

УДК 004

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОГРАММНЫХ СРЕДСТВ ДЛЯ НАЧИНАЮЩИХ РАЗРАБОТЧИКОВ КОМПЬЮТЕРНЫХ ИГР

К. В. Гуделкин, М. М. Игнатьев, В. Д. Некрасова, А. М. Чуйкова

Государственный университет морского и речного флота имени адмирала С. О. Макарова (г. Санкт-Петербург, Российская Федерация)

Выбор оптимального варианта игрового движка является первым важным шагом при создании компьютерной игры. В научной литературе нет определения наиболее подходящего игрового движка. В связи с этим целью данного исследования является анализ преимуществ и недостатков игровых движков и выбор наиболее подходящего из них для начинающих разработчиков компьютерных игр. Изучены и проанализированы самые распространенные движки Godot и Unity. Определен подходящий по критериям игровой движок Godot и представлен процесс разработки программного продукта на нем. Результатом данного исследования является создание рекомендаций для начинающих разработчиков игр.

Ключевые слова: компьютерная игра, игровой движок, языки программирования, информационные технологии, разработка игры, Godot, Unity.

COMPARATIVE ANALYSIS OF SOFTWARE TOOLS FOR NOVICE COMPUTER GAME DEVELOPERS

Kirill V. Gudelkin, Mikhail M. Ignatiev, Valeria D. Nekrasova, Anna M. Chuikova

Admiral Makarov State University of Maritime and Inland Shipping (Saint Petersburg, Russian Federation)

Choosing the optimal variant of the game engine is the first important step in creating a computer game. There is no definition of the most suitable game engine in the scientific literature. In this regard, the work objective is to analyze the advantages and disadvantages of game engines and choose the most suitable one for novice computer game developers. The most common Godot and Unity engines have been studied and analyzed. The Godot game engine is found suitable according to the criteria and the process of developing a software product on it is presented. The result of this research is the creation of recommendations for novice game developers.

Keywords: computer game, game engine, programming languages, information technology, game development, Godot, Unity.

Введение. Сегодня игровая индустрия является одной из быстроразвивающихся и прибыльных отраслей. В сети можно найти немало количество бесплатных игровых движков, подходящих под любые запросы разработчиков. Выбор игрового движка впоследствии будет влиять на весь процесс работы, потому что он является центральным программным компонентом интерактивных приложений с графикой, обрабатываемой в режиме реального времени. Он упрощает разработку и дает игре возможность запускаться на нескольких платформах. Сам термин «движок» означает, что эта программная среда обеспечивает всю внутреннюю работу игр, что позволяет программисту в значительной степени абстрагироваться от технических задач, таких как анимация, звук, локализация, и заниматься творчеством.

Важными критериями для разработчиков, имеющих мало опыта в создании игр, являются простота освоения, интеграции, взаимодействие с другими программами и частота обновлений.

Используя готовый качественный инструмент, программист уже не будет прописывать большую часть базового программного кода, в игровых движках предоставляется базовый набор функций, который можно использовать при создании игры. Благодаря этому разработчик может сконцентрироваться на реализации своей идеи, игровой механики и сюжета, не тратя время на написание кода с нуля. Пользователям также доступны все необходимые учебные материалы и тематические форумы, где обсуждаются трудности в работе с тем или иным движком. Из-за разных потребностей разработчиков игровых программных продуктов создаются различные игровые движки, одни из них просты для освоения, другие позволяют создавать фундаментальные проекты.

В научной литературе не представлены исследования на тему выбора игрового движка для начинающих разработчиков. Поэтому те из них, кто не обладает опытом в создании компьютерных игр, принимают решения, не имея теоретических основ для выбора, и основываются на обыденных знаниях. Целью исследования является определение преимуществ и недостатков рассматриваемых игровых движков и выбор наиболее подходящего из них для начинающих разработчиков компьютерных игр.

Основная часть. Наиболее популярными среди разработчиков являются игровые движки, которые уже доказали свою эффективность при использовании, так как они оказывали большое влияние на результат созданного проекта. Игровые движки имеют разный функционал, вследствие чего правильный его выбор является актуальной проблемой для каждого проекта.

Сравнительный анализ Unity и Godot. Для определения наиболее распространённых игровых движков был проведен опрос. Респондентами стали студенты, занимающиеся программированием. Им был задан вопрос о знакомых или используемых игровых движках. Результаты опроса показали, что наиболее распространёнными игровыми движками являются Unity и Godot (рис. 1).

