

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»



Направление подготовки/специальность
09.03.01 Информатика и вычислительная техника

направленность (профиль)/специализация
«Технологии разработки программного обеспечения»

Выпускная квалификационная работа

Разработка обучающего игрового приложения на базе Unreal Engine

Обучающегося 4 курса
очной формы обучения
Козырькова Игоря Викторовича

Руководитель выпускной
квалификационной работы:
д.п.н., профессор
Власова Елена Зотиковна

Рецензент:
Ученая степень (*при наличии*), ученое
звание (*при наличии*), должность
Ф. И. О. (*указывается в именительном
падеже*)

Санкт-Петербург

2023

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
ВВЕДЕНИЕ	3
ТЕОРИТИЧЕСКАЯ ЧАСТЬ	7
Глава 1. Анализ образовательной среды	7
1.Анализ методов обучения детей начальных классов с целью оценки возможности применения геймификации.	7
2.Оценка функциональности и эффективности Unreal Engine 5 в контексте образования.	11
ПРАКТИЧЕСКАЯ ЧАСТЬ	15
Глава 2 Проектирование обучающего игрового приложения «Математика в игре»..	15
1.Назначение разработки:	15
2.Стадии и этапы разработки:	15
2.1. Создание и проработка идеи приложения	15
2.2. Выбор движка	18
2.3. Реализация графического компонента игры.....	19
2.4. Разработка программного кода игры	27
2.5 Реализация музыкальной/звуковой составляющей проекта	33
2.6. Тестирование	34
3.Требования к программе или программному изделию	36
3.2. Требования к информационной и программной совместимости	38
3.3. Минимальные системные требования.....	38
4. Оценка вариативности использования разработанных систем.	38
4.1. Оценка вариативности при использовании арифметических операций:	38
4.2. Оценка вариативности с точки зрения дизайна уровней:	39
4.3. Оценка вариативности с точки зрения различных игровых механик:	40
4.4. Оценка вариативности с точки зрения игрового опыта:	41
ЗАКЛЮЧЕНИЕ	43
СПИСОК ЛИТЕРАТУРЫ	45
ПРИЛОЖЕНИЕ А:	48

ВВЕДЕНИЕ

Актуальность темы:

Создание обучающих приложений на базе Unreal Engine имеет большую актуальность в настоящее время. Данный игровой движок является одним из наиболее популярных игровых движков, который используется для создания игр, а также для разработки различных виртуальных сред, например, тренажеров для военных, авиационных, медицинских и других специальностей. Приложения на базе Unreal Engine могут использоваться в различных сферах, таких как образование, медицина, производство и другие. Эти приложения могут помочь студентам, медицинским работникам, рабочим, школьникам и другим специалистам приобретать необходимые навыки и знания в безопасной и контролируемой среде.

Кроме того, данный движок обладает высоким качеством графики и анимации, что делает обучающие приложения более интерактивными и увлекательными. Также возможности Unreal Engine позволяют создавать различные типы контента, включая визуальные эффекты, моделирование физических явлений и т.д.

Актуальной является задача разработки обучающего игрового приложения для детей начальных классов по нескольким причинам.

Во-первых, многие дети в настоящее время проводят много времени за компьютером и играют в видеоигры. Создание таких приложений может использовать эту привычку и превратить ее в полезный инструмент для обучения.

Во-вторых, данный продукт может помочь детям начальных классов в освоении базовых навыков, таких как чтение, письмо, математика и т.д. Дети могут обучаться с более высокой заинтересованностью к предмету тем самым усваивая материал гораздо лучше.

В-третьих, такие приложения могут быть персонализированы и адаптированы к индивидуальным потребностям и уровню знаний каждого ребенка, что может помочь им улучшить свои учебные результаты.

Геймификация в современном образовании актуальна, так как повышает мотивацию и интерес учащихся, развивает навыки и компетенции, индивидуализирует обучение, формирует навыки сотрудничества и снижает страх перед ошибками. Она также использует современные технологии, делая обучение более интерактивным и привлекательным. В итоге, геймификация создает стимулирующую обучающую среду, способствующую эффективному обучению и достижению успеха учащимися.

Предметом исследования является непосредственно разработка обучающего игрового приложения на базе Unreal Engine 5

Теоретическая значимость заключается в изучении уровня эффективности обучения и ее оценка в контексте игровой среды, изучение воздействия игровой механики на усвоение материала и развитие навыков. Также стоит рассмотреть педагогические аспекты обучения: разработка методики обучения, адаптация обучения к потребностям и возможностям пользователей, изучение эффективности обратной связи и мотивации игроков.

Практическая значимость разработки обучающего игрового приложения на базе Unreal Engine для детей начальных классов заключается в следующем:

Улучшение качества образования: использование новых технологий и инновационных подходов может значительно повысить качество образования, что может привести к улучшению успеваемости детей и увеличению мотивации к обучению.

Повышение интереса к обучению: использование игровых элементов и заданий в обучении помогает повысить интерес и мотивацию детей к обучению, что может улучшить результаты их обучения.

Научная новизна проекта "Разработка обучающего игрового приложения на базе Unreal Engine" проявляется в нескольких аспектах:

1.Использование Unreal Engine в образовательных целях: Применение данного игрового движка в создании обучающего приложения представляет новаторский подход, так как Unreal Engine преимущественно известен своим использованием в разработке видеоигр. Интеграция этого мощного инструмента в обучение открывает новые возможности для создания интерактивных и захватывающих образовательных сценариев.

2.Инновационные подходы к обучению: Проект может представлять новые и творческие подходы к обучению, такие как геймификация образования, интерактивные симуляции, виртуальная реальность и другие. Это включает разработку новых образовательных сценариев, методов оценки и обратной связи, а также использование адаптивного обучения и индивидуализации для оптимизации образовательного процесса.

3.Применение последних разработок и функций Unreal Engine: Если проект использует новейшие возможности и функции, доступные в последних версиях Unreal Engine, он обретает научную новизну. Например, использование реалистической графики, физических симуляций, искусственного интеллекта или расширенной реальности с использованием Unreal Engine может быть инновационным в контексте обучающих приложений.

Целью выпускной квалификационной работы стала разработка обучающего игрового приложения для детей начальных классов на базе игрового движка Unreal Engine 5.1.А также рассмотрение возможностей использования данного игрового движка с целью обучения различных возрастных групп различных профилей подготовки.

Для достижения поставленных целей необходимо решить следующие задачи:

1. Проанализировать и синтезировать знания о предметной области: изучение методов обучения и рассмотрение наиболее эффективных.
2. Спроектировать основную идею обучающего игрового приложения на основе изученного материала.
3. Рассмотреть основные возможности Unreal Engine 5.1
4. Реализовать приложение на базе Unreal Engine 5.1

Высокая степень практической значимости заключается не только в непосредственном использовании разработанного приложения при обучении, а также высокую вариативность при разработке, так как механики и системы реализованные при разработке приложения могут быть использованы под как под другие возрастные группы, так и для обучения по другим дисциплинам.

Результатом бакалаврской выпускной квалификационной работы является разработанное обучающее игровое приложение на движке Unreal Engine, а также оценка возможностей и обучающего потенциала движка на основе реализованных механик.

ТЕОРИТИЧЕСКАЯ ЧАСТЬ

Глава 1. Анализ образовательной среды.

1. Анализ методов обучения детей начальных классов с целью оценки возможности применения геймификации.

В начальной школе ученикам бывает трудно сосредоточиться и сидеть на одном месте в течение продолжительного времени. Преподавателю необходимо понимать эти особенности и учитывать их при работе с учениками. Обстановка в классе должна соответствовать образу, которого создал К. Д. Ушинский: «Нужно позволять классу свободно бурлить, волноваться, но удерживать его всякий раз в тех пределах, которые нужны для успеха учения, мёртвая тишина на уроке недопустима». В классе необходимо создавать дружественную и открытую атмосферу, чтобы ученики могли задавать вопросы, общаться между собой и чувствовать себя комфортно, что способствует более эффективному обучению. Школа должна готовить учеников к жизни в информационном мире, поэтому использование информационно-коммуникативных технологий является важным аспектом в работе с учениками начальной школы. Учителя должны понимать, что дети уже владеют определенными навыками работы с гаджетами, и традиционные методы обучения могут быть скучными для них. Следовательно, методы обучения с использованием информационных технологий в игровой форме становятся крайне актуальными.

«Игра – это огромное светлое окно, через которое в духовный мир ребёнка вливается живительный поток представлений, понятий об окружающем мире. Игра – это искра, зажигающая огонёк пытливости и любознательности»
Сухоликовский В.А.

В педагогической практике игровые технологии являются методами и средствами, используемыми для организации занимательного процесса обучения через игры. Они призваны сделать обучение более увлекательным, помогают повысить настроение учащихся и упрощают усвоение учебного материала. Для эффективного использования игровых технологий учителю

необходимо постоянно совершенствовать свои знания по теории и методике преподаваемого предмета, обновлять методическую литературу и применять новейшие технологии в работе. Применение игровых технологий на уроках способствует развитию познавательной активности учащихся и может быть использовано на всех уроках начальной школы. В начальной школе дети сталкиваются с новыми учителями и предметами, поэтому учителю начальных классов необходимо сделать процесс обучения интересным и необычным, чтобы заинтересовать учащихся в обучении.

В ходе разработки своего обучающего приложения было принято решение сосредоточиться на использование игровых технологий для изучения математики. Существует огромное множество вариантов организации игровых уровней для изучения различных дисциплин, выбор математики обусловлен несколькими вещами.

Во-первых, в ходе разработки на основе этой дисциплины можно рассмотреть основные механики и возможности Unreal Engine 5.

