

УДК 004.9: 004.415.2

Н. В. МОЗГОВОЙ, В. С. ХАРЧЕНКО

Национальный аэрокосмический университет им. Н. Е. Жуковского «ХАИ», Украина

АНАЛИЗ ХАРАКТЕРИСТИК И ВЫБОР СРЕДСТВ РАЗРАБОТКИ КОМПЬЮТЕРНЫХ ИГР

Статья посвящена разработке метода и приложения для проектно-ориентированного выбора средств разработки компьютерных игр или игровых движков. Актуальность выбранной тематики обоснована на основе современной статистики рынка компьютерных игр. В статье даны определения основных понятий, связанных с предметной областью, выделены и проанализированы основные характеристики современных средств разработки, предложен метод выбора оптимального средства разработки в зависимости от проектных требований. Предложенный метод проиллюстрирован примером выбора оптимального средства разработки из четырёх популярных игровых движков.

Ключевые слова: компьютерные игры, средства разработки, игровые движки, разработка игр.

1. Анализ проблемы и постановка цели исследования и разработки

ключевым фактором, который позволяет заметно уменьшить сложность проекта.

1.1. Актуальность

Развитие информационных технологий последние 25 - 30 лет было тесно и взаимно связано с развитием игровой индустрии. Согласно [1] 87% интернет пользователей играют в различные игры, из которых примерно половина играет каждый день. Рынок компьютерных игр неуклонно растёт. Темпы роста оцениваются в 8% в год при объёме рынка \$88 млрд. При этом объём рынка мобильных игр демонстрирует ежегодный рост в 15% [2].

Распространение мобильных устройств и появление новых средств разработки открывает новые возможности не только для существующих крупных игроков рынка, но также для маленьких студий и индивидуальных разработчиков. Существующие средства разработки предоставляют почти неограниченные возможности для создания игр с великолепной графикой, физикой и искусственным интеллектом.

Однако перед начинающим разработчиком или студией встаёт трудный вопрос, какой инструмент выбрать для разработки игры? От правильного выбора инструмента (так называемого игрового движка) зависит судьба проекта, студии и разработчика. Сам же выбор зависит от множества факторов: начиная от жанра игры, доступного бюджета и заканчивая квалификацией разработчиков. Известно, что большинство игровых проектов провальные [3]. Считается, что основной причиной провалов является недооценка сложности проекта. Однако выбор правильных средств разработки также является

1.2. Мотивация

При разработке любого IT проекта выбор технологии разработки является одной из ключевых проблем, так как этот выбор в значительной мере определяет архитектуру приложения, скорость разработки, масштабируемость системы т.д. При выборе средства разработки игры эта зависимость ещё больше, так как в этом случае избранная технология однозначно определяет используемый подход и доступные для разработчика возможности. В продуктовых и аутсорсинговых компаниях, IT компаниях для решения этой проблемы можно привлечь экспертов. Начиная же независимый разработчик остаётся один на один с десятками всевозможных игровых движков, фреймворков и технологий, каждый из которых может либо помочь с реализацией его идей, либо стать причиной потери времени и средств.

Указанные обстоятельства обуславливают актуальность разработки инструментария для поддержки принятия решений по выбору средств разработки компьютерных игр в зависимости от характеристик проекта и требований заказчика (ожиданий разработчика).

1.3. Обзор литературы

Проблема выбора средств разработки на данный момент является широко обсуждаемой темой в среде разработчиков игр. В тоже время количество публикаций по этой тематике ограничено. Литература, посвящённая разработке игр [3, 4], зачастую

затрагивает фундаментальные вопросы игрового дизайна, такие как: теоретические понятия, типология игр; игровая механика; правила игрового процесса; организационные особенности процесса разработки.

Проблема же выбора средства разработки рассматривается как техническая задача, что можно объяснить быстрыми изменениями игровых технологий и указанных средств. В тоже время актуальная информация, относящаяся к проблеме выбора игрового движка, представлена в известном онлайн журнале *Gamasutra* [5], посвящённом разработке игр, а также в блоге *Digital-Tutors* (часть проекта онлайн-обучения *Pluralsight*), содержащем материалы по созданию программного обеспечения [6]. Стоит отметить, что данная тематика не осталась также без внимания академических исследователей [7].

