

2. Абдуллаев А.М. Эконометрический анализ и моделирование тенденций развития инновационной экономики Т.: ТГЭУ, 2017-888с.
3. Гулямов С.С., Абдуллаев А.М. Инновационный потенциал и его влияние на конкурентное развитие экономики страны (теоретико-методологические аспекты) Т.:Фан ва технология, 2016-884с.
4. Гольдштейн Г.Я. Инновационный менеджмент: Учебное пособие. Таганрог: Издательство ТРТУ, 2001.— 315 с.
5. Корпоративное управление: системный подход. — М.: ЗАО Изд-во «Экономика», 2007.
6. Герасимов Б.И., Мозгов Н.Н. Маркетинговые исследования рынка: учебное пособие / Б.И. Герасимов, Н.Н. Мозгов. – М.: ФОРУМ, 2009. – 336 с.
7. Голубкова Е.Н. Деловые ситуации в маркетинге: 33 кейса / Е.Н. Голубкова, И.П. Широченская. – М.: Дело и Сервис, 2009. – 352 с

УДК 004.051

**Н.В. Филон, О.А. Новосельская**

Белорусский государственный технологический университет  
Минск, Беларусь

## **СРАВНИТЕЛЬНЫЙ АНАЛИЗ ДВИЖКОВ UNITY И UNREAL ENGINE**

*Аннотация.* В статье проведен анализ современных средств разработки компьютерных игр. На примере простого 2D экшен-шутера произведен сравнительный анализ движков Unity и Unreal Engine. Показаны примеры разработки фрагментов кода, настраивания сцены, создания интерфейса. В заключении отмечены преимущества и недостатки обоих движков.

**N.V. Filon, O.A. Novoselskaya**

Belarussian State Technological University  
Minsk, Belarus

## **COMPARATIVE ANALYSIS OF UNITY AND UNREAL ENGINES**

*Abstract.* The article analyzes modern means for developing computer games. On the example of a simple 2D action shooter, a comparative analysis of the Unity and Unreal Engine engines has been made. Examples of code fragments for assignment action to a character, setting up a scene, creating an interface are given. In conclusion, the advantages and disadvantages of both engines are noted.

Индустрия компьютерных игр – это постоянно растущий и развивающийся сектор экономики, главные цели которого – разработка, продажа и продвижение компьютерных игр. Каждая игра является творческим продуктом, как кинофильм или живопись, однако, из-за возможности полноценного взаимодействия с окружением внутри игры, пользователи получают от них наибольшее удовольствие. В зависимости от жанра, концепции и размера проекта, игра может разрабатываться от недели до нескольких лет. В прошлом, игры приходилось делать с абсолютного нуля. Простые 2D игры занимали тысячи строк кода, а добиться полноценного 3D могли лишь единицы. Все изменилось с появлением игровых движков. Код базовых игровых механик уже заложен в движок и пользователю лишь необходимо правильно их использовать. Сегодня почти все игры используют заранее написанный движок. В настоящее время наиболее распространенными являются следующие игровые движки [1]:

– Unity. Самый популярный и используемый сегодня относительно бесплатный редактор. Универсальность Unity – его основная проблема, так как создание уникальных проектов требует переписывания движка, что повлечет дополнительную оплату для доступа к подобным функциям. Основной язык разработки C# (си-шарп).

– Unreal Engine. Сложный для освоения редактор. Изначально созданный для игр, он менее универсальный, чем Unity, но при этом область его использования не ограничивается только видеоиграми. В последнее время обрел популярность в киноиндустрии и проектировании, за счёт лучшей графики и дешевизны использования. Основной язык разработки C++ (си-плюс-плюс), но также позволяет использовать альтернативное программирование – визуальное, при помощи блоков и чертежей. Абсолютно бесплатный, нужно лишь отдавать процент от своей прибыли разработчику.

– Godot. Новый и универсальный игровой движок. Позиционирует себя как «лучший выбор для первой игры». Простота в освоении, визуальное программирование и гибкость в настройке позволяют Godot быстро набирать популярность.

– CryEngine. Ранее, конкурент Unreal и Unity, но сегодня, из-за прекратившейся поддержки со стороны разработчиков и отсталости графики, почти не используется для серьёзных проектов.

Для сравнения были выбраны два самых популярных редактора: Unreal Engine и Unity. Основными критериями выбора жанра являются действия, которые игрок совершает в процессе взаимодействия с игровым пространством. Среди всех жанров (экшен, шутер,

приключения, ролевые игры RPG, симуляторы, стратегии [2]) было принято решение создать игру в жанре 2D экшн-шутера «Space Asteroids (Космические астероиды)», написанная на обоих движках.

Основная цель игры – не дать астероидам попасть по игроку. Для этого игрок, орудуя космическим кораблём, сбивает астероиды из оружия. За поражение астероидов игроку начисляются личные очки. Скриншот игры показан на рис. 1.

