

**БУТКИНА А. А., КОЧКАЕВА Ю. С., ЧУДНОВА Р. А.**

**РАЗРАБОТКА ОСНОВНОГО ФУНКЦИОНАЛА ИГРОВОГО ПРИЛОЖЕНИЯ ДЛЯ ПЕРСОНАЛЬНЫХ КОМПЬЮТЕРОВ С ИСПОЛЬЗОВАНИЕМ UNREAL ENGINE**

**Аннотация.** В статье описана реализация авторами основного функционала, разработанного на игровом движке «Unreal Engine» тестового приложения, предназначенного для запуска на персональных компьютерах. Разработка выполнялась с использованием визуального блокового языка программирования «Blueprint».

**Ключевые слова:** игровой движок, Unreal Engine 4, функциональные возможности, скрипт, визуальный блоковый язык программирования Blueprint.

**BUTKINA A. A., KOCHKAREVA YU. S., CHUDNOVA R. A.**

**DEVELOPMENT OF THE MAIN FUNCTIONALITY OF GAMING APPLICATION FOR PERSONAL COMPUTERS USING UNREAL ENGINE**

**Abstract.** The article describes the authors' implementation of the basic functionality of a test application developed on the game engine Unreal Engine and designed to run on personal computers. The development was performed using the node-based visual programming language Blueprint.

**Keywords:** game engine, Unreal Engine 4, functionality, script, node-based visual programming language Blueprint.

**Введение.** В настоящее время наблюдается всё возрастающий интерес разработчиков программного обеспечения к сфере разработки компьютерных игр и других интерактивных развлечений. Главным образом это связано с возросшей популярностью игр во время пандемии в 2020 году, когда игры стали своего рода альтернативой другим видам досуга. Однако многие аналитические источники [1] утверждают, что тенденция к росту популярности видеоигр будет сохраняться и в дальнейшем. Так, в 2021 году объем российского рынка видеоигр достиг 177,4 млрд рублей, показав увеличение по сравнению с соответствующими итогами 2020 года на 9% [2]. Поэтому среди разработчиков приложений всё большую актуальность приобретает изучение основных принципов работы в игровых движках, а также создание приложений в них с применением языков визуального программирования.

Таким образом, данное исследование, посвященное рассмотрению реализации тестового игрового приложения, предназначенного для персональных компьютеров, на визуальном языке программирования с применением в качестве среды разработки игрового движка является актуальным.

Для достижения этой цели были поставлены и решены следующие задачи:

- выполнить анализ наиболее популярных игровых движков и на его основе сделать выбор оптимальной среды разработки;
- рассмотреть особенности использования языков визуального программирования;
- реализовать основные функции разрабатываемого приложения.

**Анализ игровых движков.** Игровой движок – особый класс программного обеспечения, который используется в качестве среды разработки игровых приложений. Современные разработчики видеоигр чаще всего используют следующие игровые движки: CryEngine, Unity, Unreal Engine. Каждый из них имеет свои особенности, но в тоже время существуют и общие черты, которые позволяют создавать проекты высокого уровня. Авторами были изучены основные принципы работы в каждой из данных сред разработки, а также проведён их сравнительный анализ, основные результаты которого приведены в таблице 1.

Таблица 1

#### Сравнение игровых движков

	<b>CryEngine</b>	<b>Unity</b>	<b>Unreal Engine</b>
Лицензия	Условно- бесплатная	Условно- бесплатная	Условно-бесплатная
Язык программирования	C++	C#	C++
Визуальная система программирования	Flow Graph	Bolt – требуется покупка	Blueprint
Внедрение 3D моделей в движок	+	+	+
Библиотека ассетов	Маленькая	Большая	Большая
Оптимизация	+	-	+/-
Кроссплатформенность	Windows, PS, XBOX	Windows, IOS, Android	Windows, IOS, Android, PS, XBOX
Тестирование в реальном времени	+	+	+
Дополнительное ПО для работы	-	+	-
Учебные пособия	-	+	+
Язык интерфейса	Английский	Английский	Английский
Техническая поддержка	-	+	+

Таким образом, в результате проведенного сравнительного анализа в качестве наиболее подходящего для разработки тестового приложения был выбран игровой движок «Unreal Engine», являющийся безоговорочным лидером среди всех аналогов.

**Unreal Engine.** В частности, «Unreal Engine» был выбран авторами благодаря своей доступности (является условно-бесплатным) и большим количеством возможностей, за которые в других движках пришлось бы доплачивать, например, за визуальный блоковый язык программирования «Blueprint», который облегчает и ускоряет процесс написания кода и не требует особых знаний C++, языка программирования, поддерживаемого для написания скриптов в «Unreal Engine». Кроме того, данный движок имеет достаточно большое количество учебных пособий и возможность обратиться в техподдержку по решению возникшей проблемы или за получением дополнительной информации [3].

Следует учитывать, что для полноценной стабильной работы в данной среде разработки на компьютере необходимо выполнение следующих системных требований для операционных систем семейства Windows [4]:

- операционная система: Windows 10 64-bit;
- процессор: Intel или AMD 4 ядра и более, тактовая частота 2,5 ГГц или выше;
- оперативная память: 8 ГБ;
- видеокарта: совместимая с DirectX 11 или DirectX 12;
- 22 ГБ свободного места на жёстком диске.