1.4. Цель и структура

Цель данного исследования – исследование современных средств разработки компьютерных игр на основе анализа множества наиболее важных, по нашему мнению, характеристик, таких как:

1. Доступные режимы рендеринга графики (2D/3D).
2. Наличие звукового движка.
3. Наличие физического движка.
4. Поддержка игрового искусственного интеллекта.
5. Целевые жанры игр.
6. Доступные языки разработки.
7. Полнота и удобство графического интерфейса.
8. Поддерживаемые IDE (среды разработки, от англ. *Integrated development environment*).
9. Поддерживаемые операционные системы для разработки.
10. Поддерживаемые целевые платформы.
11. Лицензия и цена.
12. Качество документации.
13. Уровень развития сообщества.
14. Целевая аудитория (средства разработки могут быть ориентированы для использования как профессионалами, так и аматорами).

На основе данных, полученных в результате анализа современных движков, будет создана база данных, содержащая актуальную информацию о характеристиках каждого из них. В свою очередь, база данных станет основой для приложения по формированию рекомендаций в отношении выбора средства разработки в зависимости от исходных характеристик проекта и предъявляемых требований. Конечная цель – создание приложения для вы-

бора (формирования рекомендаций по выбору) средств разработки в зависимости от проекта.

Структура статьи следующая: во втором разделе дано определение понятия и подробное описание игрового движка и его наиболее значимых характеристик. В третьем разделе в качестве примера приводится анализ четырёх известных современных движков. В четвёртом разделе дано описание предлагаемого метода выбора движка, в пятом – описание разрабатываемого приложения.

2. Анализ характеристик игровых движков

2.1. Понятие игрового движка

Игровой движок (англ. *game engine*) – это программная платформа для создания и разработки компьютерных игр или любых других приложений с графикой, обрабатываемой в реальном времени [8]. Игровой движок, как правило, состоит из следующих компонентов:

1. Графический движок (англ. *graphics engine*) – программный компонент, основной задачей которого является визуализация (рендеринг) двухмерной или трёхмерной компьютерной графики.

2. Физический движок (англ. *physics engine*) – программный компонент, который производит моделирование физических законов в виртуальном окружении.

3. Звуковой движок (англ. *sound/audio engine*) – программный компонент, отвечающий за воспроизведение звука (шумовое и музыкальное оформление, голосов персонажей) в компьютерной игре или другом приложении.

4. Игровой искусственный интеллект (англ. *game artificial intelligence*) – набор программных методик, которые используются в компьютерных играх для создания иллюзии интеллекта в поведении персонажей, управляемых компьютером.

До появления игровых движков новые игры создавались с нуля, будучи оптимизированы под целевую платформу. Необходимость использования подобного подхода обуславливалась ограниченными возможностями аппаратного обеспечения того времени, а также тем фактором, что архитектура аппаратного обеспечения часто менялась.

Некоторые современные игровые движки представляют собой одни из самых сложных, когда-либо созданных приложений и включают в себя многие слаженно работающие подсистемы, позволяющие эффективно использовать доступные аппаратные средства для достижения наиболее полного опыта взаимодействия пользователя с системой (*user experience*).

Непрерывное развитие игровых движков привело к чёткому разделению между созданием графики, логики, дизайном персонажей и уровней. Для каждого этапа применяются отдельные средства разработки.

В своём развитии игровые движки, с одной стороны, стали намного дружелюбнее к пользователю, некоторые даже позволяют обходиться без навыков программирования. С другой стороны, стало более широким их применение. Теперь они применяются не только в индустрии игр, но и для создания приложений визуализации технических процессов, всевозможных симуляторов или просто для создания интерфейса пользователя [9].

2.2. Характеристики игровых движков

Характеристики игрового движка как средства разработки обуславливают область его применения. Значительная их часть определяется компонентами, из которых он состоит. Современный игровой движок является комплексной программной платформой, состоящей из нескольких основных компонентов, каждый из которых может быть как непосредственной частью движка, так и поставляться отдельно. Далее будет рассмотрен каждый из используемых компонентов.