Во-вторых, многие дети считают математику скучной и сложной, что может привести к тому, что они теряют интерес к изучению этой дисциплины. Использование игровых технологий может помочь сделать процесс обучения более интересным и увлекательным, что способствует повышению мотивации учащихся.

Современная наука подтверждает эффективность использования игровых методов при обучении математике в начальной школе. Многие исследования, такие как [5, 6], демонстрируют, что игры и задачи, применяемые в обучении математике, могут значительно улучшить успеваемость учащихся и способствовать развитию их критического мышления. Тем не менее, как утверждается в исследовании [7], следует помнить, что не все игры подходят для обучения математике, и необходимо тщательно выбирать игры, соответствующие уровню знаний учащихся, а также не забывать о других

Одним из примеров игр, которые имеют значительную эффективность являются игры с карточками. Например, школьникам могут предложить составить карточки с цифрами и знаками операций в правильном порядке для получения правильного математического выражения. Другой пример - это игры, в которых дети должны соотнести карточки с числами и математическими знаками, чтобы завершить выражение. Карты могут использоваться для обучения различным математическим операциям, таким как сложение, вычитание, умножение и деление.

Дети в возрасте от 6 до 11 лет имеют свои особенности, которые следует учитывать при использовании игровых методов для обучения математике в начальной школе, а также при разработке игрового приложения. Исследования в области нейропсихологии показывают, что дети в этом возрасте мотивированы к игре и могут легко участвовать в игровом процессе, а также могут абстрагироваться от реальной ситуации и представлять себе абстрактные концепции. Поэтому использование игровых элементов в обучении математике для начальных школьников может быть эффективным, так как оно может заинтересовать и мотивировать их к изучению математики, а также помочь развивать их математические навыки. Однако дети в этом возрасте обычно имеют низкий уровень концентрации, поэтому игры должны быть достаточно короткими и захватывающими, чтобы удерживать их внимание.

Несмотря на множество очевидных преимуществ геймификация обучения имеет и ряд некоторых рисков и недостатков, которые необходимо учитывать.

1. Отсутствие достаточного количества высококачественных игр и сервисов, специально предназначенных для использования в начальной школе, может быть недоступно как для учителей, так и для учеников. Именно с этой проблемой также может и помочь мое исследование, а также разработка приложения на базе Unreal Engine.

2. Использование в процессе обучения игровых методик может создавать сложности для оценивания усваивания материала, поскольку игра может не совсем в полной мере отражать понимание ребенка той или иной темы

3. Необходимость обучения учителей использованию игровых методов и технологий в обучении. Несмотря на высокий уровень цифровизации общества на сегодняшний день остается множество людей слабо знакомых с современными цифровыми технологиями, среди них могут быть и учителя. Поэтому данная проблема может стоять довольно остро для некоторых преподавателей.

4. Игровые методы могут не учитывать индивидуальные особенности каждого ученика, что может снижать их эффективность в обучении тех учеников, которые имеют специальные образовательные потребности.

5. Игровые методы могут привести к перегрузке учеников информацией из игр, отвлекая их от основного учебного материала.

Для того чтобы избежать недостатков использования данных методов при обучении математике в начальных классах, нужно более пристально подходить к планированию и организации процесса обучения. Это означает, что необходимо находить баланс между использованием игровых и классических подходов к обучению ребенка, используя игры как дополнительный инструмент обучения, не перегружая им основной процесс. Крайне важно правильно оценить, насколько ученики понимают цели и принципы игры, а также соответствие уровня обучения сложности игры для каждого ученика.

2. Оценка функциональности и эффективности Unreal Engine 5 в контексте образования.

Unreal Engine 5 - это программное обеспечение, разработанное компанией Epic Games для создания видеоигр, анимации и виртуальной реальности с высокой производительностью и передовыми функциями. Оно имеет расширенный редактор материалов, систему анимации персонажей, инструменты для создания мира и уровней, поддержку физической симуляции, инструменты для создания и редактирования света и многое другое. Unreal Engine 5 позволяет создавать очень крупные и детализированные игровые миры с высоким уровнем графической проработки и сложной физической симуляцией, а также создавать виртуальную реальность для образовательных приложений. Однако, его функциональности также могут быть применены в образовательных целях.

Оценка функциональных возможностей Unreal Engine 5 в контексте образования может включать такие факторы, как:

1. Возможность создания интерактивных сред с высокой степенью реалистичности, которые могут быть использованы для обучения на практике. Например, использование виртуальных симуляторов для обучения медицинским процедурам или технике.

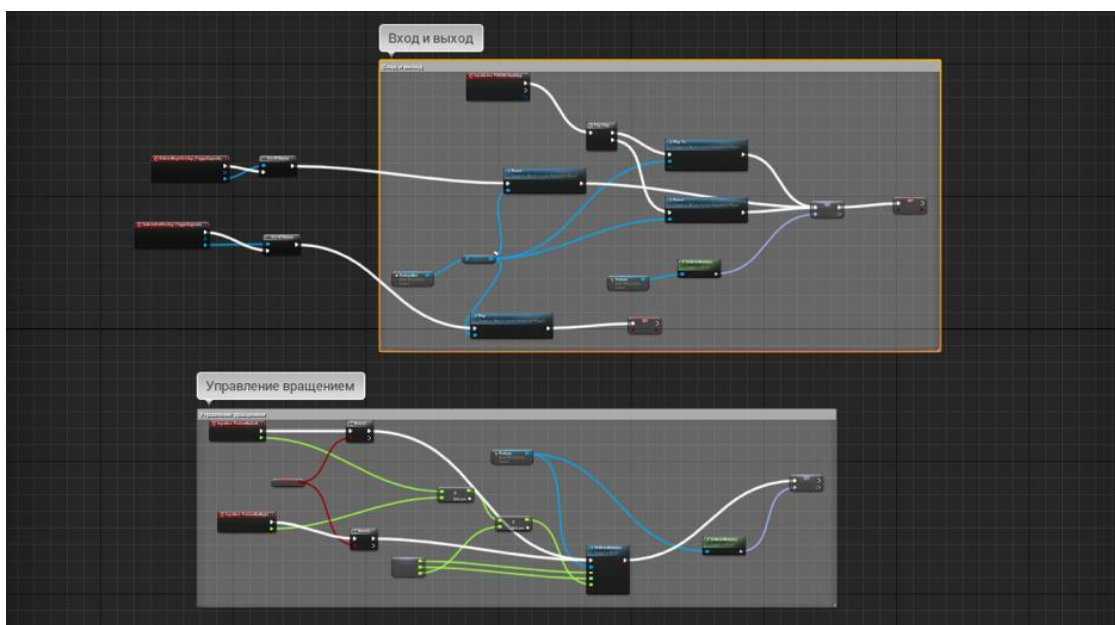
2. Возможность создания визуальных презентаций, которые помогают ученикам лучше понимать и запомнить сложные концепции. Например, создание визуализаций для обучения геометрии или физики.

3. Возможность создания игровых приложений, которые могут быть использованы для обучения и проверки знаний. Например, создание обучающих игр для изучения иностранных языков или математики.

4. Возможность сотрудничества и создания общих проектов, что может помочь ученикам развивать навыки коммуникации и коллективной работы.

5. Удобный и интуитивно понятный интерфейс, который делает создание и использование образовательных материалов доступным для широкого круга пользователей.

Существует множество сфер обучения в которых применяется Unreal Engine. По сути главная задача при использовании данного движка – это имитация тех или иных процессов или явлений. На сегодняшний день рынок предоставляет множество интерактивных приложений, включая использование VR и AR. В работе над интерактивными приложениями визуальное программирование, доступное в Unreal Engine, представляет значительную помощь. Кроме того, наличие обширного учебного контента и множества готовых решений в сообществе существенно упрощает процесс создания таких приложений, что позволяет значительно экономить время и силы.



(рис.1) Пример визуального скрипта Unreal Engine

Примером реализованного проекта является созданная специалистами кафедры ИГД НИУ МИЭТ совместно с детским технопарком "СМАРТ-ПАРК МИЭТ" и Зеленоградским инновационно-технологическим центром (ЗИТЦ) Виртуальная экскурсия по участку производства микроэлектромеханических систем.

Данный продукт создан для ознакомления учащихся старших классов с процессом производства микроэлектромеханических систем. Используя данную виртуальную экскурсию, пользователь имеет возможность ознакомиться с работой установок и принять участие непосредственно в создании изделия. Данной приложение может использоваться как средствами виртуальной реальности, так и средствами обычного монитора мыши и клавиатуры.

В условиях пандемии во многих учебных заведениях встала проблема отсутствия у студентов доступа к лабораторному оборудованию, в этой ситуации Unreal Engine стал отличным инструментом для некоторых преподавателей, которые смогли реализовать работу такого оборудования непосредственно с помощью данного движка. Интерактивные приложения в образовании могут быть использованы не только для создания симуляторов и демонстраций работы реальных процессов и устройств, но и в других областях. Тем не менее, разработка симуляторов и демонстраций является одной из самых сложных задач, требующих большого количества времени и ресурсов.

Кроме создания симуляций производств или копий лабораторного оборудования Unreal Engine открывает доступ к интерактивным приложениям и симуляциям, который могут помочь школьникам начальных классов в изучении математике. Например, в Unreal Engine можно создать интерактивную математическую головоломку или задачу, которую школьник должен решить, используя знания математики. Также можно создать визуализации математических концепций, которые могут быть трудны для понимания в абстрактной форме.

Также можно создать симуляцию графических объектов и использовать их для объяснения геометрических концепций, таких как углы, длины, площади и объемы. Помимо этого, можно создать интерактивную доску, на которой учитель или ученик могут рисовать графики, графы и преобразования.