Для написания функциональных скриптов программа «Unreal Engine» поддерживает два языка программирования: «C++» и «Blueprint». Для разработки программы авторами было принято решение использовать язык программирования «Blueprint».

**Blueprint.** «Blueprint» – строго типизированный визуальный скриптовый язык, с помощью которого можно описать логику приложения любой сложности без использования языков программирования. Другими словами, это тот же C++, но в более удобном блочном виде [5]. Главным плюсом такого языка является простота при его использовании. Вместо написания кода предлагается возможность составлять системы из логических блоков. С помощью такой системы можно «собрать» программу любой сложности, главное – понимать основные принципы объектно-ориентированного программирования [6].

Однако, у данного языка программирования также есть и недостатки. В частности, одним из его недостатков является громоздкость, т.к. при составлении простого действия необходимо использование нескольких блоков, а при наличии большого количества операций система быстро становится слишком нагруженной и трудной для восприятия.

**Реализованный функционал приложения.** Авторами были реализованы следующие основные функциональные возможности разрабатываемого приложения, позволяющие пользователю осуществлять управление главным персонажем.

- 1) Управление игровым персонажем

В первую очередь перед созданием логики передвижения персонажа в игровом движке были определены и настроены клавиши, которые при нажатии вызывают определённое действие (рис. 1).

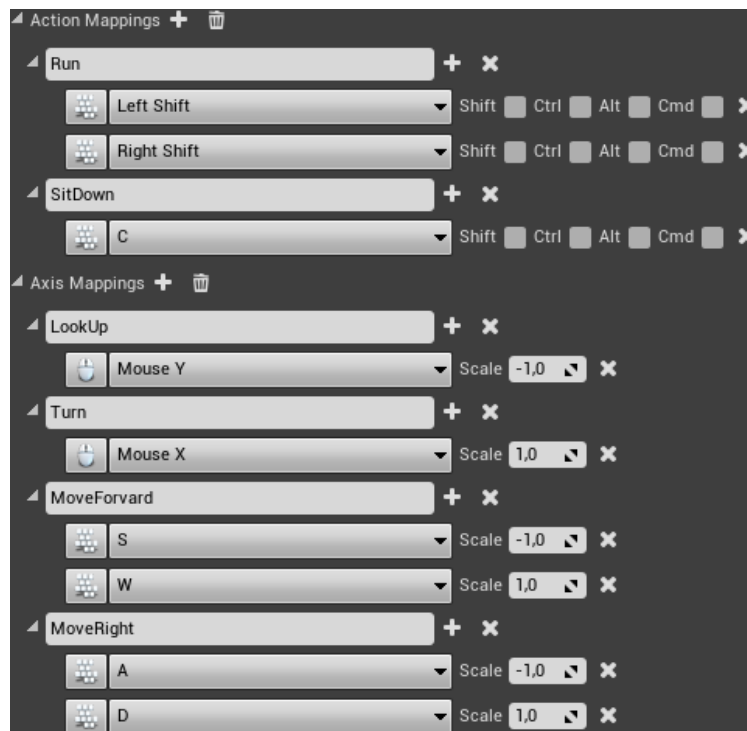


Рис. 1. Выбор и настройка клавиш.

Таким образом, персонаж будет передвигаться вперёд/назад при нажатии на клавиши «W»/«S», а также вправо/влево при нажатии на клавиши «D»/«A». Считывание сигнала с клавиатуры реализовано следующим образом (рис. 2).