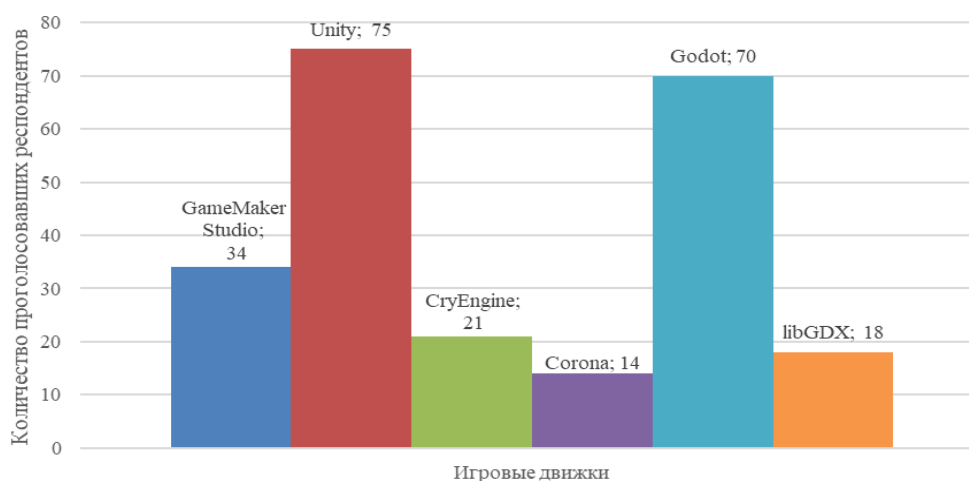


Рис. 1. Распространенность игровых движков

Самым известным движком в мире разработки игр является Unity, выпущенный в 2005 году. Genshin Impact, Cuphead и Hearthstone разработаны с помощью игрового движка Unity, эти и другие проекты представлены на персональных компьютерах, игровых консолях и мобильных платформах, так как разработчики с разным опытом используют данный движок для создания своих проектов [1]. Однако, чтобы его полностью освоить, необходимы большой объем

знаний и обширная практика, но в этом могут помочь большое количество документации и видеоуроки. Кроме того, есть возможность создать прототип без программного кода.

Движок Unity дает возможность создавать полноценные игры. В нем используется компонентно-ориентированный подход к программированию, при котором разработчик создает объекты и присоединяет к ним различные компоненты. Например, объектом является главный персонаж, а компонентами могут быть его визуальное представление и варианты управления им. Из-за функционального графического движка и удобного интерфейса Drag & Drop движок позволяет расставлять объекты, рисовать локации в реальном времени и сразу же тестировать результат [2].

При помощи большой библиотеки ассетов и плагинов, которые можно импортировать и экспортировать, значительно ускоряется разработка проекта. Unity поддерживает множество платформ, технологий, API [3].

С помощью данного игрового движка могут быть реализованы физика твердых тел, ragdoll и тканей, система Level of Detail, коллизии между объектами, сложные анимации. Движок Unity имеет ряд преимуществ, описанных выше, однако есть и недостатки, одним из наиболее важных является низкая скорость работы.

Создание масштабных, сложных сцен с большим количеством компонентов может негативно сказаться на производительности игры, в результате чего разработчикам придется потратить дополнительное время и ресурсы на оптимизацию, а возможно, и удалить некоторые элементы из проекта [4]. Помимо этого, проекты, созданные на Unity, очень много весят: даже самая простая игра, например с пиксельной графикой, может занимать несколько сотен мегабайт.

Игровой движок Godot является относительно новым, он выпущен в 2014 году, но быстро завоевал популярность. Более того, Godot поддерживает ряд распространенных языков программирования для игр, таких как C++, C# и GDScript, встроенный язык сценариев Godot, который был разработан как комбинация двух распространенных и хорошо протестированных языков Lua и Python [5]. Также следует заметить, что Godot производителен, так как использует дизайн, основанный на сценах [6]. Это означает, что всё в игре организовано в сцены и каждая сцена может загружаться и выгружаться по мере необходимости. Это помогает поддерживать низкий уровень использования памяти в игре, что, в свою очередь, способствует более плавной работе. Также это дает возможность запуска игр на малопроизводительных устройствах.

В Godot присутствуют все инструменты для сборки, исправления и тестирования игры. Программное обеспечение для архитектурных частей игры встроено. Для создания визуализации и звуков потребуются дополнительные программы. Godot умеет взаимодействовать со многими IDE, что дает возможность пользователю изменить встроенные инструменты для того, чтобы использовать при создании проекта привычный функционал.

Для создания идей игровой движок имеет 2D и 3D-пространства, в каждом из которых присутствуют стандартные классы, объекты и редакторы скриптов, также имеются редакторы шейдеров для прямого программирования и визуальной настройки [7].