Кроме того, Unreal Engine может быть использован для создания серьезных образовательных игр, которые помогают школьникам изучать математику. Например, можно создать игру, в которой школьник должен использовать математические знания для решения головоломок и преодоления препятствий. Использование данного движка в образовании математике может сделать изучение этой дисциплины более интерактивным и увлекательным для школьников начальных классов.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Глава 2 Проектирование обучающего игрового приложения «Математика в игре»

1. Назначение разработки:

Дипломная работа, в рамках которой создаётся обучающее игровое приложение на базе Unreal Engine ориентированная на детей начальных классов, являющаяся продуктом сферы компьютерных развлечений и образовательного контента. Задача - создать обучающее игровое приложение, которое поможет детям начальных классов развивать когнитивные навыки и улучшать навыки в области математике. Приложение должно быть привлекательным и увлекательным, чтобы заинтересовать детей и стимулировать их желание учиться.

Дипломная работа является комплексным проектом, который охватывает все стадии разработки ПО.

2.Стадии и этапы разработки:

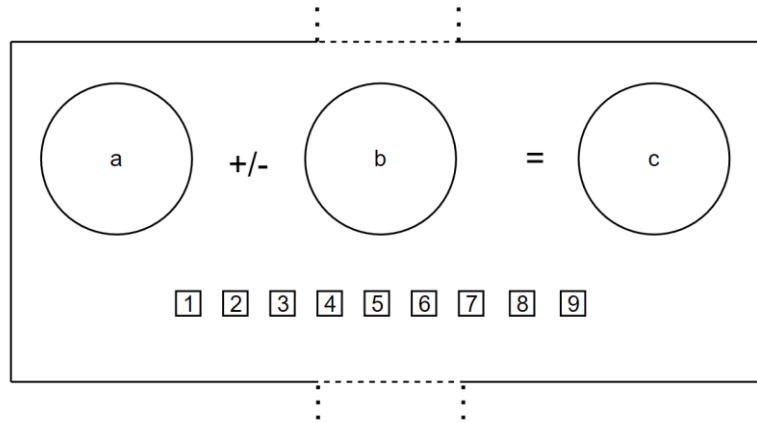
2.1. Создание и проработка идеи приложения

– Продумывание основных игровых механик проекта.

Основная задача – это сделать стандартные методы обучения математике в начальной школе более интересными благодаря переходу этих методов в компьютерную игру.

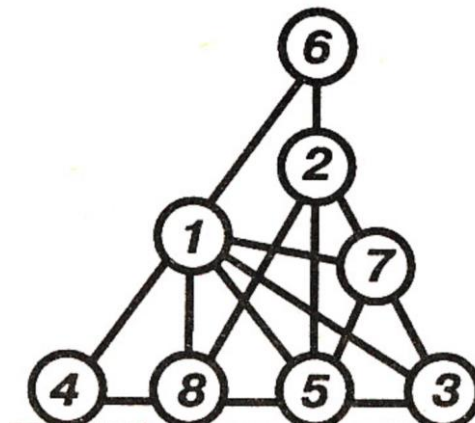
В начальной школе дети изучают основы арифметики, величины, алгебры и геометрии. Обучение математике связано как с теорией, так и с практикой, и представлено в виде блоков, которые связаны с поэтапным изучением нумерации чисел. Было принято решение сосредоточиться именно на арифметических операциях.

Основная идея – это сделать из игрового пространства поле для решения простых примеров, где игроку даны кубики со значением $[1;9]$, и ему необходимо перетаскивать эти кубы в нужную область, чтобы составить верное равенство. Как только игрок составит верное равенство, он получает доступ к следующей игровой области, в которой решает следующее равенство.



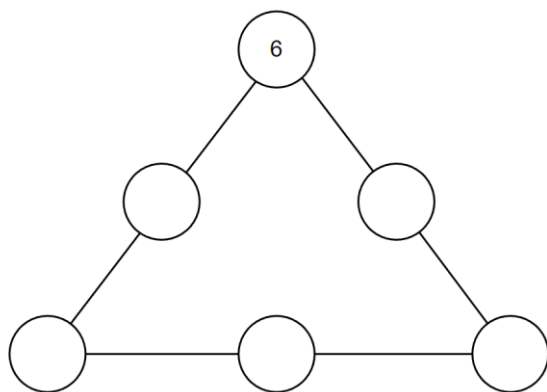
(рис.2)

Помимо этого, можно организовать решение математических головоломок, которая основана на той же механике переноса кубиков со значениями от $[1;8]$. Необходимо расставить в областях кубики, так чтобы две последовательные цифры не оказались в кружочках, соединённым одним отрезком.

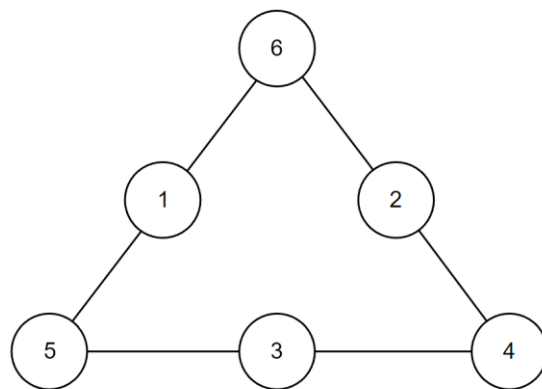


(рис.3)

Также похожим образом можно организовать уровни, в которых необходимо, например, расставить числа 6,5,4,3,2,1 в кружках так, чтобы сумма чисел вдоль каждой прямой равнялась 12.



(рис.4)



(рис.5)

Для того чтобы математические головоломки были эффективными для детей начальных классов, необходимо обеспечить правильный уровень сложности, который не будет вызывать у детей стресс и дискомфорт. Также следует предоставлять достаточное количество подсказок и объяснений, чтобы помочь детям справляться с задачами и научить их решать их самостоятельно.

Хотя существует множество математических головоломок и различных вариаций организации игровых уровней, в данном проекте был сделан акцент на простых арифметических операциях. Главной целью проекта является демонстрация основного функционала игрового движка в контексте образования, а также показать, какие возможности предоставляются благодаря реализованным механикам. В рамках моей дипломной работы я не рассматриваю необходимость реализации всех возможных видов игр, а сосредоточусь на разработке определенного набора заданий, которые помогут детям лучше понять математические концепции и сделать обучение более интересным и увлекательным.

Главная цель разработки приложения «математика в игре»– это отразить основные обучающие возможности игрового движка Unreal Engine 5. Задача состоит в том, что необходимо реализовать отдельные элементы, которые будут достаточно универсальны под различные проекты, в зависимости от целей разработки, конечно, охватить все возможные механики, чтобы они подходили под абсолютно любую идею невозможно, но реализованные механики в ходе разработки могут быть применены для широкого спектра различных вариаций организации игровых уровней.

2.2. Выбор движка

Приложение будет создано на базе игрового движка Unreal Engine 5.1

При выборе игрового движка для создания приложения могут учитываться различные факторы. В случае Unreal Engine можно отметить несколько преимуществ, которые обусловили выбор. Во-первых, данный движок имеет высокий уровень функциональности, что позволяет создавать производительные и качественные игры и приложения. Такой выбор может быть важен для обеспечения плавной работы и удобства использования приложения. Во-вторых, наличие широкого сообщества разработчиков и пользователей может облегчить процесс разработки и поддержки приложения, а также предоставить доступ к множеству бесплатных и платных ресурсов. В-третьих, Unreal Engine относительно прост в использовании и освоении. Движок предоставляет простой интерфейс и интуитивно понятные инструменты, что делает его доступным для разработчиков с разным уровнем опыта.

Также, существует множество ресурсов, посвященных разработке на Unreal Engine, включая онлайн-курсы, форумы и документацию. Это позволяет разработчикам быстро овладеть навыками работы с движком и быстро находить ответы на возникающие вопросы. Кроме того, Unreal Engine поддерживает множество платформ, что обеспечивает возможность запуска приложения на различных устройствах. В целом, выбор данного движка может быть обусловлен

его функциональными возможностями, наличием сообщества разработчиков и поддержкой множества платформ, что может обеспечить успешное развитие и распространение приложения.

2.3. Реализация графического компонента игры

– разработка окружения героя (локации, объекты для взаимодействия);

Разработка визуальной составляющей проекта: модели персонажей, модели окружающих объектов (props), модели зданий и элементов окружения(environment). В рамках реализации обучающего игрового приложения исходя из предположительных объемов работ по реализации данного проекта было принято решение воспользоваться готовыми решениями, которые сообщество Unreal Engine предоставляет бесплатно в широком спектре различных дизайнов и концепций. Дело в том, что проект, разрабатываемый одним человеком, неизбежно сталкивается с проблемой нехватки рабочего ресурса. Если мы говорим о разработке игры в рамках крупной игровой студии, которая самостоятельно разрабатывает весь контент для своей игры, при этом не используя готовых решений, такая кампания состоит из многих сотен людей, а также может прибегать к помощи аутсорс (Outsource) компаний, который могут брать на себя часть необходимых задач при разработке проекта.

В свою очередь существует множество инди-студии, которые не имеют коммерческой поддержки со стороны каких-либо издателей их коллектив состоит из нескольких или вообще из одного человека, в таком случае отсутствуют ресурсы для разработки полностью своего авторского контента.

Если рассматривать конкретно процесс реализации визуального контента игрового приложения, то он состоит из нескольких стадий: 1.Первый драфт (draft от англ. – черновик) дизайна. Сюда входят самые первые зарисовки общей визуальной концепции проекта, не сосредотачиваясь на конкретных элементах. Данный процесс может занимать довольно большое время пока не будет получен желаемый результат. Также данный этап требует от

разработчика умение реализовывать свои идеи на бумаге, то есть непосредственно умение рисовать.