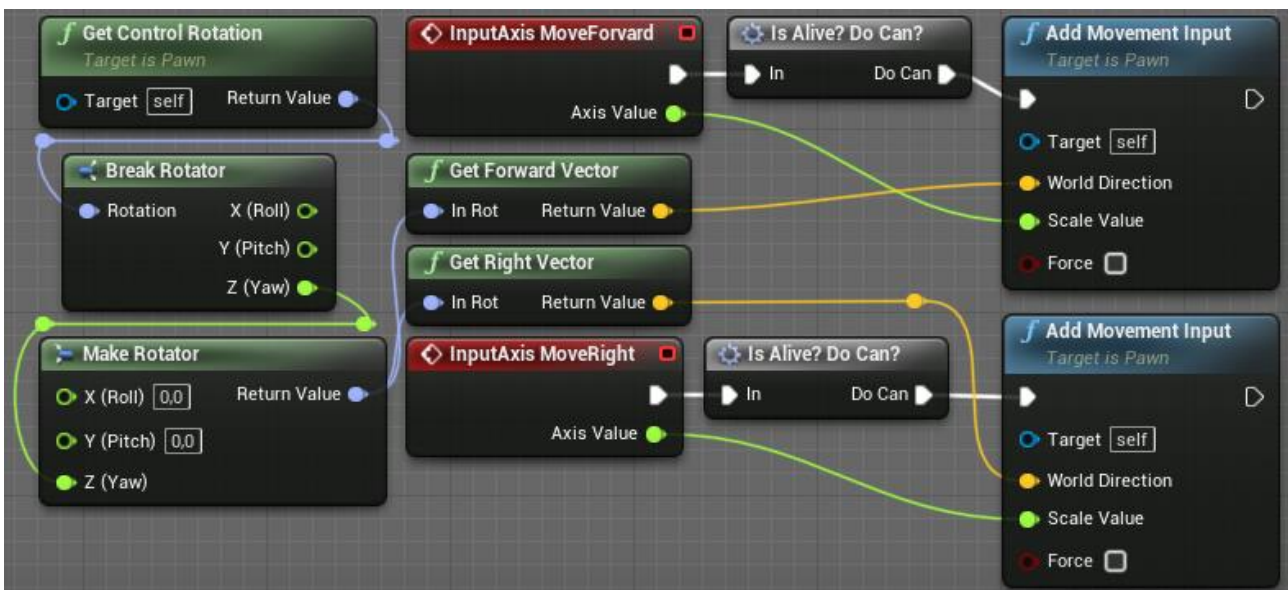


Рис. 2. Модуль логики управления главным персонажем.

Для более реалистичного игрового процесса была добавлена возможность вращения камерой с помощью мыши. Вектор X отвечает за поворот влево/вправо, вектор Y - за поворот вверх/вниз. В данном случае нам нужно получать текущие значения осей мыши и вращать камеру в зависимости от её движения (рис. 3).



Рис. 3. Модуль логики управления камерой.

Для бега и передвижения персонажа в положении «полуприсед» авторами были реализованы отдельные модули логики (рис. 4, 5). Основная логика построения блоковой системы для функции «бег» состоит в следующем: изначально необходимо определить, какова скорость персонажа. Далее, при нажатии и удерживании любой клавиши Shift, задаётся значение скорости бега/шага персонажа путём прибавления/вычитания ускорения к полученному на предыдущем шаге результату. Передвижение в положении «полуприсед» происходит после получения сигнала о смене положения персонажа, т.е. после нажатия на клавишу С, при этом скорость передвижения самого персонажа становится медленнее.

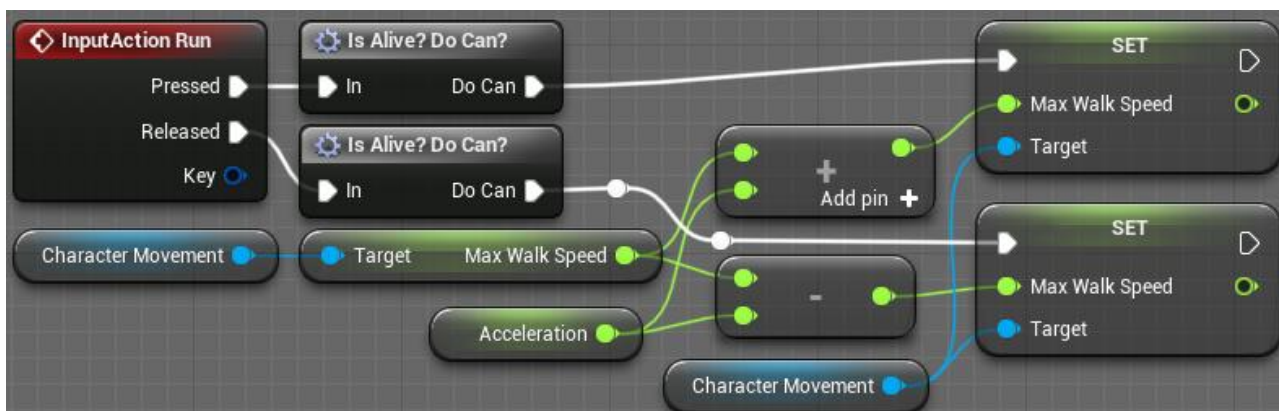


Рис. 4. Модуль логики, отвечающий за быстрое передвижение персонажа.

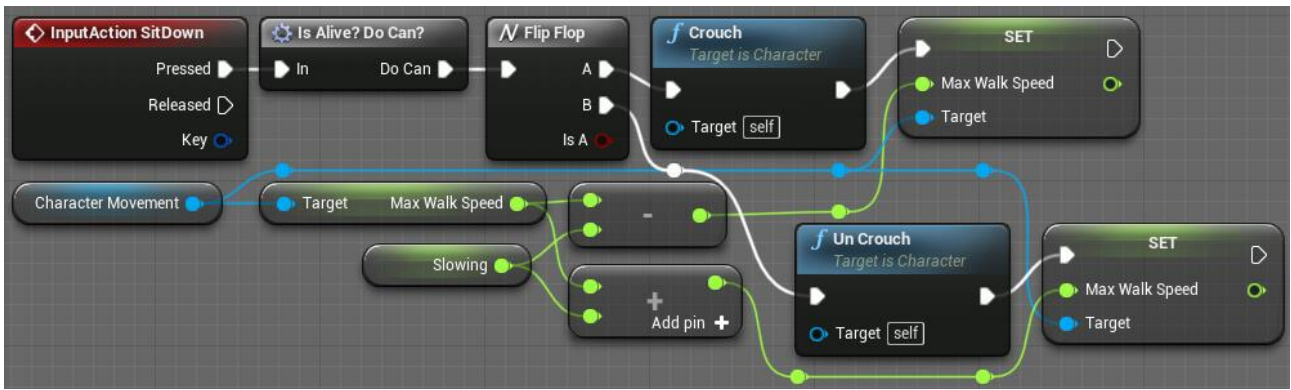


Рис. 5. Модуль логики, отвечающий за передвижение персонажа в положении «полуприсед».