Графический движок (англ. graphics engine) – промежуточное программное обеспечение (англ. middleware), основной задачей которого является визуализация компьютерной графики. Графический движок определяет возможности по отображению компьютерной графики, которые включают в себя:

1. Доступные режимы рендеринга (2D/3D).
2. Поддерживаемые программные интерфейсы (OpenGL/DirectX).
3. Различные графические характеристики: доступные параметры разрешения (количество точек на единицу площади), разрешение и фильтрация текстур, параметры освещения и теней.

Графический движок может существовать как отдельный продукт, но зачастую является частью игрового движка. Главным отличием игрового графического движка является обработка графики в режиме реального времени.

На начальном этапе развития компьютерных игр именно графический движок составлял большую часть игрового движка (90 – 95%). И сейчас, как правило, графические движки не распространяются отдельно от игровых, так как, во-первых, создание большинства игр невозможно без графического движка (исключение – некоторые логические и текстовые игры). Во-вторых, графического движка самого по себе недостаточно для создания игры.

Физический движок (англ. physics engine) – программный компонент, который производит компьютерное моделирование физических законов реального мира в виртуальном мире, с той или иной степенью приближения к реальности. Как и графический движок, игровой физический движок должен работать в режиме реального времени (в отличие, к примеру, от научного физического движка). Однако, в отличие от графических движков, физический движок далеко не всегда является частью игрового движка и часто поставляется отдельно. К примеру, наиболее популярные движки являются полностью независимыми программными продуктами [10]: NvidiaPhysX (26,8% рынка); Havok (22,7% рынка); Bullet Physics Library (10,3% рынка); Open Dynamics Engine (4,1% рынка).

Физические движки могут симулировать следующие физические явления и состояния: динамика абсолютно твёрдого тела; динамика деформируемого тела; динамика жидкостей; динамика газов; поведение тканей; поведение верёвок (тросы, канаты и т.д.). В прошлом физическая симуляция производилась исключительно за счёт ресурсов центрального процессора, однако сейчас физические движки имеют аппаратную поддержку, например, Nvidia PhysX использует ресурсы видеокарт Nvidia.

Важно отметить, что далеко не все игры нуждаются в такой симуляции.

Звуковой движок (англ. sound/audio engine) – программный компонент игрового движка, отвечающий за воспроизведение звука (шумовое и музыкальное оформление, голоса персонажей) в компьютерной игре или другом приложении. Зачастую является частью игрового движка, может использовать следующие известные программные интерфейсы: OpenAL; DirectSound3d; Environmental Audio Extensions (EAX); FMOD.

Игровой искусственный интеллект (ИИИ) (англ. Game artificial intelligence) – программный компонент и/или набор программных методик, которые используются в компьютерных играх для создания иллюзии интеллекта в поведении виртуальных персонажей. В отличие от традиционного в игровом ИИ широко применяются различного рода упрощения, обманы и эмуляции, так как это позволяет: снизить бюджет игры; уменьшить потребление ресурсов системы.

Наличие игрового искусственного интеллекта совершенно необходимо в следующих одиночных играх следующих жанров:

- стратегии как пошаговые, так и реального времени;
- шутеры (англ. shooter – «стрелялка»);
- в ролевых играх (RPG).

В то же время многие игры могут обойтись и без ИИИ, либо он может быть заменён относительно простой эмуляцией, среди них:

- многие многопользовательские игры;
- аркадные игры;
- большинство платформеров;
- головоломки;
- музыкальные игры.

Важно отметить, что игровой искусственный интеллект, должен преследовать цель не обыграть игрока, а красиво проиграть ему.

Целевые жанры игр. Одним из признаков, по которым можно классифицировать игры, является жанр. Жанр игры определяется целью игры и методом её достижения. Критерии принадлежности игры к тому или иному жанру не определены однозначно, и классификация игр недостаточно систематизирована, в разных источниках сведения о жанре могут отличаться.

В то же время существует негласный консенсус среди разработчиков игр, и принадлежность игры к одному из основных жанров можно почти всегда определить однозначно. Зачастую игра принадлежит к одному или нескольким жанрам и только в уникальных случаях игра представляет совершенно новый игровой процесс, который невозможно отнести к известным жанрам.

Основными жанрами компьютерных игр являются:

1. Аркада (англ. Arcade) – игры для игровых автоматов, а также их аналоги для компьютера.