Встроенный IDE имеет автоотступы, подсветку синтаксиса, быстрый доступ к API движка и документам. Если пользователь совершил ошибку при написании кода, то Godot автоматически, без запуска проекта выделит неверную строку.

Как упоминалось ранее, Godot поддерживает GDScript, а он, в свою очередь, состоит частично из Python, поэтому по синтаксису GDScript похож на Python. Примеры кодов на GDScript и Python представлены на рис. 2 и 3.

```

func _physics_process(delta):
    > move()
    > animate()
    > vel.x += gravity * delta
    > vel.y += gravity * delta
    
```

Рис. 2. Функция в GDScript

```

def int_multiply(a, b):
    product = a * b
    return int(product)
    
```

Рис. 3. Функция в Python

При проведении опроса респондентам был задан вопрос с целью выявления наиболее простого языка программирования для начинающих разработчиков из двух предложенных: Python и C#. Результат показан на рис. 4.



Рис. 4. Результат опроса

Судя по результатам опроса, можно сделать вывод, что для респондентов язык программирования Python проще, чем C#, следовательно, GDScript, как упоминалось ранее, состоящий частично из Python, будет проще в освоении синтаксиса, чем C#.

Для сравнения общие и технические характеристики игровых движков Godot и Unity представлены в таблице 1.

Таблица 1

Сравнение Godot и Unity

Общие характеристики		
Особенности	Godot	Unity
Назначение программного комплекса	Это бесплатный кроссплатформенный игровой движок с открытым исходным кодом, что означает, что он может работать как на мобильных	Это лицензированный игровой движок, который используется для создания видеоигр и их элементов

Общие характеристики		
Особенности	Godot	Unity
	устройствах, так и на ПК	
Время создания	14.01.2014	08.06.2005
Легкость в использовании	Да	Нет
Технические характеристики		
ОС	Microsoft Windows, Mac OS, Linux, Free BSD, Open BSD, Android	Microsoft Windows, Mac OS, Linux, Ubuntu и Cent OS
Видеокарта	Drivers with support for OpenGL 3.3	Графические процессоры с поддержкой DX10, DX11 и DX12
Место на жестком диске	60 Мб	2300 Мб
Поддержка русского языка в интерфейсе	Да	Нет
Поддерживаемые языки	C#, C++, GDScript	C#
Встроенный IDE	Да	Нет
Поддержка 3D	Да	Да
Поддержка 2D	Да	Да, но на основе 3D-сцен
Количество внедрений	24,7 тыс.	123 тыс.
Стоимость использования	Бесплатно	Бесплатно, но если игра приносит доход более \$ 100 тыс. в год, то надо делиться последующим доходом с Unity

Преимущества и недостатки. При опросе респондентам был также задан вопрос о наиболее важных особенностях игрового движка. Ответы представлены на рис. 5.

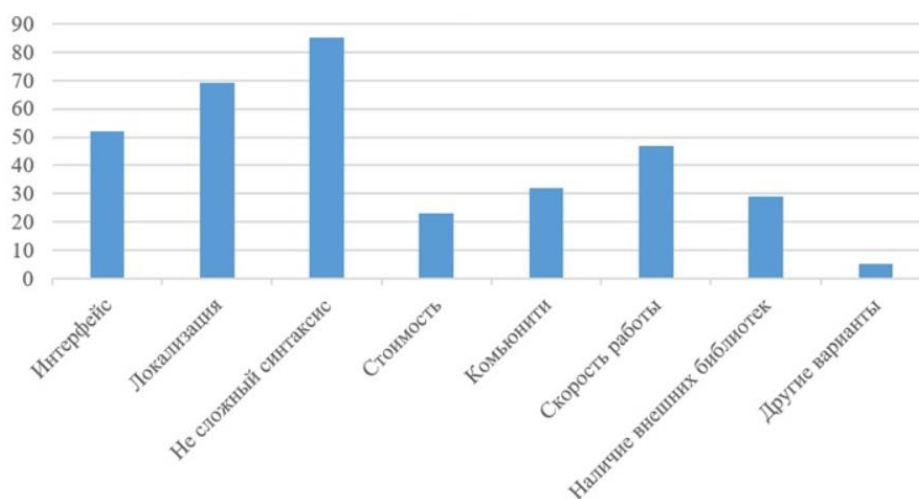


Рис. 5. Результаты опроса об особенностях игрового движка

После анализа полученных ответов была составлена таблица преимуществ и недостатков рассматриваемых игровых движков. Данные представлены в таблице 2. На их основе был сделан вывод, что лучше начинать разработку игр с помощью игрового движка Godot. Для примера на нем разработана игра в жанре симулятор GUMRF GUIDE.