## 2) Виджет шкал жизненных показателей

Для визуального представления специальных численных показателей «Выносливость» и «Рассудок», реализуемых в тестовом игровом приложении, авторами используется виджет. Виджет (widget) представляет собой элемент графического интерфейса пользователя, используемый в программе «Unreal Engine» для создания человеко-машинного интерфейса программы. Для этого первоначально создаётся дизайн виджета, в который добавляются все необходимые компоненты из встроенных в игровой движок элементов, например, «Полоса прогресса», «Текстовое поле» и т.п. (рис. 6). Далее с помощью языка «Blueprint» реализуется логика работы виджета, т.е. добавляются отклики на определённые действия (рис. 7–9). На заключительном этапе устанавливается событие, позволяющее игроку открыть определённый виджет в процессе работы приложения (рис. 10).

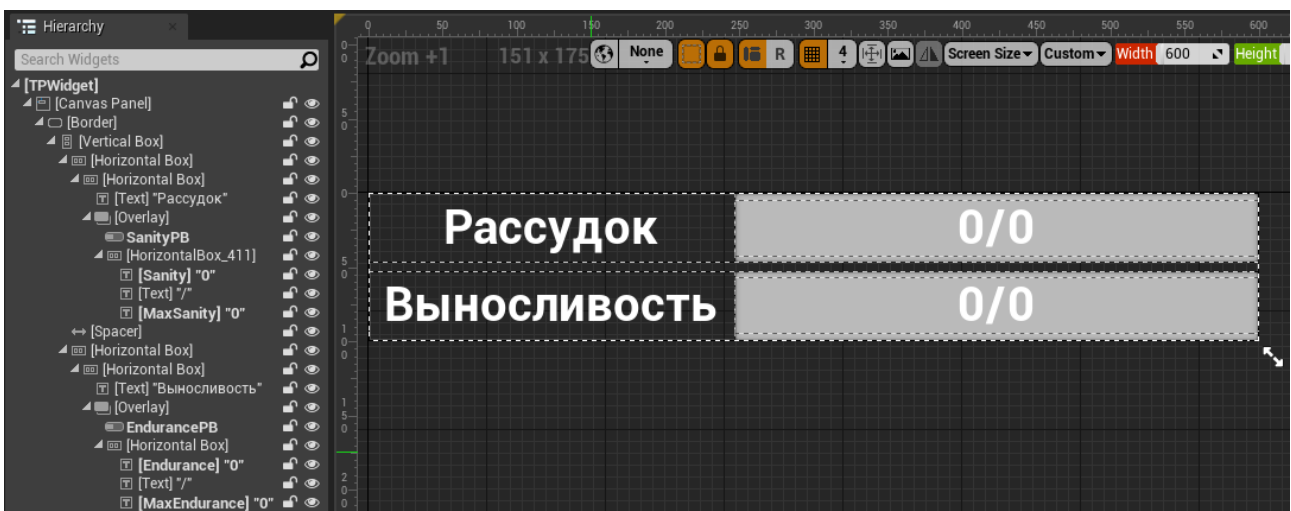


Рис. 6. Создание интерфейса виджета «Шкалы жизненных показателей».