2. Экшн - действие (англ. action) жанр компьютерных игр, в которых успех игрока в большой степени зависит от его скорости реакции и способности быстро принимать тактические решения. Действие таких игр развивается очень динамично и требует напряжения внимания и быстрой реакции на происходящие в игре события.

3. Платформер (англ. platformer) – жанр компьютерных игр, в которых основной чертой игрового процесса является прыгание по платформам, лазанье по лестницам, собирание предметов, обычно необходимых для завершения уровня.

4. Квест (англ. quest), или приключенческая игра (англ. adventure game) – один из основных жанров компьютерных игр, представляющий собой интерактивную историю с главным героем, управляемым игроком.

5. Ролевая игра (RPG – англ. Role Playing Game) – игра, отличительной особенностью которой является наличие у персонажей определённых навыков и характеристик, которые можно обрести, а впоследствии развивать, выполняя какие-либо действия.

6. Стратегия (Strategy) – игра, представляющая собой управление масштабными процессами, как, например, строительство городов, ведение бизнеса, командование армией и т. д. Игровой процесс может идти как в реальном времени (RTS – real time strategy), так и в пошаговом режиме (TBS – turn based strategy).

7. Симулятор (Simulator) – игра, полностью имитирующая какую-либо область реальной жизни, например, имитация управления гоночным автомобилем или самолётом.

8. Головоломка (Puzzle) – игра, полностью или более чем наполовину состоящая из решения различных логических задач и головоломок.

От жанра игры во многом зависит необходимое техническое обеспечение со стороны игрового движка. Игровой движок может иметь специализацию и быть оптимизированным для создания игр определённого жанра, как например:

1. Unreal Engine от Epic Games – 3D Action/RPG.

2. SAGE от EA – стратегии реального времени.

3. Construct2 – двухмерные платформеры, аркады.

Некоторые игровые движки, как Unity3d являются универсальными и могут использоваться для создания как 2D, так и 3D игр любых жанров.

Целевые платформы. Игровой движок определяет доступные целевые платформы, для которых с его помощью могут быть созданы игры, например, проприетарный CryEngine 2, с помощью которого был создан бестселлер Crysis, мог быть использован только для создания игр под платформу Microsoft Windows. Unity и Unreal Engine 4 поддерживают почти все существующие платформы.

Доступные средства разработки и их интеграция. Современные игровые движки зачастую включают в себя интегрированную среду разработки, которая в свою очередь может состоять из многих специализированных редакторов сравнимых по функционалу с полноценными программами [11].

Для большего удобства разработчика графический движок может предлагать интеграцию с существующими профессиональными средствами разработки, например, Unity, Unreal Engine и MonoGame предлагают интеграцию с Visual Studio.

Лицензия – это нетехническая характеристика игрового движка, которая, тем не менее, представляет особый интерес для разработчика. Обычно разделяют две большие группы лицензий на программное обеспечение:

- 1) несвободные (они же проприетарные):

– оплачиваемые лицензии (при этом порой имеется возможность получения исходного кода) – пример Cry Engine 3, Unity;

– условно – бесплатные (shareware) – существует множество моделей лицензий такого рода, например, может существовать бесплатная версия продукта с частично ограниченной функциональностью (Unity, Construct2) либо могут взиматься лицензионные отчисления от прибыли (Unreal Engine);

2) свободные – программное обеспечение распространяется бесплатно и имеет открытый исходный код (например, MonoGame).

Начинающий разработчик зачастую имеет весьма ограниченный бюджет, что в прочем не лишает его возможности от использования профессиональных средств, которые теперь зачастую имеют бесплатную версию. Впрочем, лицензионные отчисления как в случае с Unreal Engine могут сказаться на получаемой прибыли.

Документация. Наличие качественной документации и доступных инструкций является ещё одним ключевым фактором. Многие игровые движки бесплатны, однако далеко не всегда имеют качественную и доступную документацию. Порой важно наличие документации на родном для разработчика языке, даже если он и его команда хорошо владеют английским, качественная документация на родном языке может существенно ускорить знакомство с новым инструментом.