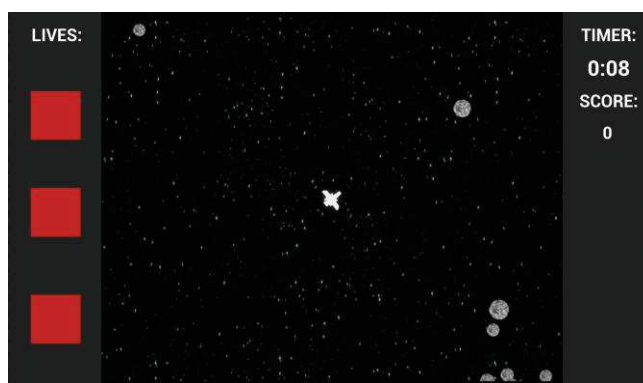


Рис. 1 – Скриншот из игры «Space Asteroids»

В ходе разработки проанализированы:

1. Процесс установки.
2. Создание проекта.
3. Игровая сцена.
4. Разработка персонажа игрока.
5. Разработка ботов (астероидов).
6. Создание интерфейса и системы подсчета очков.

Для сравнения необходимо отметить, что различия заметны уже на стадии установки. Несмотря на то, что оба движка используют специальную утилиту, итоговый размер движков на диске отличается в 4 раза (Unity занимает меньше дискового пространства, но при этом требует выбора вида лицензии).

При создании проекта в Unreal Engine есть готовые шаблоны 3D игр (Шутер, Гонки, VR, Головоломка), в то время как Unity позволяет выбрать только пространство – 2D или 3D.

По умолчанию в Unreal Engine на сцене ничего нет. Для возможности видеть сцену, в процессе игры, нужно поместить на неё камеру. Для ненастроенной камеры нужно установить разрешение изображения, игрока, угол обзора и тип проекции. В Unity, на сцене уже находится камера. Она сразу подстраивает разрешение, игрока и тип проекции под монитор. После этого нужно создать на сцене площадку, где будет происходить всё действие. В Unreal для этого необходимо из

текстуры звёзд сделать 2D спрайт и переместить его на сцену. В Unity достаточно переместить текстуру на сцену.

Разработка персонажа. В Unreal Engine создание персонажа оказалось достаточно сложным. Нужно создавать отдельный 3D компонент персонажа (актёра), добавлять ему возможность столкновения (коллизию), написать код поворота персонажа за мышкой и стрельбу. Для написания кода использовалось визуальное программирование Blueprint (рис. 2 а). В Unity создание персонажа оказалось более легким. Потребовалось поместить текстуру корабля на сцену и привязать к нему три компонента: Rigidbody (для физики и движения), Collider (для столкновений) и скрипт (программу для управления). Для написания кода использовался язык C#. Фрагмент кода для поворота представлен на рис. 2 б.

Разработка ботов (астероидов). В Unreal Engine потребовалось изменить код всей сцены, чтобы астероиды появлялись в нужных местах. В отличие от Unity, астероиды появлялись на определенных координатах и двигались к центру экрана.

Создание интерфейса и системы подсчета очков. Unreal Engine предоставляет удобную систему работы с интерфейсом. Достаточно создать класс «UI» и добавить туда необходимые окна. Основные элементы интерфейса: левая полоса (количество жизней) и правая полоса (время со старта игры и общий счёт очков). В Unity для создания интерфейса достаточно воспользоваться готовым шаблоном из меню GameObject–UI–.... После этого станет доступен инструмент настройки вида и положения кнопок и элементов. В обоих движках требуется создание программы, которая считывает количество жизней и очков для каждого элемента.