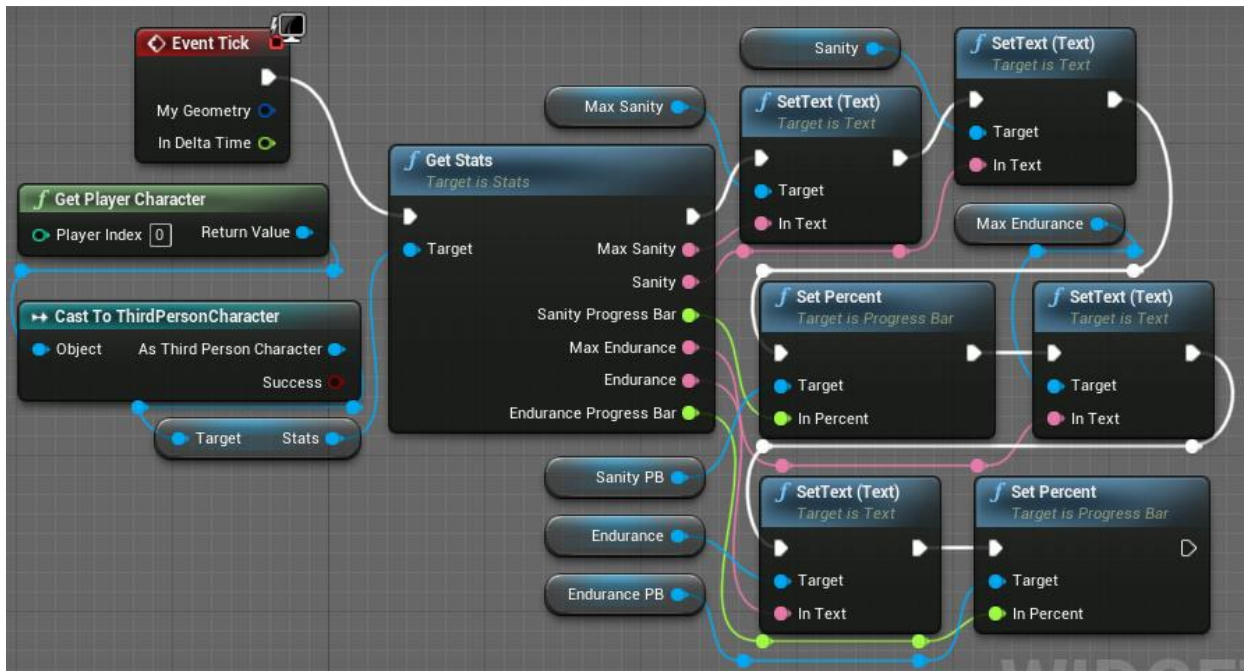


Рис. 7. Модуль логики передачи значений жизненных показателей в виджете «Шкалы жизненных показателей».

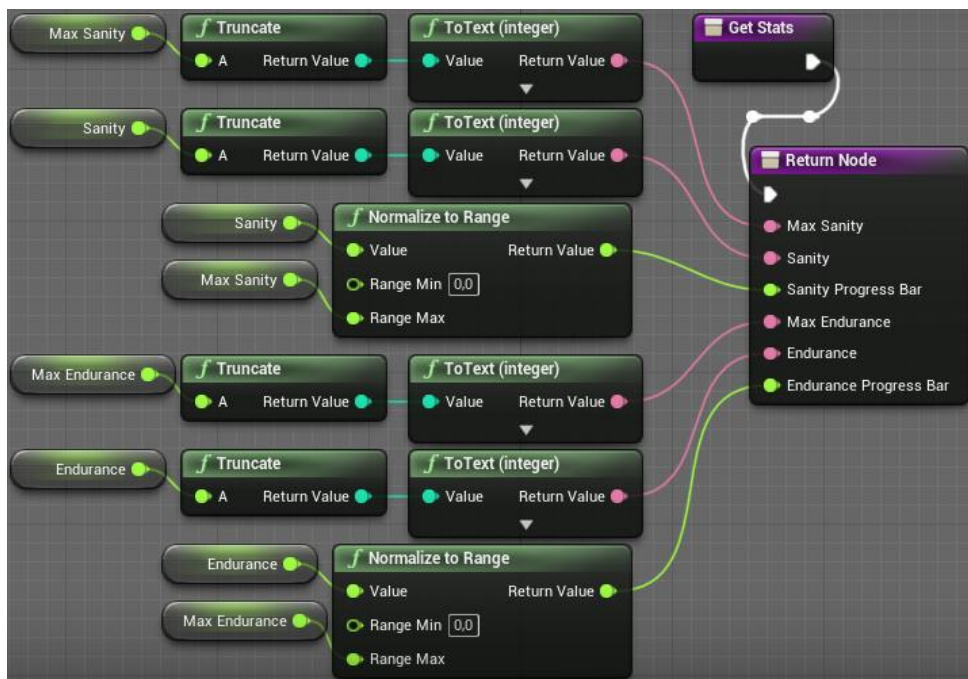


Рис. 8. Модуль логики получения значений жизненных показателей.

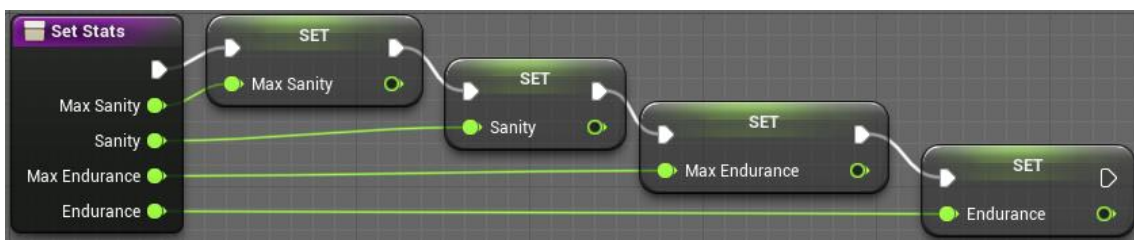


Рис. 9. Модуль логики записи значений жизненных показателей.



Рис. 10. Настраиваемое событие для виджета «Шкалы жизненных показателей».