2.Драфт отдельных элементов (персонажи, окружение и т.д). На этом этапе мы углубляемся непосредственно в разработку сырого дизайна, например, главного персонажа нашей игры. Необходимо проработать, что конкретно это за существо, его размеры, пропорции, одежда, конечно, исходя из целей проекта.

3.Детальная проработка исходя из концептов второго этапа. На этом этапе происходит доработка концептов до мельчайших деталей.

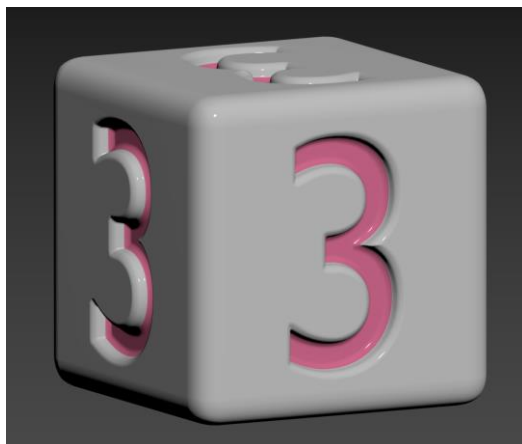
4.Следующий этап – это создание 3д моделей необходимых объектов. На данном этапе также требуются особые навыки, который у разработчика инди-проекта может не быть. Художник исходя их концепта разработанному ранее пытается максимально повторить концепт только уже в 3д, используя для этого такие программы как, например: 3dsMax, Blender, Maya, Marmoset, Substance Painter. Далее мы более подробнее остановимся на этом этапе, так как, не смотря на множество готовых решений в ходе реализации проекта возникла необходимость в дополнительных 3д моделях, которые я успешно создал.

Процесс создания 3d – моделей:

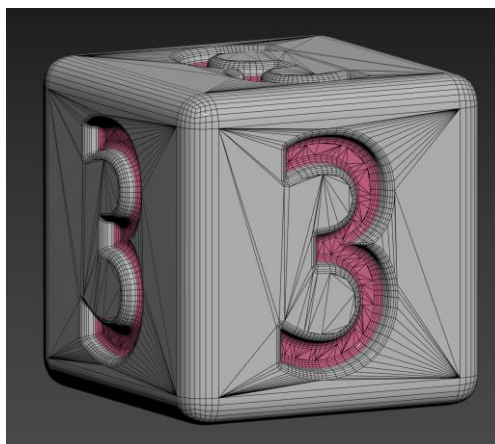
В ходе реализации проекта я использовал следующее программное обеспечение для создания необходимых моделей: Blender (является бесплатным), Marmoset Toolbag 3.08 (является платным), Substance Painter 3D, (является платным). Использование Двух последних перечисленных программ не является обязательным, так как Blender, содержит в себе необходимый функционал. Использование мною этих программ обусловлено приобретенными версиями программного обеспечения, а также хорошим уровнем знаний для использования их.

В ходе создания программного продукта возникла необходимость создания кубов с цифрами [1...9], необходимых для реализации задуманных механик.

Стадия 1 (Создание высоко полигональных версий кубов). На данном этапе мы создаем непосредственно модель такой, какой мы хотели бы ее видеть, не ограничивая себя в количестве полигонов.

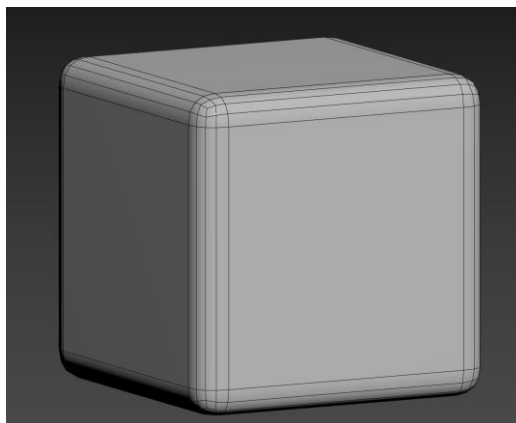


(рис.6)



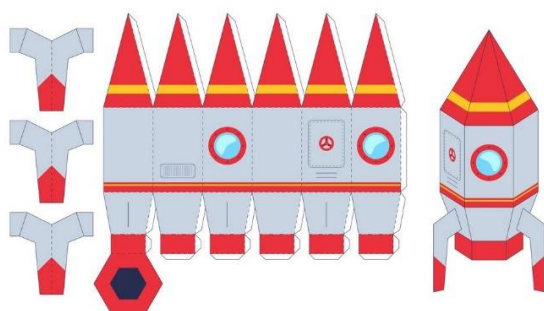
(рис.7)

Стадия 2 (Создание низко полигональной версии модели): На данном этапе нам необходимо оптимизировать нашу геометрию, так как на высоко полигональном объекте слишком много полигонов, которые будут потреблять ресурсы компьютера в процессе игры. Даже несмотря на то, что Unreal Engine 5, обладает технологиями способными справляться с огромными количествами полигонов, и в целом в нашем случае их можно было бы спокойно использовать в игре, высоко полигональной модели присуща высокая сложность при создании ее UV развертки. Скорее именно это главная причина оптимизации нашей геометрии.

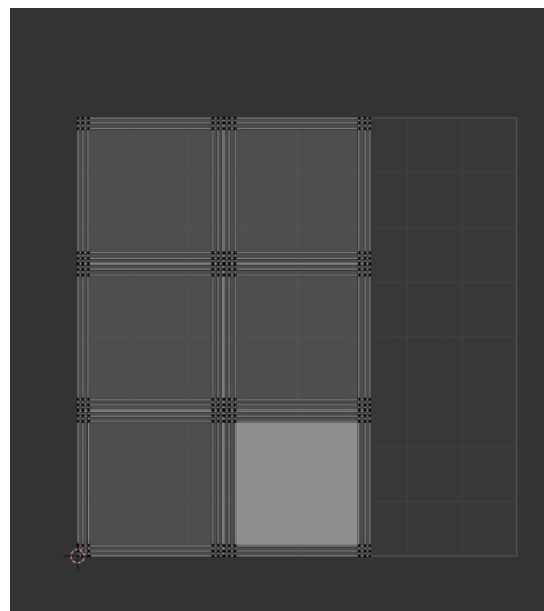


(рис.8)

Стадия 3 (UV – развертка): На данном этапе нам предстоит сделать по сути проекцию нашего трехмерного куба на двумерную плоскость, чтобы лучше понять, что из себя представляет UV развертка можно привести в пример развертку из бумаги. На правом рисунке изображена готовая развертка кубов.

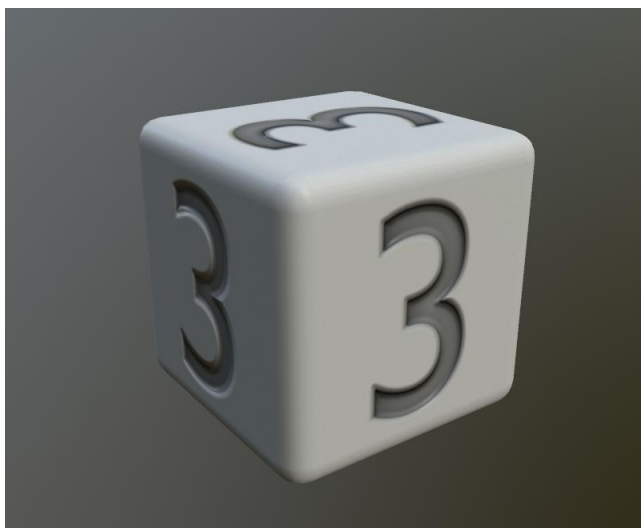


(рис.9)



(рис.10)

Стадия 4 (Запечка (от англ. Bake)): На этом этапе нам необходимо сделать так, чтобы наша низко полигональная модель, стала выглядеть как высоко полигональная, как мы можем заметить, на рисунке низко полигональной модели, отсутствуют цифры. Для того чтобы, на кубе появились цифры нам необходимо запечь текстуры Normal Map (карта нормалей) и Ambient occlusion map (карта затенения). Normal Map – отвечает за правильную отработку освещения исходя из цветов на самой текстуре, именно эта карта позволяет иметь на нашей модели цифры и при этом они будут выглядеть объемно. Ambient occlusion map необходима для судя из самого названия, для отображения затенений, которые помогают модели выглядеть более реалистично.



(рис.11)

Стадия 5 (создания текстур). На данном этапе нам необходимо создать текстуры для нашей модели исходя из требований проекта. Основной материал – это слегка шероховатый фиолетовый металл, также на этом этапе мной была создана карта Emissive. Это карта, отвечающая за свечение объектов. Я создал ее такой, чтобы непосредственно свечение распространялось от самих цифр.