Сообщество. Не менее важным, чем документация, фактором является наличие развитого сообщества разработчиков – пользователей игрового движка. С помощью сообщества (при его наличии) разработчик всегда сможет получить помощь в изучении нового инструмента, а также в разрешении проблем, связанных с нюансами его использования. Наличие развитого сообщества тоже служит показателем того, что инструмент является актуальным и востребованным.

Физический движок. Характеристикой заслуживающего отдельного внимания являются примеры успешных проектов, выполненных с использованием данного игрового движка. Если имеются примеры коммерчески успешных проектов, сделанных с использованием рассматриваемого движка, то на него определённо следует обратить внимание.

3. Анализ игровых движков

Для того чтобы лучше понять критерии выбора проанализируем примеры реальных движков, их характеристик и созданных с их помощью игр.

3.1. Unity

Unity (или Unity3d) – один из самых известных современных игровых движков. Появился в 2005 году, на данный момент последней версией является Unity 5. По состоянию на лето 2013 года количество зарегистрированных пользователей Unity3D достигло 2 миллионов [12], что свидетельствует об огромной популярности движка. Отличительными его особенностями являются:

- продуманная архитектура проекта;
- поддержка всех актуальных целевых платформ;
- широкое сообщество специалистов;
- наличие бесплатной версии, которая почти не имеет ограничений;
- универсальность – возможно создание как 2D, так и 3D игр всех жанров.

Известные игры: Temple Run (2011), Endless Space (2012), Deus Ex: The Fall (2013), Hearthstone: Heroes of Warcraft (2014), Endless Legend (2014), Ori and the Blind Forrest (2015).

3.2. MonoGame / XNA

XNA – игровой движок, созданный Microsoft на основе .Net Framework выпущенный в 2006 году. Позволял создавать игры для всех windows платформ до появления windows 8. В 2013 году Microsoft официально отказались от дальнейшей поддержки проекта. Однако ввиду популярности и простоты XNA была создана её свободная реализация – MonoGame, которая продолжает развиваться и сейчас [13]. Ее особенностями являются:

- относительно низкий порог вхождения;
- интеграция с Visual Studio;
- поддержка основных платформ (включая мобильные).

Известные игры: Magicka (2011), Bastion (2011), Fez (2013), Transistor (2014).

3.3. Unreal Engine

Известный игровой движок с длинной историей, разрабатывается и поддерживается компанией Epic Games. Был выпущен в 1998 году вместе с одноименной игрой, созданной с его помощью. Изначально движок отличался высокой технологичностью, продуманной архитектурой и развитыми средствами разработки. С тех пор на основе Unreal Engine было выпущено сотни игры, большей частью в жанре Action, многие из которых стали бестселлерами. В 2015 году движок стал бесплатным, однако, разработчики игр, должны передавать 5% от прибыли игры компании Epic Games.

К особенностям движка можно отнести:

- мощный редактор, включающий в себе несколько узкоспециальных редакторов;
- передовые технологии рендеринга;
- хорошая переносимость;
- относительно бесплатная лицензия.

Известные игры: Unreal (1998), Серия игр Tom Clancy's Splinter Cell Series (2004 - 2013), серия игр BioShock (2007 - 2013), Might & Magic Heroes VII (2015).

3.4. Construct 2

Construct 2 - относительно новый игровой движок для создания 2D игр, ориентированный на начинающих и инди разработчиков, выпущен в 2011 году. Движок основан на HTML5 и JavaScript, что делает созданные игры доступными почти для всех платформ. Позволяет создавать игры в удобном редакторе и даже не требует навыков программирования. Лицензия платная, но относительно дешёвая. Есть бесплатная ознакомительная версия.

Особенности Construct 2:

- удобный и простой WYSIWYG интерфейс;
- не требует навыков программирования;
- игры доступны для всех основных платформ, в том числе для браузеров;
- хорошая документация, в том числе на разных языках.

К настоящему времени неизвестны игры, разработанные с использованием этого движка, ввиду его новизны.