### 3) Отображение жизненных показателей главного персонажа

Во многих играх выносливость персонажа зависит от скорости прохождения уровней, причём под скоростью подразумевается скорость шага/бега персонажа. Так в разрабатываемом тестовом приложении при передвижении персонажа в режиме «бег» теряется выносливость. Когда её количество достигает отметки «0», персонаж перестаёт передвигаться в режиме «бег». Количество выносливости автоматически восстанавливается при условии, если персонаж передвигается обычным шагом (рис. 11).

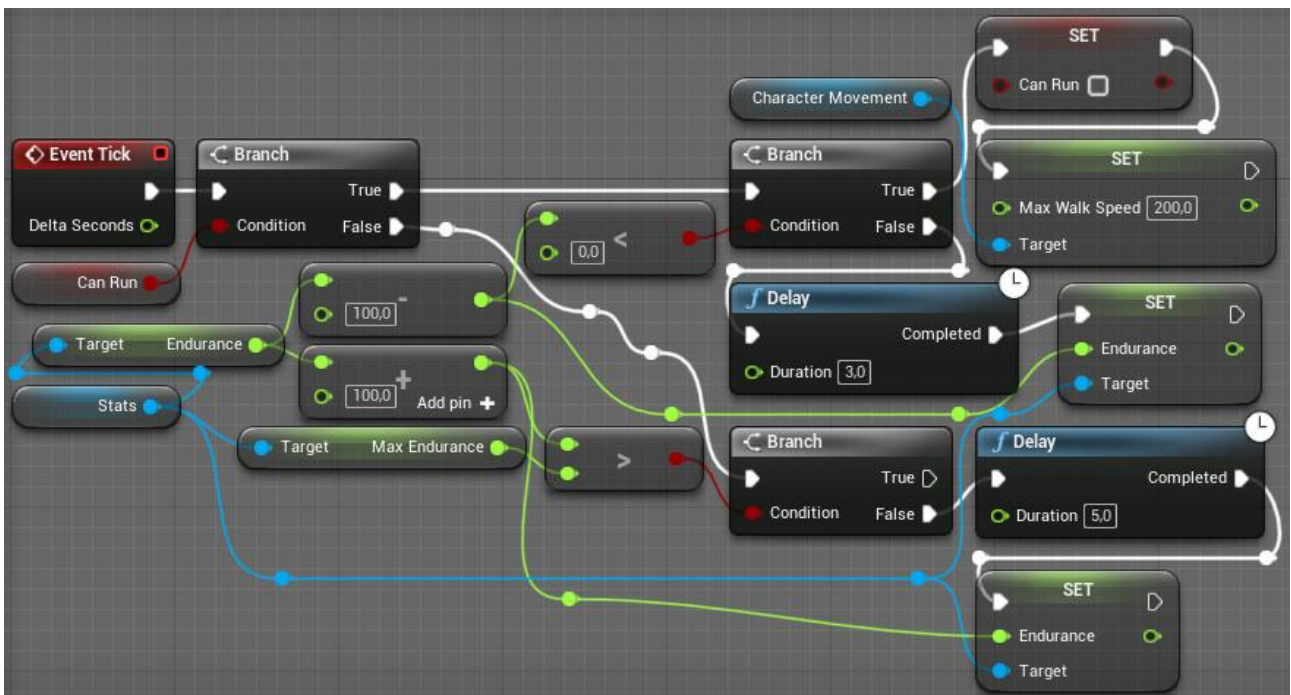


Рис. 11. Модуль логики для реализации игрового механизма «выносливость».

Принцип работы механизма «рассудок» в рассматриваемом приложении аналогичен механизму «здоровье» персонажа в подобных игровых программах. Была реализована следующая логика рассматриваемой опции: после столкновения персонажа с неигровыми персонажами (получение урона) значение рассудка снижается (рис. 12), в случае достижения шкалой «рассудок» отметки «0» – персонаж погибает (рис. 13).