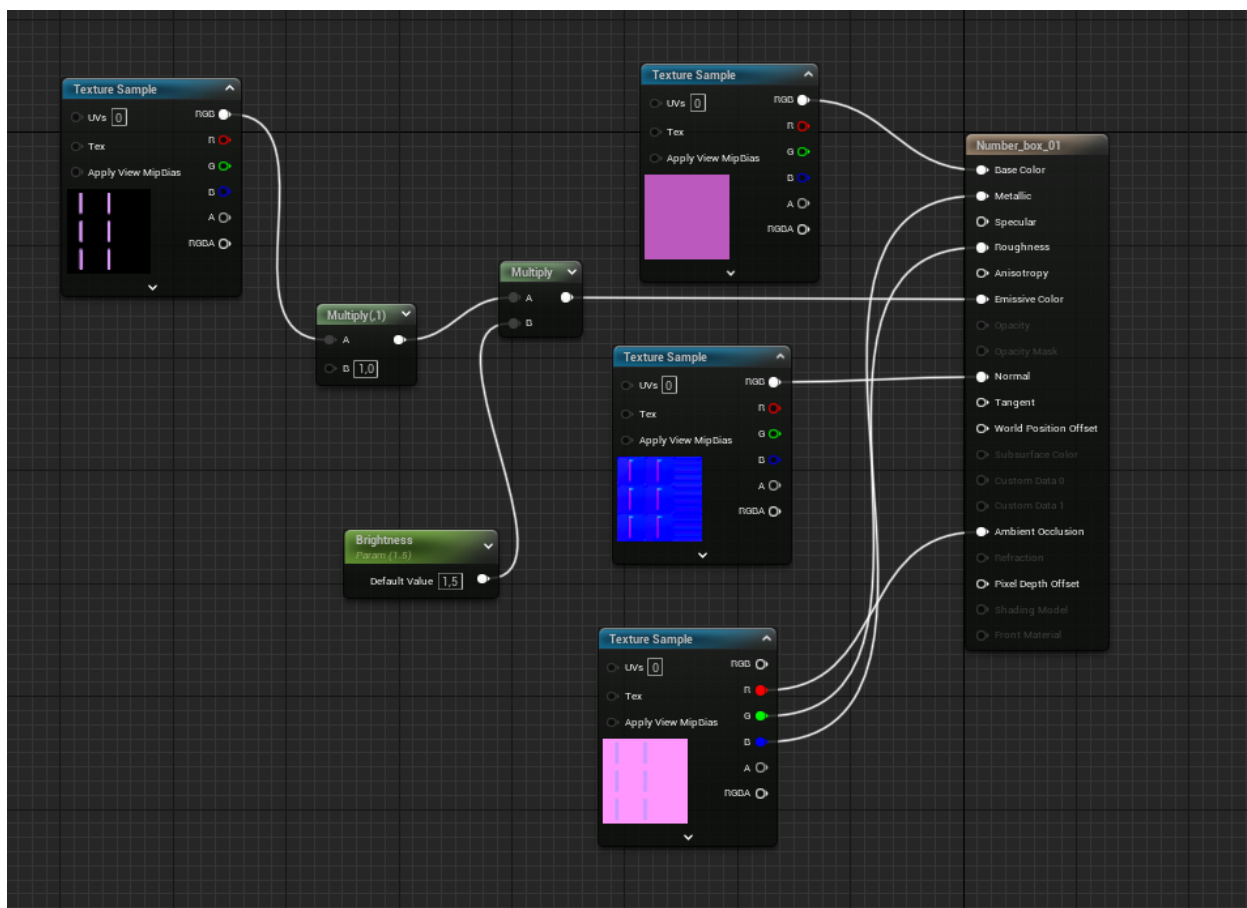


(рис.12)

После того как мы создали наши модели нам необходимо добавить их в свой проект. Как результат мы имеем 1 низко полигональную модель, которая является основной для всех 9 цифр. Также мы имеем для каждого из 9 вариантов по 4 текстуры: Base color (текстура, которая отвечает непосредственно за цвет

объекта), Emissive (свечение), Normal Map (карта нормалей), OcclusionRoughnessMetallic (Эта карта содержит в себе сразу 3 текстуры, которые разложены по разным каналам изображения. Красный канал – ambient occlusion, карта которую мы создали в результате запечки, отвечающая за затенения. В зеленом канале Roughness - это текстурная карта, которая определяет уровень шероховатости или гладкости поверхности объекта в компьютерной графике. В синем канале располагается Metallic map - это текстурная карта, которая определяет, какие части поверхности объекта являются металлическими, а какие - нет.

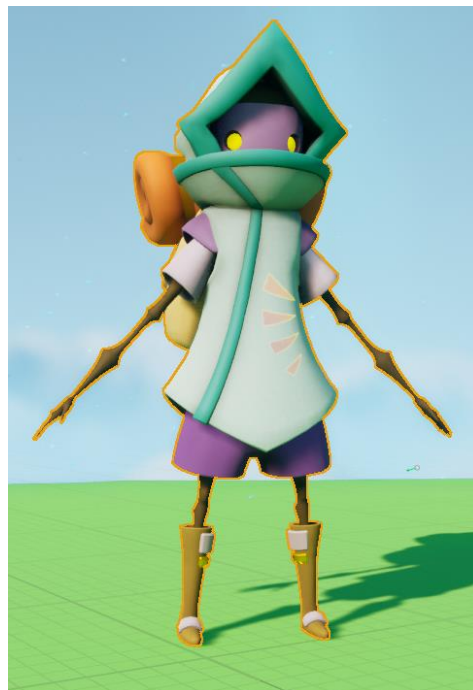
Для того чтобы все текстуры были видны на нашей модели необходимо их правильно подключить к материалу, который привязывается непосредственно к низкополигональной модели. Необходимо корректно подключить карты в нужные каналы и слоты, а также подключая карту Emissive, нужно создать возможность корректировать параметры карты для того чтобы настроить уровень свечения до приемлемых параметров. Для этого мы создаем множитель значения, которое изначально имеет значение в 1, при регулировке которого мы сможем изменять интенсивность карты Emissive.





(рис.14)

Также этот набор содержит персонажа с готовой анимацией:



(рис.16)

Помимо этого, набор содержит ряд различных предметов, которые могут пригодиться разработчику:



(рис.17)

2.4. Разработка программного кода игры

- изучение необходимой для разработки документации;
- разработка непосредственно программного кода игры; Создание базовых механик, отвечающим целям игрового обучающего приложения

Реализация механики подбора предметов:

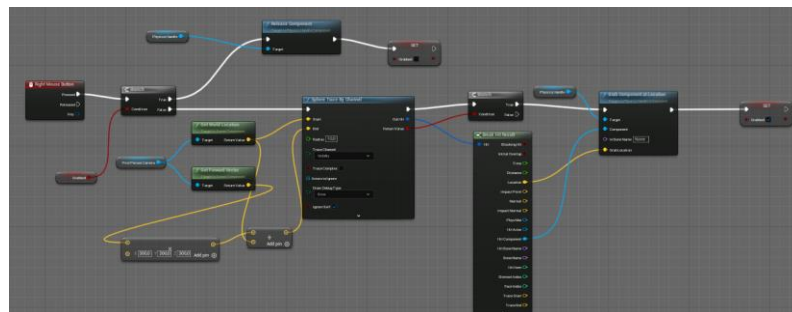
Реализация подбора предметов была реализована по следующему принципу.

Для того чтобы поднять предмет нам необходимо отследить насколько близко предмет находится к игроку. Для этого программный код был реализован на основе выпускания луча из камеры от первого лица главного игрового персонажа. Это позволяет подбору активироваться только в том случае, когда игрок направлен лицом к объекту взаимодействия. Луч выпускается с определенной толщиной, подобранной в ходе реализации опытным путем. Это позволяет отслеживать попадание по объекту даже если игрок слегка промахнулся лучом мимо объекта. Затем Unreal отслеживает попадание с помощью ноды Break Hit Result, передавая значение True или False, а далее при

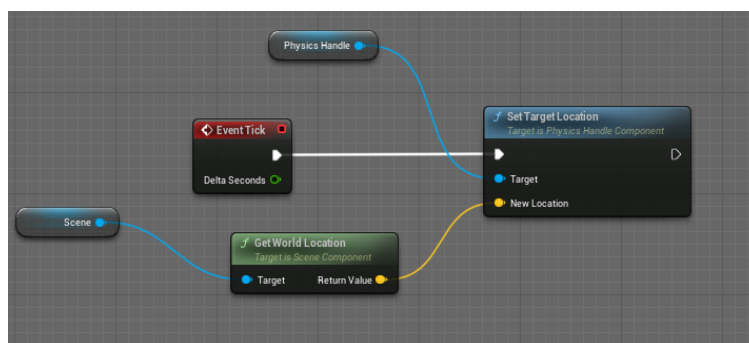
значении True вызывается нода Grab Component Location, которая ответственна непосредственно за возможность привязки предмета к игровому персонажу к объекту Physics Handle, которую необходимо разместить ранее непосредственно к центру масс нашего персонажа, и которая отвечает за отработку движения объекта взаимодействия с использованием физики взаимодействия.

Помимо этого, для корректной работы объекта Physics Handle необходимо отслеживать его положение в реальном времени с помощью ноды Set Target Location, подключенной к ноде Event Tick (отправка данных каждый Tick) Таким образом, при поднятии предмета игроком, луч выпускается из камеры, попадание объекта отслеживается с помощью Break Hit Result, и при успешном попадании предмет привязывается к игровому персонажу через Physics Handle. Затем Set Target Location обновляет положение Physics Handle в реальном времени, чтобы обеспечить корректное перемещение предмета с использованием физики взаимодействия.

