: <https://www.yoyogames.com/ru/gamemaker> (дата обращения : 06.04.2022).

*Об авторах:*

**Богданов Сергей Игоревич**, магистрант кафедры «Робототехника и мехатроника», Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [ghostofsergo@gmail.com](mailto:ghostofsergo@gmail.com)

**Костюкова Елена Петровна**, магистрант кафедры «Робототехника и мехатроника», Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [snowfaaall@mail.ru](mailto:snowfaaall@mail.ru)



**Лысенко Алексей Федорович**, заместитель директора Института опережающих технологий Донского государственного технического университета, к.т.н., доцент каф. «Робототехника и мехатроника» (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [aflysenko@gmail.com](mailto:aflysenko@gmail.com)

*About the Authors:*

**Bogdanov, Sergey I.**, Master's degree student, Department of Robotics and Mechatronics, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), [ghostofsergo@gmail.com](mailto:ghostofsergo@gmail.com)

**Kostyukova, Elena P.**, Master's degree student, Department of Robotics and Mechatronics, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), [snowfaaall@mail.ru](mailto:snowfaaall@mail.ru)

**Lysenko, Aleksey F.**, Associate Director, Institute of Emerging Technologies, Don State Technical University, Cand.Sci., Associate professor, Department of Robotics and Mechatronics (1, Gagarin sq., Rostov-on-Don, 344003, RF), [aflysenko@gmail.com](mailto:aflysenko@gmail.com)