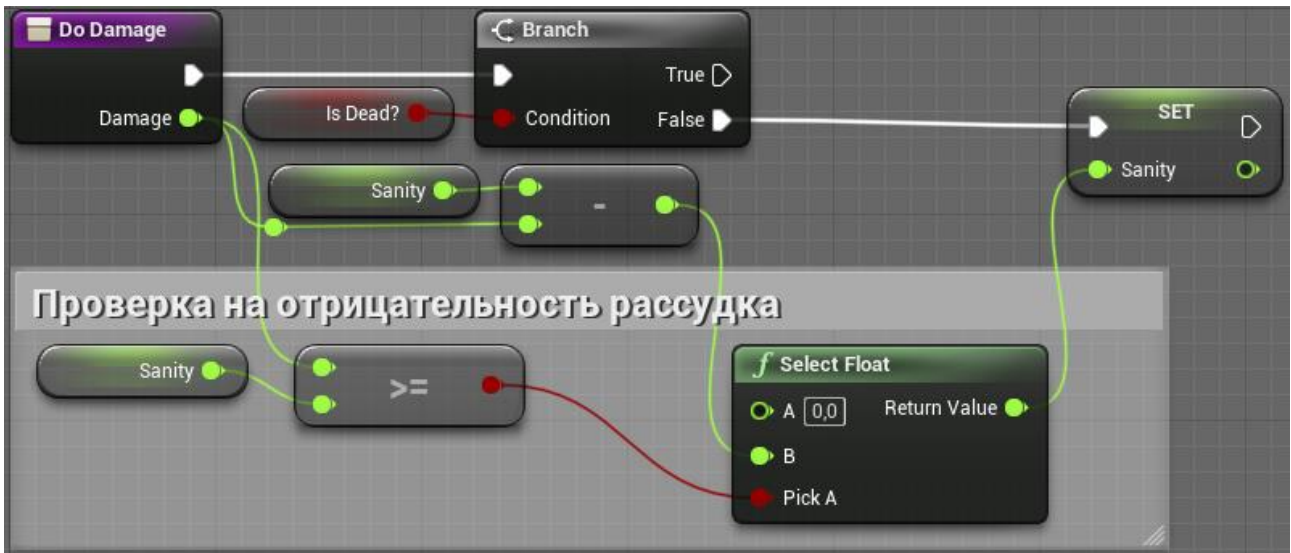


Рис. 12. Модуль логики для нанесения урона персонажу.

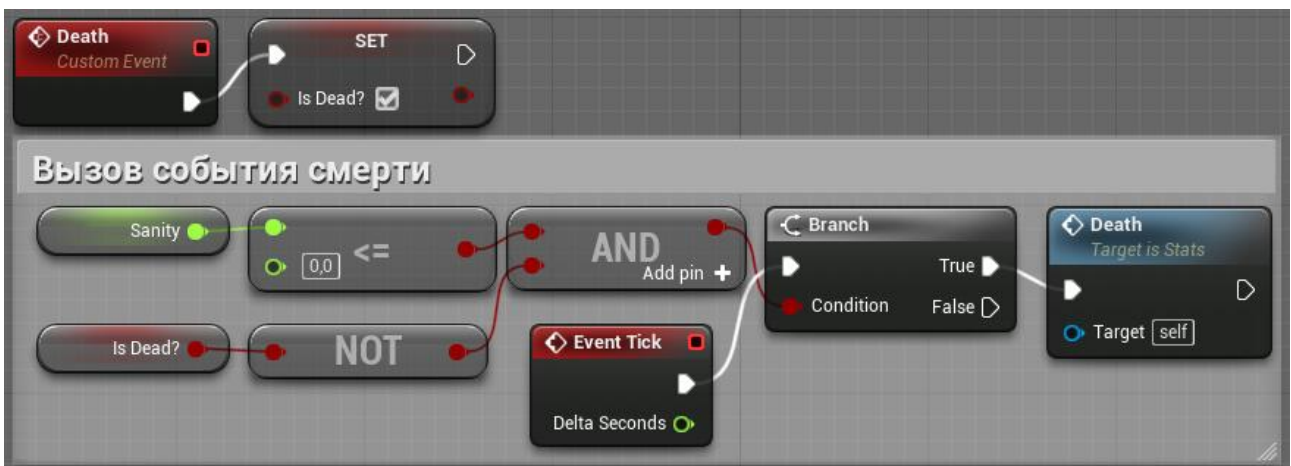


Рис. 13. Модуль логики вызова события смерти персонажа.

Мёртвый персонаж не может совершать каких-либо действий, поэтому авторами была создана функция для предотвращения таких ситуаций. Данная функция проверяет состояние персонажа и в случае его смерти блокирует любые его движения (рис. 14).



Рис. 14 Модуль логики блокировки любых действий персонажа в случае его смерти.

#### 4) Разрушение объектов

Иногда пользователю необходимо будет найти определённые предметы в реализуемом приложении для дальнейшей работы в нём. Для усложнения поиска данные предметы будут спрятаны внутри других, например, в вазе. Чтобы достать их, персонаж должен разбить данную вазу или, другими словами, разрушить объект.

Чтобы при взаимодействии с определённым объектом главный герой мог его разрушить, первоначально потребовалось создать и настроить разрушаемый объект (рис. 15, красным прямоугольником обозначен триггер разрушения объекта, жёлтым прямоугольником – количество частей, на которые распадётся объект при столкновении). Далее авторами была реализована логика, позволяющая объекту, разрушиться при взаимодействии с персонажем (рис. 16).