Подбор предметов реализован нажатием на правую кнопку мыши



(рис. 18)



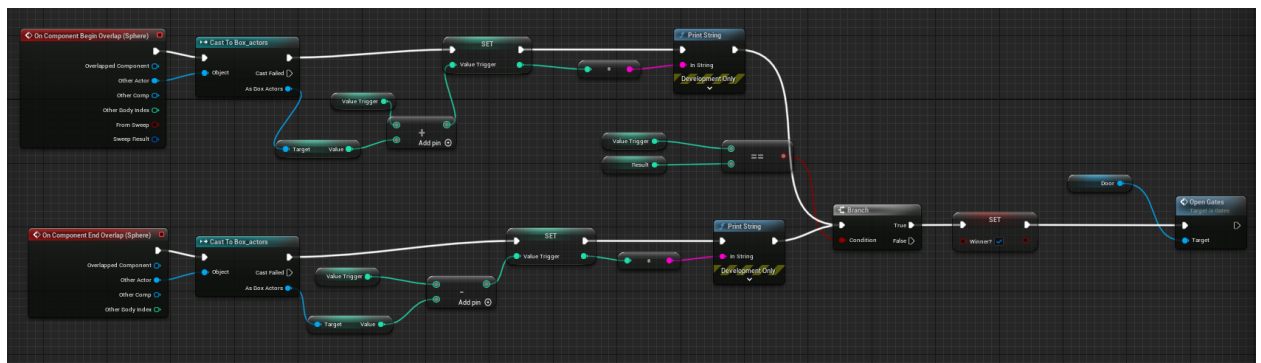
(рис. 19)

Реализация области взаимодействия с объектами

Исходя из целей проекта встала, задача создать область, при внесении в которые кубы с различными значениями происходило бы отслеживание наличия в области объекта, а также суммирование значений нескольких игровых объектов, и передача True или False при верном или неверном вычислении соответственно, для передачи их на ворота, отвечающие за ограничение прохода на следующий игровой уровень

Область вычислений реализована следующим образом:

Сама область состоит из платформы, которая непосредственно видна игроку, а также из сферы, окружающей ее Sphere Collision, именно пересечение этой области будет означать попадание или отсутствие игрового объекта на платформу. За это отвечают ноды On Component Begin Overlap и On Component End Overlap. При пересечении объектом области, мы обращаемся кубам и забираем у них цифровое значение [1...9] и суммируем, а также при покидании этой самой области мы вычитаем покинувшее значение, тем самым мы отслеживаем сумму всех кубов в области в реальном времени. Для того, чтобы открывать доступ игроку к следующим уровням, открывая игровые ворота, нам необходимо сделать так, чтобы каждая отдельная область отвечала за отдельные ворота на уровне. Для этого потребовалось создать массив. В ячейки которых обращаются каждые отдельные ворота, позволяя указать какая игровая область отвечает за конкретные ворота.



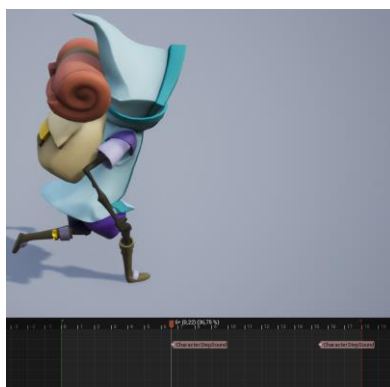
(рис. 20)

Логика открытия ворот основана, но получении вызова из зоны взаимодействия, при получении которого у ворот становится невидимым полупрозрачный «энергетический блок», а также у него отключается collision, что означает, что персонаж может свободно проходить сквозь ворота без препятствий. Отключение коллизии позволяет игровому персонажу физически проникнуть сквозь ворота и продолжить свой путь на следующий игровой уровень.

Реализация звукового сопровождения (звук ходьбы персонажа, звук падения предметов, музыка)

Звук ходьбы персонажа реализован следующим образом: в поверхность под персонажем выпускается луч, который отслеживает попадание по поверхности. А также при попадании получает тип объекта под ногами, которые записаны в массиве «Surface». Данный массив позволит нам при добавлении объекта на сцену выбирать тип поверхности (земля, камень, дерево). Вся логика звука записывается в функцию, которая вызывается на каждый шаг персонажа.

Таким образом, при каждом шаге персонажа функция звука ходьбы будет вызываться, проверять тип поверхности под ним и воспроизводить соответствующий звук для этой поверхности. Это создает реалистичный звуковой эффект ходьбы персонажа в зависимости от того, по какой поверхности он передвигается.

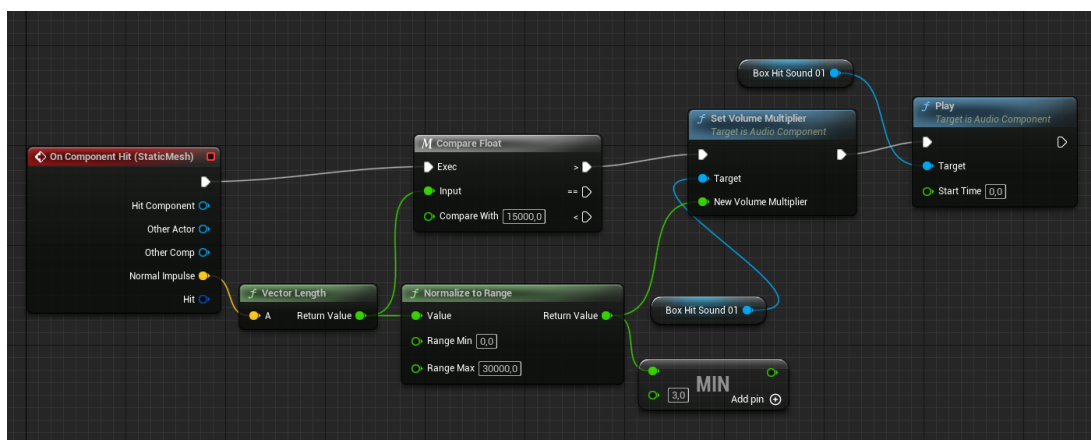


(рис.21)

Расставить функцию необходимо расставить для нескольких анимации: для анимации начала бега, конца бега, начала и окончания ходьбы, начала и окончания прыжка. Для прыжка необходимо создать отдельные функции с аналогичной логикой. Для того чтобы звук менялся в зависимости от типа поверхности, а также не был излишни повторяющимся были собраны наборы различных звуков (4 звука для земли, 4 звука для камня, 4 звука для дерева). При этом каждый набор подключается к своему типу поверхности, а также при каждом шаге проигрывается 1 из 4 случайных звуков, тем самым делая звук не таким скучным и заикленным.

Звук падения предметов реализован несколько другим образом. Нода On Component Hit (используя выход Normal Impulse, отвечающий за регистрацию столкновения с твердыми объектами, а также получает значение импульса). Исходя из значений импульса, можно установить различные уровни громкости для звуков падения предметов. Например, при небольшом импульсе звук может быть более тихим, а при большом импульсе звук будет громче и более резким.

Таким образом, при каждом столкновении предмета с другим объектом, нода On Component Hit получает значение импульса и на основе этого значения определяет громкость звука падения предмета. Это позволяет создать более реалистичный звуковой эффект в игре, отражающий интенсивность падения объектов.



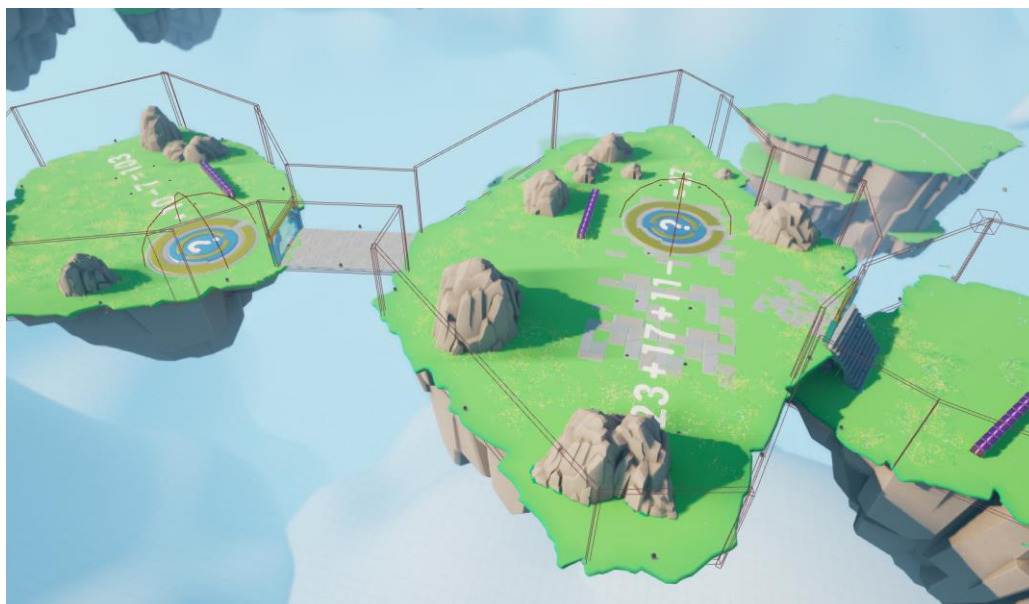
(рис.22)

Сборка игрового уровня

На данном этапе мы имеем лишь набор механик, которые можно использовать при разработке проекта. Необходимо создать само игровое приложение. По главной задумке, нам необходимо разместить отдельные игровые площадки, связанные друг с другом воротами, которые открываются при верной сумме кубов внутри игровых платформ. Также нужно придумать арифметические выражение, которые бы решал игрок от уровня к уровню. Расположены они с возрастанием сложности. Сначала с применением сложения и вычитания, а затем с делением и умножением.

Первым делом мы размещаем сами игровые площадки, по которым будет перемещаться игрок, они созданы из летающих островов, которые в себе уже содержат Unreal Learning Kit. Далее мы размещаем ворота, от одного острова к другому, при этом используя лестницы и каменные платформы. Затем мы размещаем платформы, в которые игрок будет переносить объекты и в которых будет происходить вычисление суммы кубов и подключаем каждую из платформ к нужным воротам, так чтобы при верной сумме открывался доступ именно в том месте, в котором это необходимо. Далее рядом с каждой из игровых платформ мы размещаем кубы от 1 до 9 включительно.

Из-за того, что игровые поля, по которым перемещается игрок по сути это летающие в воздухе острова, встает проблема того, что игрок может свалиться с этого острова вниз или же случайным образом сбросить вниз один из кубов, без которого верное значение арифметического выражения не получится. Данная проблема решается построением ограды для игрока. Нам нужно создать невидимый объект, обладающий коллизией, который бы не пропускал ничего сквозь себя. Затем необходимо расставить этот объект по всему периметру игровой области куда бы мог попасть игрок, дабы избежать его падения и невозможность продолжать игру.