4. Метод рационального выбора средств

При выборе средств (разработке рекомендаций по выбору) необходимо учесть следующее:

1) требования условно можно разделить на два подмножества:

– обязательные требования, несоответствие которым однозначно исключает движок из выборки (например, разработчик собирается создавать 3D игру, а движок не поддерживает 3D рендеринг);

– необязательные требования, выполнение которых желательно (например, наличие игрового искусственного интеллекта, даже если в движке нет этого компонента, его можно получить от сторонних производителей);

2) некоторые требования присутствуют априори, например: наличие и качество документации; развитое сообщество; история успешного применения.

Для большей гибкости должна предусматриваться возможность для пользователя указать, какие требования считаются обязательными. По умолчанию все требования будут опциональными.

Для обработки опциональных и априорных требований предполагается воспользоваться методом рейтинговой оценки [14]. Это значит, что, если движок соответствует такому требованию, он получает дополнительный балл рейтинга. Нужно заметить, что оценку таких характеристик как документация, сообщество и история применения необходимо произвести заранее. Для каждой из этих характеристик предполагается использовать трехбалльную шкалу оценивания:

1) опциональное требование:

- 0 баллов – если движок не соответствует ему;
- 2 балла – если соответствует;

2) документация:

- 1 балл – почти нет документации в открытом доступе;
- 2 балла – хорошая документация на английском языке;
- 3 балла – документация доступна на родном языке;

3) история применения:

- 0 баллов – нет известных игр;
- 1 балл – 1 – 5 известные игры;
- 2 балла – 5 – 9 известных игр;
- 3 балла – 10 и более известных игр;

4) сообщество, оценка сообщества пока не проработана, рассматривается два варианта его оценки:

- количество зарегистрированных пользователей на форуме производителя;
- релевантность по поисковым движкам / количество найденных результатов поиска.

Алгоритм выбора средств разработки представлен на UML диаграмме деятельности (рис. 1).

Пример использования метода приведён в таблице 1, где обязательные требования обозначены символом ●, а необязательные – символом ○. В примере условный разработчик собирается разрабатывать 3D игру в жанре RPG, ему необходимо наличие физического и звукового движков, игровой ИИ – опционален, в качестве языков разработки ему подходят C# или JavaScript, необходима интеграция с Visual Studio, целевые платформы – Windows и OSX, средство разработки должно быть бесплатным. В результате, исходя из указанных требований, наилучшим выбором являются движки Unity и Unreal.

Задача выбора может быть сформулирована и решена как задача оптимизации. В этом случае в качестве целевой функции следует выбирать (1) при ограничениях (2).

$$z = \sum_{i=0}^n o_i \rightarrow \max \quad (1)$$

$$\begin{cases} r_0 > r_{u0} \\ \dots \\ r_n > r_{un} \\ p \leq p_u \end{cases}, \quad (2)$$

где $o_0 \dots o_n$ – рейтинги характеристик игрового движка, $r_{u0} \dots r_{un}$ – обязательные требования пользователя, $r_0 \dots r_n$ – реальные характеристики движка, которые должны этим требованиям соответствовать (обозначено символом >), p_u – допустимая для пользователя цена на средство разработки и соответственно p – реальная цена средства разработки.

Планируется создать приложение для генерации рекомендаций по выбору средства разработки игр в зависимости от требований пользователя. Приложение будет очень простым, возможные действия представлены на диаграмме прецедентов (рис. 2).

Выводы

В результате исследования и анализа современных средств разработки компьютерных игр получено множество характеристик для их сравнения и выбора. Следует отметить, что в связи с динамичным развитием рынка это множество может обновляться. К числу наиболее стабильных и весомых характеристик отнесены: доступные режимы рендеринга графики (2D/3D); наличие звукового и физи-

ческого движков; поддержка игрового искусственного интеллекта; доступные языки разработки; поддерживаемые целевые платформы; лицензия и цена.