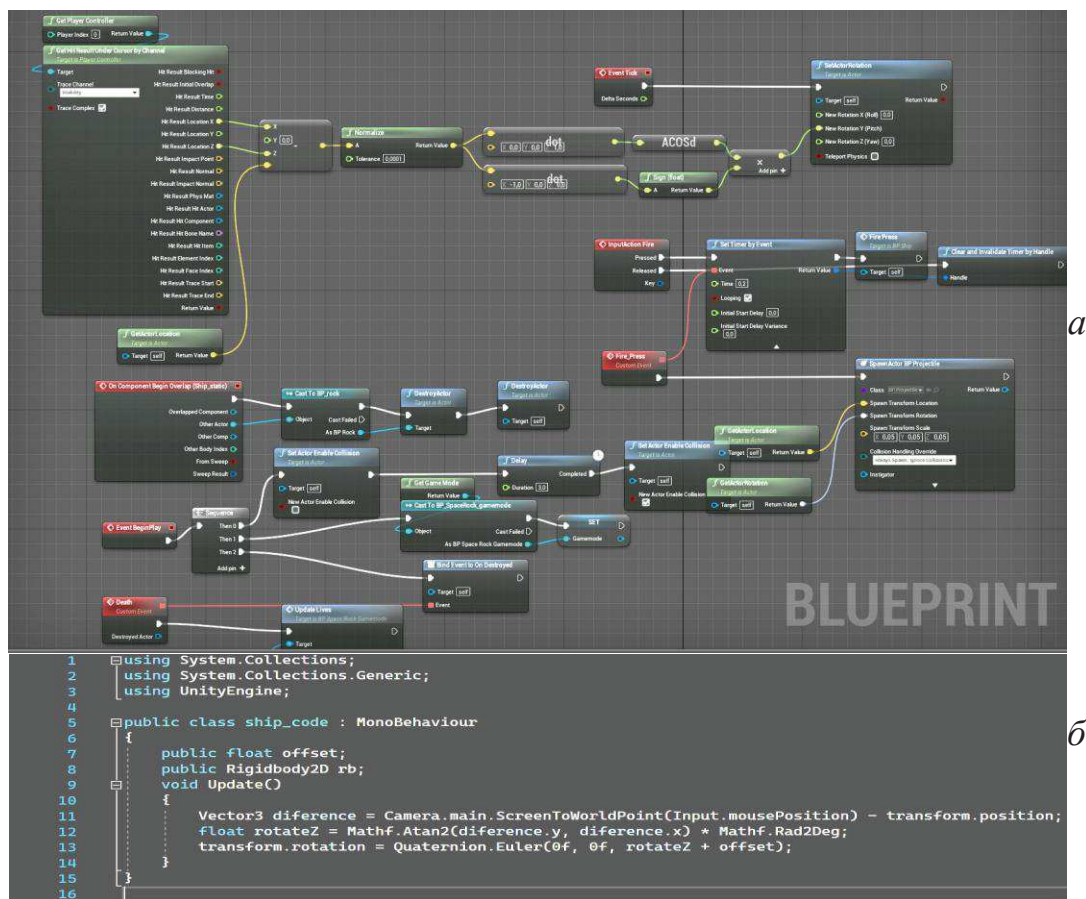


Рис. 2 – Сравнение программирования механик персонажа в Unreal Engine (а) и Unity (б)

В заключении следует отметить, что для каждой игры необходимо выбирать среду разработки в соответствии с жанром и гейм-дизайном. Unreal Engine определённо не подходит для создания простых 2D игр. Сложная реализация простых механик даже через визуальное программирование не позволяет действительно оптимизировать процесс создания игры. В Unity, движке, специально созданном для создания простых и сложных игр, разработка игры более легкая. Лёгкость в написании кода и понятный интерфейс позволили в более короткие сроки создать такую же игру, как и на Unreal Engine.

Однако при создании сложных визуально проработанных со сложными механиками игр на первое место выходит Unreal. Наилучшее качество графики, стабильность и доступность среды позволяют создавать различные 3D-игры, не волнуясь об оптимизации и стабильности работы движка.

### Список использованных источников

1. Сулопаров Д. На чем создают игры? Обзор популярных игровых движков [Электронный ресурс]. / Тргогер – издание о разработке и обо

всём, что с ней связано. – 2022. – Режим доступа: <https://tproger.ru/articles/na-chjom-sozdajut-igry-obzor-populjarnyh-igrovyh-dvizhkov/?ysclid=199jeemoh5388547437>. – Дата доступа: 08.10.2022.  
2. Компьютерные игры как искусство [Электронный ресурс]. / GamesIsArt. – 2022. – Режим доступа: [https://gamesisart.ru/game\\_dev\\_create.html](https://gamesisart.ru/game_dev_create.html). – Дата доступа: 08.10.2022.

УДК 34.04

**В.В. Харин**

Тамбовский государственный университет имени Г.Р. Державина  
Тамбов, Россия

## **«ПРАВОВЫЕ ПЕСОЧНИЦЫ» КАК ИНСТРУМЕНТ ПРАВОВОГО РЕГУЛИРОВАНИЯ ПЕРСПЕКТИВНЫХ ЦИФРОВЫХ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

*Аннотация.* В современных условиях для государства и общества большое значение имеет использование научных достижений (новых технологий и продуктов) и, как следствие, их своевременная законодательная регламентация. Оптимальным инструментом в данной деятельности выступает использование «правовых песочниц» (экспериментальных правовых режимов), которые позволяют апробировать использование и правовое регулирование новых технологий и продуктов.

**V.V. Kharin**

Tambov State University named after G.R. Derzhavin  
Tambov, Russia

## **«LEGAL SANDBOXES» (EXPERIMENTAL LEGAL REGIMES) AND SOME ASPECTS OF THEIR IMPLEMENTATION IN RUSSIA**

*Abstract.* In modern conditions, the use of scientific achievements (new technologies and products) and, as a result, their timely legislative regulation is of great importance for the state and society. The optimal tool in this activity is the use of "legal sandboxes" (experimental legal regimes), which allow testing the use and legal regulation of new technologies and products.

В условиях XXI большое внимание уделяется развитию новых технологий и новым технологическим открытиям. Цифровой прогресс (изобретение и внедрение новых технологий и продуктов) стал