(рис.23)

Также необходимо разнообразить наш игровой мир. Для этого была добавлена трава, которая наносится на поверхность игрового поля как бы с помощью кисти, в которую можно загрузить множество вариантов растительности, которой обладает Unreal Learning Kit. Затем на сцене можно разместить различные визуальные эффекты, например, эффекты летающей листвы или потоков ветра – это неплохо разбавляет общий визуальный стиль приложения.

2.5 Реализация музыкальной/звуковой составляющей проекта

При разработке продумывание музыкальной составляющей и звукового сопровождения является важным аспектом. Эти элементы могут значительно повысить привлекательность приложения и помочь детям лучше воспринимать материал. Необходимо учитывать возрастную категорию детей и соответствовать им по стилю и настроению музыки. Звуки в игре могут быть использованы для акцентирования внимания детей на важных моментах, а также для улучшения понимания игровых процессов. Однако музыка и звуки не должны отвлекать детей от обучения и не должны быть слишком громкими или назойливыми.

При разработке приложения, в ходе реализации звуковой составляющей был реализован ряд задач:

1. Звук ходьбы персонажа, причем звук ходьбы был реализован таким образом, что при смене типа поверхности под ногами персонажа меняется и звук на тот, которому характерен данный тип поверхности
2. Был добавлен звук падения предметов, как и в случае с ходьбой звук должен отличаться в зависимости от того какой поверхности касается предмет, а также помимо этого необходимо, чтобы звук корректно отрабатывал скорость с которой падает предмет тем самым регулируя его громкость
3. Реализован звук открытия дверей, а также звук сигнализирующий о том, что уровень пройден. Важным условием было, чтобы звук открытия дверей затихал при удалении.
4. Реализована фоновая музыка, а также фоновые звуки окружения.

2.6. Тестирование

– юзабилити-тестирование:

Тестирование главного меню:

В ходе разработки приложения «математика в игре» было проведено юзабилити тестирование:

- Проверил навигацию и понятность пользовательского интерфейса главного меню.
- Оценил удобство настройки громкости звука и других параметров. Убедился, что изменения в настройках применяются корректно.

- Проверил работу кнопок и элементов управления в главном меню. Убедился, что они реагируют на нажатия пользователя и выполняют соответствующие действия.

- Оценил визуальное оформление и эстетическое восприятие главного меню. Убедился, что оно соответствует ожиданиям пользователей и общей тематике приложения. В случае дальнейшего развития проекта, можно улучшить общий визуальный стиль главного меню.

Тестирование базовых внутриигровых механик:

- Проверил функциональность перетаскивания кубов со значениями цифр от 1 до 9. Убедился, что кубы перемещаются плавно и точно в соответствии с действиями пользователя.

- Проверил взаимодействие с областью, в которой кубы складываются и решается арифметическое выражение. Убедился, что область правильно распознает и обрабатывает перемещение кубов и решение арифметических примеров.

- Проверил корректность вычислений и верность решения арифметических примеров. Убедился, что приложение дает правильные ответы и реагирует на неправильные решения.

- Проверил механизм открытия ворот и перехода к следующему игровому уровню при правильном решении. Убедился, что игрок может продолжать прохождение игры без проблем.

-

Оценка производительности приложения:

Оценил использование ресурсов, таких как процессор, память и графическая производительность. Убедился, что приложение эффективно использует ресурсы и не вызывает неприемлемой нагрузки на устройства.

3. Требования к программе или программному изделию

3.1. Разработка главного меню.

Разработка главного меню начинается с поиска основных кнопок, подходящих визуальному стилю игры. В открытом доступе существует множество шаблонов в различных стилях и сразу без фона, что позволяет быстро разделить изображение на отдельные кнопки и использовать их в своем проекте. Далее мы размещаем их в Widget, который будет отрисовываться, при вызове главного меню. В окнах мы размещаем надписи: «Играть», «Обучающее приложение», «Настройки» и «Выход». Внутри кнопок, которые мы добавили ранее. Также необходимо добавить вариативности нашим кнопкам. Buttons в Unreal Engine, подразумевают возможность их изменения при наведении на них мыши и нажатии. Были добавлены варианты тех же самых кнопок, что были созданы ранее, только с более ярким текстом, для того чтобы игроку было понятно, когда он наводит на кнопку, что она активна.

Реализация функционала меню начинается с добавления ноды OpenLevel (By Object Reference), которая ссылается на кнопку «Играть», тем самым давая возможность запускать наш уровень при нажатии кнопки.

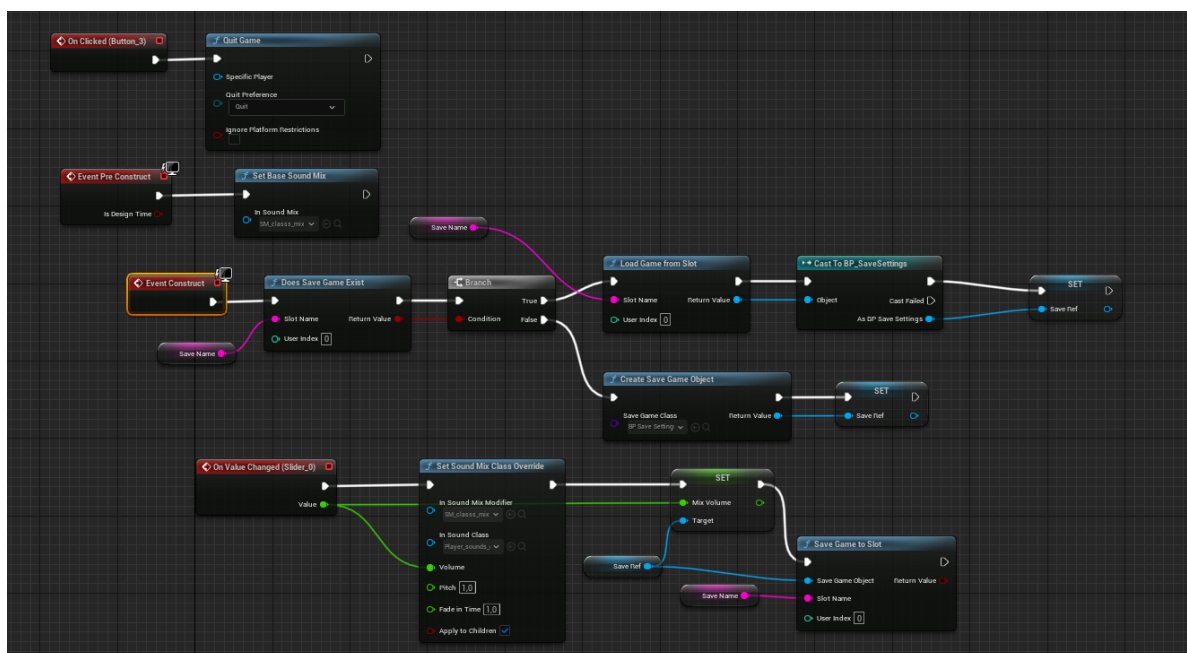
Далее необходимо добавить функционал кнопке «Выход». Для нее мы подключаем ноду Quit Game. Данная нода ответственна за выключения уровня, что и позволяет выйти из игры.

– звуковой функционал главного меню:

- регулировка громкости;

Для реализации настройки уровня громкости в игре необходимо в Widgets ползунок, которым будет возможность регулировать общую громкость игры. За изменение ползунка ответственна нода On Value Changed. Далее она подключается к ноде Set Sound Mix Class Override, она нужна для того чтобы конкретно обозначить какие классы звуков будут изменяться. Все звуки, которые есть в игре записываются в классы, это необходимо для того чтобы была возможность разделить звуки и сделать возможной настройку отдельных звуковых элементов. Но сама настройка звука это еще не все. На данном моменте регулировка уже будет работать, но как только меню будет закрыто, звук вернется к изначальному значению громкости. Поэтому нам необходимо создать возможность сохранения значения громкости. Для этого обозначаем переменную «Save Name» которую подключаем к ноде Does Save Game Exist, которая записывает string значение в переменную.

Финальный функционал главного меню выглядит следующим образом.



(рис. 24)

3.2. Требования к информационной и программной совместимости

– игра должна работать под ОС семейства Windows

3.3. Минимальные системные требования

- ОС: Windows
- оперативная память: 4 ГБ;
- клавиатура, мышь.

4. Оценка вариативности использования разработанных систем.

Можно оценить разнообразие использования разработанных систем в приложении «Математика в игре» следующим образом:

4.1. Оценка вариативности при использовании арифметических операций:

Различные операции: Включение различных арифметических операций, таких как сложение, вычитание, умножение и деление, позволит игроку сталкиваться с разными типами примеров. Например, на одном уровне игроку может быть предложено решить примеры, содержащие только сложение и вычитание, а на другом уровне - примеры с умножением и делением.

Разное число кубов: Вариация в числе кубов, которые игрок должен перетаскивать и использовать для решения примеров, может добавить разнообразие в игровой процесс. Например, на одном уровне может быть всего несколько кубов, а на другом - значительно больше, что потребует от игрока более тщательного планирования и оценки.

Варианты расположения кубов: Различные варианты расположения кубов на игровом поле могут создать разнообразие в решении арифметических примеров. Например, на одном уровне кубы могут быть расположены в линию, а на другом - в случайном порядке или в виде групп.

Увеличение сложности: По мере продвижения игрока через уровни, можно постепенно увеличивать сложность арифметических примеров. Это может включать увеличение числового диапазона, добавление более сложных операций или увеличение количества примеров, которые игрок должен решить за ограниченное время.