Разработка игры на игровом движке Godot. GUMRF GUIDE — компьютерная игра, созданная для абитуриентов, студентов и выпускников вуза (Государственного университета морского и речного флота имени адмирала С. О. Макарова, ГУМРФ).

Таблица 2

Преимущества и недостатки игровых движков Unity и Godot

	Unity	Godot
Преимущества	1 – быстрое и удобное прототипирование 2 – бесплатная версия для начинающих разработчиков и студентов 3 – кроссплатформенный 4 – возможность разработки больших проектов	1 – прост в освоении 2 – бесплатный 3 – кроссплатформенный 4 – несложный синтаксис языка 5 – занимает мало памяти на жестком диске
Недостатки	1 – медленная работа 2 – ограниченный инструментарий 3 – закрытый исходный код 4 – нестабильность редактора и отладчика	1 – ограниченный инструментарий 2 – не подходит для игр на консолях

В этой игре абитуриенты смогут в игровом формате ознакомиться с тем, как проходит обучение в Институте водного транспорта ГУМРФ. Студенты же имеют возможность посмотреть на себя со стороны, а выпускники — снова оказаться в стенах университета.

Главным героем игры является персонаж по имени Аня. Ей придется выполнять различные задания, с которыми сталкиваются студенты в университете. На рис. 6 для примера представлена одна из сцен игры. После отрисовки всех объектов и анимации была добавлена физика для колонн здания.

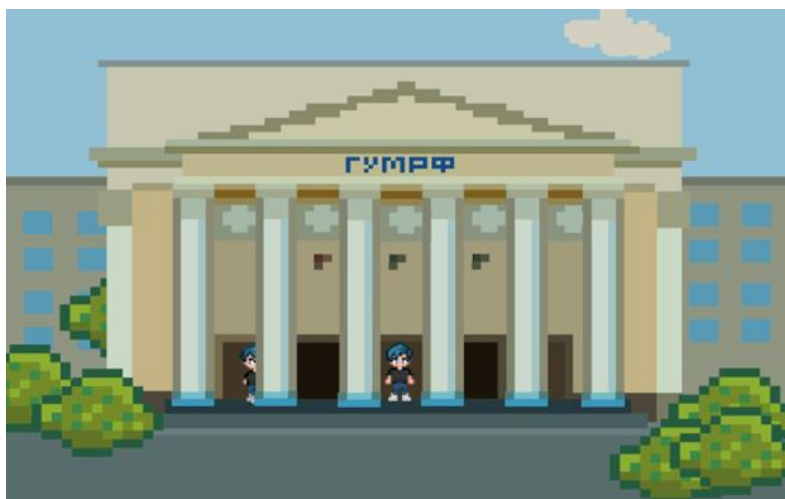


Рис. 6. Пример взаимодействия персонажа и физики

Для того чтобы игрок смог с физической клавиатуры двигать персонаж, был прописан код, представленный на рис. 7.


```
15 ▾ func move():
16 ▾     if Input.is_action_pressed("player_left") and not Input.is_action_pressed("player_right"):
17 ▾         vel.x = -SPEED
18 ▾         if Input.is_action_pressed("player_up"):
19 ▾             vel.y = -SPEED
20 ▾         elif Input.is_action_just_released("player_up"):
21 ▾             vel.y = 0
22 ▾         if Input.is_action_pressed("player_down"):
23 ▾             vel.y = SPEED
24 ▾         elif Input.is_action_just_released("player_down"):
25 ▾             vel.y = 0
26 ▾     elif Input.is_action_pressed("player_right") and not Input.is_action_pressed("player_left"):
27 ▾         vel.x = SPEED
28 ▾         if Input.is_action_pressed("player_up"):
29 ▾             vel.y = -SPEED
30 ▾         elif Input.is_action_just_released("player_up"):
31 ▾             vel.y = 0
32 ▾         if Input.is_action_pressed("player_down"):
33 ▾             vel.y = SPEED
34 ▾         elif Input.is_action_just_released("player_down"):
35 ▾             vel.y = 0
36 ▾     elif Input.is_action_pressed("player_up") and not Input.is_action_pressed("player_down"):
37 ▾         vel.y = -SPEED
38 ▾     if Input.is_action_pressed("player_left") and not Input.is_action_pressed("player_right"):
39 ▾         vel.x = -SPEED
40 ▾     elif Input.is_action_just_released("player_left"):
```

Рис. 7. Связь игрока и персонажа с помощью кода

Godot полностью состоит из узлов сцен, поэтому практически каждая сцена делается по алгоритму, приведенному на рис. 8.