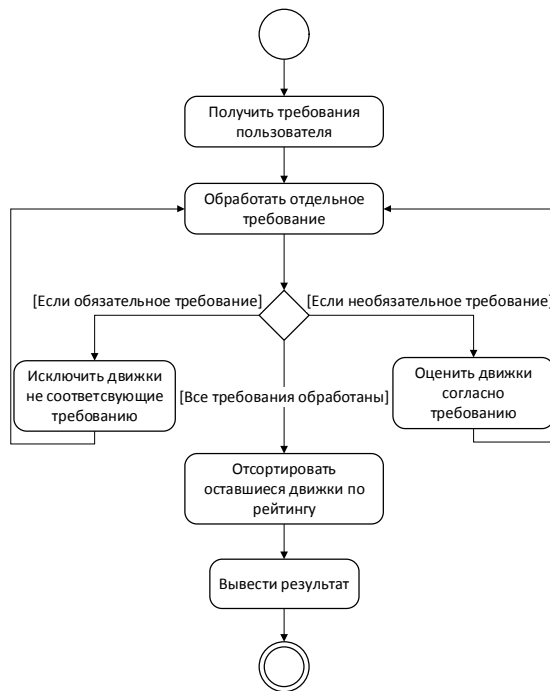


Рис. 1. Диаграмма деятельности с алгоритмом работы приложения

Таблица 1

Пример использования метода выбора средств разработки

Характеристика	Требование	Unity	Unreal	MonoGame	Construct 2
Режим рендеринга	3D	✓	✓	✓	✗
Звуковой движок	•	✓	✓	✓	✓
Физический движок	•	✓	✓	✓	✓
Игровой ИИ	○	✗ (0)	✓ (+1)	✗ (0)	✗
Целевой жанр	RPG	✓	✓	✓	✓
Язык разработки	C# или JavaScript	✓	✓	✓	✓
Интеграция с IDE	Visual Studio	✓	✓	✓	✗
ОС для разработки	Windows	✓	✓	✓	✓
Целевые платформы	Windows, OSX	✓	✓	✓	✓
Цена	Бесплатно	✓	✓	✓	✗
Целевая аудитория	Для профессионалов	✓	✓	✓	✗
Качество документации		3	3	3	3
Уровень сообщества		3	2	1	1
Интерфейс пользователя		3	3	2	3
История применения		3	3	2	0
Результирующий рейтинг		12	12	8	Не подходит

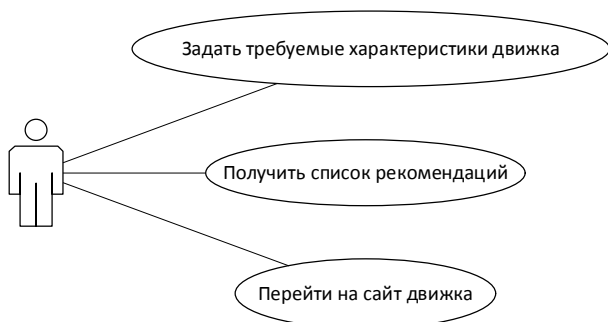


Рис. 2. Диаграмма прецедентов для разрабатываемого приложения

Обоснован алгоритм выбора средств на основе простого принципа рейтингового оценивания.

Основываясь на собранной базе данных с характеристиками современных средств разработки компьютерных игр, а также алгоритмов поиска оптимальных (рациональных) вариантов будет создано приложение для поддержки принятия решений по выбору средств в зависимости от требований проекта. Кроме того, это приложение будет поддерживать процесс формирования множества требования для недостаточно опытных разработчиков игр.

Литература

1. *Игровой рынок в России.* [Electronic resource] / Mail.ru Group. – Access mode: <https://corp.imgsmaill.ru/media/files/igrovoj-rynok-v-rossiimail.ru-group2012.pdf>. – 08.11.2015.
2. *Games software revenues to reach \$110 billion by 2018.* [Electronic resource] / James Brightman. – Access mode: <http://www.gamesindustry.biz/articles/2015-05-04-games-software-revenues-to-reach-usd110-billion-by-2018-digi-capital>. – 08.11.2015.
3. Bethke, E. *Game development and production* [Text] / E. Bethke. – Wordware Publishing, Inc, 2003. – 415 с. – ISBN 1-55622-951-8.
4. Rogers, S. *Level Up* [Text] / S. Rogers. – John Wiley & Sons, Ltd, 2010. – 492 с. – ISBN 978-0-470-68867-0.
5. *Choosing the Perfect Game Engine.* [Electronic resource] / Miko Charbonneau. – Access mode: <http://www.gamasutra.com/blogs/MikoCharbonneau>

/20130222/187185/Choosing_the_Perfect_Game_Engine.php. – 08.11.2015.