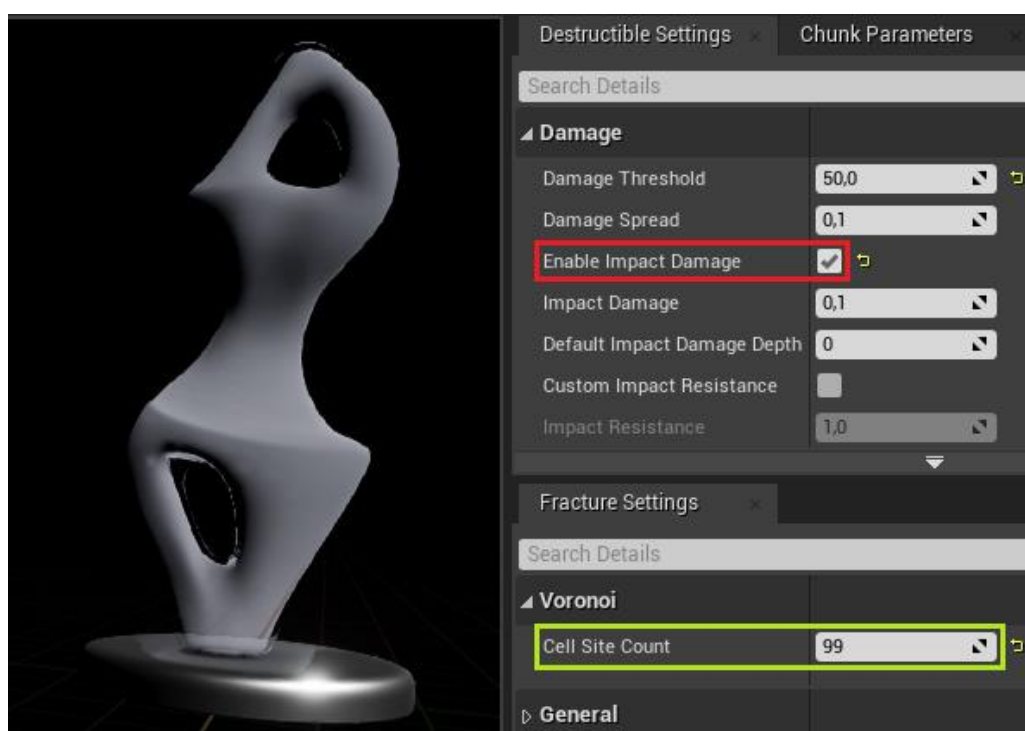


Рис. 15. Создание и настройка разрушаемого объекта.

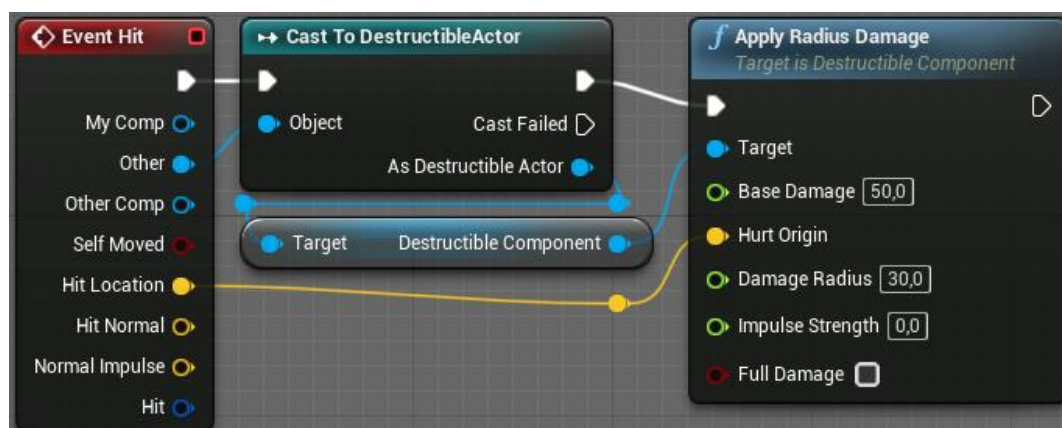


Рис. 16. Модуль логики разрушения объекта при столкновении с главным персонажем.

**Заключение.** Все задачи, поставленные в рамках данного исследования, были решены. Результатом работы является разработанное тестовое игровое приложение с описанным функционалом, позволившее в процессе своей реализации изучить основные принципы работы в игровом движке «Unreal Engine», а также особенности программирования на визуальном языке программирования «Blueprint».

#### СПИСОК ЛИТЕРАТУРЫ

1. Объемы игрового рынка в 2022 году могут превысить 200 млрд долларов [Электронный ресурс]. – Режим доступа: <https://club.dns-shop.ru/digest/obemyi-igrovogo-ryinka-v-2022-godu-mogut-prevyisit-200-mlrd-dolla/> (дата обращения: 02.11.2022).
2. Компьютерные и видеоигры (российский рынок) [Электронный ресурс]. – Режим доступа: [https://www.tadviser.ru/index.php/Статья:Компьютерные\\_и\\_видеоигры\\_\(российский\\_рынок\)](https://www.tadviser.ru/index.php/Статья:Компьютерные_и_видеоигры_(российский_рынок)) (дата обращения: 02.11.2022).
3. 5 лучших программ для создания собственной игры [Электронный ресурс]. – Режим доступа: <https://geeker.ru/games/programmy-dlya-sozdaniya-igr> (дата обращения: 02.11.2022).
4. Hardware and Software Specifications [Электронный ресурс]. – Режим доступа: <https://docs.unrealengine.com/4.27/RecommendedSpecifications/> (дата обращения: 02.11.2022).
5. Unreal Engine 4 [Электронный ресурс]. – Режим доступа: <https://habr.com/ru/post/249965/> (дата обращения: 02.11.2022).
6. Основные приёмы Blueprint-программирования [Электронный ресурс]. – Режим доступа: <https://it-cube48.ru/основные-приёмы-blueprint-программирования/> (дата обращения: 02.11.2022).