Рис. 8. Алгоритм создания сцены игры

Заключение. Выявив преимущества и недостатки игровых движков Godot и Unity и изучив результаты опроса среди студентов, занимающихся программированием, можно сделать вывод, что Godot является наиболее подходящим игровым движком для начинающих разработчиков компьютерных игр, так как в его интерфейсе можно разобраться быстрее, в отличие от Unity. Еще одно из преимуществ Godot в том, что он занимает очень мало памяти на жестком диске. У Godot респонденты отметили и простой синтаксис языка, так как он состоит из Python.

Библиографический список

1. Клименко, А. В. Сравнительный анализ игровых 3D-движков / А. В. Клименко // Modern science. — 2020. — № 1–1. — С. 401–403.
2. Саяпин, Н. С. Сравнительный анализ сред разработки игровых приложений / Н. С. Саяпин // Мавлютовские чтения. Материалы XV Всеросс. молодежной науч. конфер.: в 7 т. — 2021. — Т 4. — С. 803–806.
3. Сейдаметов, Э. Э. Современные средства разработки игр / Э. Э. Сейдаметов, А. Э. Шабанов // Информационно-компьютерные технологии в экономике, образовании и социальной сфере. — 2020. — № 1(27). — С. 54–56.
4. Абунагимов, А. А. Обзор и сравнение межплатформенной среды для разработки игровых движков / А. А. Абунагимов // Современная школа России. Вопросы модернизации. — 2021. — № 8–2 (37). — С. 174–176.
5. Грузин, Н. А. Сравнение движков для разработки игр: Godot Engine и Unity / Н. А. Грузин // Modern science. — 2021. — № 1–1. — С. 440–444.
6. Дмитриев, В. С. Сравнительный анализ мультиплатформенных движков для разработки игр / В. С. Дмитриев, А. А. Зоткина // Современные технологии: актуальные вопросы, достижения и инновации. Сб. статей XXXIII Междун. науч.-практ. конфер. — Пенза, 2019. — С. 71–73.
7. Пушкарев, Н. Н. Перспектива разработки 3D-конструктора / Н. Н. Пушкарев, М. В. Янаева // Молодой исследователь Дона. — 2022. — №5 (38). — С. 73–76.

Об авторах:

Игнатъев Михаил Михайлович, студент кафедры математического моделирования и прикладной информатики Государственного университета морского и речного флота имени адмирала С. О. Макарова (198035, РФ, г. Санкт-Петербург, ул. Двинская, 5/7), mishuta.maus@mail.ru

Некрасова Валерия Дмитриевна, студентка кафедры математического моделирования и прикладной информатики Государственного университета морского и речного флота имени адмирала С. О. Макарова (198035, РФ, г. Санкт-Петербург, ул. Двинская, 5/7), nekrasovavaleriya@icloud.com

Гуделкин Кирилл Вячеславович, студент кафедры математического моделирования и прикладной информатики Государственного университета морского и речного флота имени адмирала С. О. Макарова (198035, РФ, г. Санкт-Петербург, ул. Двинская, 5/7), gudinikir@gmail.com

Чуйкова Анна Михайловна, студентка кафедры математического моделирования и прикладной информатики Государственного университета морского и речного флота имени адмирала С. О. Макарова (198035, РФ, г. Санкт-Петербург, ул. Двинская, 5/7), chuykova.anna@mail.ru

*About the Authors:*

Ignatiev, Mikhail M., Student, Mathematical Modeling and Applied Informatics Department, Admiral Makarov State University of Maritime and Inland Shipping (5/7, Dvinskaya str., St. Petersburg, 198035, RF), mishuta.maus@mail.ru

Nekrasova, Valeria D., Student, Mathematical Modeling and Applied Informatics Department, Admiral Makarov State University of Maritime and Inland Shipping (5/7, Dvinskaya str., St. Petersburg, 198035, RF), nekrasovavaleriya@icloud.com

Gudelkin, Kirill V., Student, Mathematical Modeling and Applied Informatics Department, Admiral Makarov State University of Maritime and Inland Shipping (5/7, Dvinskaya str., St. Petersburg, 198035, RF), gudinikir@gmail.com

Chuikova, Anna M., Student, Mathematical Modeling and Applied Informatics Department, Admiral Makarov State University of Maritime and Inland Shipping (5/7, Dvinskaya str., St. Petersburg, 198035, RF), chuykova.anna@mail.ru