6. *Unity, Source 2, Unreal Engine 4, or CryENGINE - Which Game Engine Should I Choose?* [Electronic resource] / Mark Masters. – Access mode: <http://blog.digitaltutors.com/unity-udk-cryengine-game-engine-choose/>. – 08.11.2015.

7. *Comparison and evaluation of 3D mobile game engines.* [Electronic resource] / Akekarat Pattrasitidecha – Access mode: <http://publications.lib.chalmers.se/records/fulltext/193979/193979.pdf>. – 08.11.2015.

8. *Game engine.* [Electronic resource] : Материал из Википедии – свободной энциклопедии : Версия 68823161, сохранённая в 06:59 UTC 24 февраля 2015 / Авторы Википедии // Википедия, свободная энциклопедия. – Электрон. дан. – Сан-Франциско: Фонд Викимедиа, 2015. – Access mode: https://en.wikipedia.org/wiki/Game_engine. – 08.11.2015.

9. *Video games starting to get serious.* [Electronic resource] / Steve Barberich. – Access mode: http://www.gazette.net/stories/083107/businew11739_32356.shtml. – 08.11.2015.

10. *Physics engines survey results.* [Electronic resource] / Real-Time Physics Simulation. – Access mode: <http://bulletphysics.org/wordpress/?p=88>. – 08.11.2015.

11. *Unreal vs. Unity – which engine is better for mobile games.* [Electronic resource] / Salomon Zwecker, Daniel Wilkins, Guido Schmidt. – Access mode: <http://www.makinggames.biz/features/unreal-vs-unity-which-engine-is-better-for-mobile-games,8472.html>. – 08.11.2015.

12. *Another million unity developers in the house.* [Electronic resource] / David Helgason. – Access mode: <http://blogs.unity3d.com/2013/07/09/another-million-unity-developers-in-the-house>. – 08.11.2015.

13. *MonoGame 3.4.* [Electronic resource] / Tom Spilman. – Access mode: <http://www.monogame.net/2015/04/29/monogame-3-4/>. – 08.11.2015.

14. *Метод – балльная оценка.* [Electronic resource] // Большая энциклопедия нефти и газа. – Access mode: <http://www.ngpedia.ru/id146794p1.html>. – 08.11.2015.

Поступила в редакцию 15.11.2015, рассмотрена на редколлегии 18.11.2015

АНАЛІЗ ХАРАКТЕРИСТИК ТА ВИБІР ЗАСОБІВ РОЗРОБЛЕННЯ КОМП'ЮТЕРНИХ ІГОР

М. В. Мозговий, В. С. Харченко

Статтю присвячено розробленню методу та додатку для проектно-орієнтованого вибору засобів створення комп'ютерних ігор або гральних рушіїв. Актуальність обраної тематики обґрунтовано на основі сучасної статистики ринку комп'ютерних ігор. В статті визначено основні поняття, пов'язані з предметною областю, обґрунтовано та проаналізовано основні характеристики сучасних засобів розроблення. Запропоновано метод вибору засобів розробки в залежності від проектних вимог. Запропонований метод проілюстровано прикладом вибору оптимального засобу розробки з чотирьох популярних гральних рушіїв.

Ключові слова: комп'ютерні ігри, засоби розробки, гральні рушії, розроблення ігор.

ANALYSIS OF THE CHARACTERISTICS AND THE CHOICE OF COMPUTER GAME DEVELOPMENT TOOLS

M.V. Mozghoyi, V.S. Kharchenko

The article is dedicated to development of the method and application for project-oriented choice of computer game development tool (i.e. game engine). Relevance of the chosen subject is justified by modern statistics of the video games market. The paper defines the basic concepts related to the application domain. The main characteristics of the game engines are identified and analyzed. The method of choosing game engine depending on project requirements is offered. Offered method is illustrated by an example of choosing optimal development tool from among the four popular game engines.

Key words: computer games, development tools, game engines, game development.

Мозговий Николай Викторович – магистрант каф. компьютерных систем и сетей, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.

Харченко Вячеслав Сергеевич – д-р техн. наук, проф., зав. каф. компьютерных систем и сетей, Национальный аэрокосмический университет им. Н. Е. Жуковского «Харьковский авиационный институт», Харьков, Украина.