4.2. Оценка вариативности с точки зрения дизайна уровней:

Изменение размеров и формы игрового поля:

Предлагать игроку уровни с разными размерами и формами игрового поля может добавить разнообразие в игровой процесс. Например, уровень может иметь квадратное, прямоугольное или даже необычную форму поля, что потребует от игрока адаптации своих действий и стратегии.

Добавление препятствий:

Размещение препятствий на игровом поле, таких как блоки или преграды, может создать дополнительные вызовы для игрока. Например, игрок может столкнуться с препятствием, которое блокирует доступ к определенным кубам или ограничивает перемещение по полю, требуя более тщательного планирования и выполнения задач.

Добавление дополнительных элементов:

Включение дополнительных элементов, таких как кнопки, выключатели или специальные блоки, может добавить разнообразие в игровой процесс. Например, на уровне могут быть размещены кнопки, которые активируют или

отключают определенные части поля, требуя от игрока стратегического использования этих элементов для решения арифметических примеров.

Вариация визуального оформления:

Вариация визуального оформления уровней может также создать разнообразие в игровом опыте. Например, каждый уровень может иметь уникальные цветовые схемы, темы или фоны, что поможет отличить один уровень от другого и добавит визуального интереса для игрока.

Разнообразие уровней и дизайна уровней:

Это поможет убедиться, что игроку предлагается достаточное разнообразие сред, где он будет решать арифметические примеры. Это создаст ощущение прогрессии и увлекательности, поскольку каждый уровень представляет новые вызовы и возможности для игрока.

4.3. Оценка вариативности с точки зрения различных игровых механик:

Дополнительные элементы управления:

Добавление дополнительных элементов управления может обогатить геймплей и предоставить игроку больше возможностей. Например, вместо простого перетаскивания кубов можно добавить возможность вращения или масштабирования кубов, что расширит вариативность действий игрока.

Использование специальных блоков:

Включение специальных блоков с уникальными свойствами и функциями может добавить разнообразие в игровой процесс. Например, можно добавить блоки, которые изменяют значения кубов, блоки-препятствия, которые нужно обходить, или блоки-бонусы, которые предоставляют дополнительные возможности игроку.

Влияние игрока на игровое поле:

Предоставление игроку возможности влиять на игровое поле и его состояние может добавить интерактивности и вариативности в игру. Например, игрок может иметь способность менять местоположение определенных блоков или создавать временные платформы для достижения целей.

Постепенное введение новых механик:

По мере продвижения игрока через уровни, можно постепенно вводить новые механики и элементы геймплея. Это поможет поддерживать интерес игрока и создавать ощущение прогрессии. Например, на начальных уровнях игрок может только перетаскивать кубы, а на более продвинутых уровнях могут появляться новые элементы и действия.

4.4. Оценка вариативности с точки зрения игрового опыта:

Скорость игры:

Дать возможность пользователю настроить скорость игры может быть полезно для тех, кто предпочитает более быстрый или медленный темп игры. Некоторые игроки могут предпочитать более вызывающий и динамичный геймплей, в то время как другие могут предпочитать более спокойный и расслабляющий опыт.

Временные ограничения:

Дать возможность пользователю настроить длительность временных ограничений на уровне может быть полезным для тех, кто предпочитает более напряженный и быстрый геймплей или, наоборот, более расслабленный и неторопливый подход к решению арифметических примеров.

Общая оценка вариативности в приложении «Математика в игре» будет определяться синергией всех перечисленных аспектов: вариативностью в арифметических операциях, дизайном уровней и использованием различных игровых механик. Комбинированное воздействие этих факторов создает максимально разнообразный игровой опыт, который способствует поддержанию интереса игрока на протяжении всей игры.

ЗАКЛЮЧЕНИЕ

Цель данной выпускной квалификационной работы заключалась в Разработке обучающего игрового приложения на базе Unreal Engine и ее последующая оценка образовательного потенциала.

Для решения этого вопроса автором данной работы было сделано следующее:

1.Проведен анализ образовательных задач и целей: Автор определил образовательные задачи и цели, которые должно решать разработанное приложение. Это позволило сфокусироваться на разработке функционала и контента, поддерживающих эти задачи и цели.

2.Проведен анализ предметной области, связанной с обучением и использованием игровых технологий в образовании. Автором были рассмотрены следующие аспекты:

- Теоретический обзор: Была изучена существующая литература и исследования, касающиеся применения игровых технологий в образовании. Были проанализированы основные концепции, методы и тенденции в данной области, а также их потенциал для улучшения образовательного процесса.

- Преимущества игровых технологий в обучении: Были исследованы преимущества использования игровых технологий, таких как Unreal Engine, в образовании. Были выявлены такие факторы, как повышение мотивации и вовлеченности учащихся, развитие критического мышления и проблемного мышления, улучшение визуального и сенсорного восприятия, а также возможности для создания интерактивного и индивидуализированного обучения.

- Применение игровых технологий в различных образовательных сферах: Было проведено исследование примеров применения игровых технологий в

различных образовательных сферах, таких как школьное образование, высшее образование и профессиональное обучение. Были рассмотрены конкретные примеры проектов, которые успешно использовали игровые технологии для достижения образовательных целей.

- Выбор Unreal Engine: Автор провел анализ преимуществ и возможностей Unreal Engine в контексте образования. Были выделены такие факторы, как мощный визуальный редактор, поддержка VR и AR, широкий набор инструментов и ресурсов для разработки образовательных приложений.

4. В ходе выпускной квалификационной работы было спроектировано и разработано обучающее игровое приложение на базе Unreal Engine 5.1

3. Рассмотрены основные технические возможности Unreal Engine, а также проведена оценка обучающего потенциала данного игрового движка на примере разработке обучающего игрового приложения.

Таким образом, поставленные задачи полностью решены, игровое приложение разработано, произведена оценка обучающего потенциала, следовательно, цель выпускной квалификационной работы достигнута.

СПИСОК ЛИТЕРАТУРЫ

1. Аблаева Лола Фрунзевна ИГРОВЫЕ ТЕХНОЛОГИИ НА УРОКАХ В НАЧАЛЬНОЙ ШКОЛЕ // 2022. №9. URL: <https://cyberleninka.ru/article/n/igrovye-tehnologii-na-urokah-v-nachalnoy-shkole> (дата обращения: 22.04.2023).
2. Сидоров А.В., Сидорова Н.Н. ИСПОЛЬЗОВАНИЕ ИГРОВЫХ МЕТОДОВ ПРИ ОБУЧЕНИИ МАТЕМАТИКЕ В НАЧАЛЬНОЙ ШКОЛЕ // Международный журнал гуманитарных и естественных наук. 2023. №2-1 (77). URL: <https://cyberleninka.ru/article/n/ispolzovanie-igrovyyh-metodov-pri-obuchenii-matematike-v-nachalnoy-shkole> (дата обращения: 22.04.2023).
3. Бегиева Белла Муратовна, Кабжихов Астемир Артурович Роль игровой технологии в обучении и воспитании младших школьников // Вопросы науки и образования. 2020. №24 (108). URL: <https://cyberleninka.ru/article/n/rol-igrovoy-tehnologii-v-obuchenii-i-vospitanii-mladshih-shkolnikov> (дата обращения: 22.04.2023).
4. Кукушкина Зинаида Владимировна Цитаты педагогов об игре //ИНФОУРОК 2023. [Электронные ресурсы] URL: <https://infourok.ru/user/kukushkina-zinaida-vladimirovna/blog/citaty-pedagogov-ob-igre-162685.html> (дата обращения 27.04.2023)
5. Umit Tokac, Elena Novak, Christopher G. Thompson Effects of game-based learning on students' mathematics achievement: A meta-analysis // Journal of computer assisted learning. – 2019. – Vol. 35 (3). – P. 407-420.
6. Weijie Mao, Yunhuo Cui, Ming M. Chiu, and Hao Lei Effects of Game-Based Learning on Students' Critical Thinking: A Meta-Analysis // Journal of Educational Computing Research. – 2021. – Vol. 59 (8). – P. 1682-1708.
7. Шипилова, С.С. Использование игровых технологий в процессе обучения предмету «Математика» // Воспитание и обучение детей младшего возраста. – 2018. – № 7. – С. 208- 209.

8. «Загарских А. С., Хорошавин А. А., Александров Э. Э. Введение в разработку компьютерных игр» (Загарских, А. С. Введение в разработку компьютерных игр : учебно-методическое пособие / А. С. Загарских, А. А. Хорошавин, Э. Э. Александров. — Санкт-Петербург : НИУ ИТМО, 2020. — 79 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/190910> (дата обращения: 21.05.2023). — Режим доступа: для авториз. пользователей. — С. 7.).
9. «КАКОЙ ИГРОВОЙ ДВИЖОК ВЫБРАТЬ?» (Виноградов, С.В. КАКОЙ ИГРОВОЙ ДВИЖОК ВЫБРАТЬ? / С.В. Виноградов, К.В. Черкасов // Вестник научного общества студентов, аспирантов и молодых ученых. — 2016. — № 4. — С. 9-14. — ISSN 9999-7444. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/journal/issue/300037> (дата обращения: 21.05.2023). — Режим доступа: для авториз. пользователей. — С. 6.).
10. «Кононова О.В. Проектирование информационно-обучающей веб-среды с элементами геймификации. Вопросы организации текстового и игрового контента» (Кононова, О. В. Проектирование информационно-обучающей веб-среды с элементами геймификации. Вопросы организации текстового и игрового контента : учебное пособие / О. В. Кононова. — Санкт-Петербург : НИУ ИТМО, 2017. — 70 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/110430> (дата обращения: 21.05.2023). — Режим доступа: для авториз. пользователей. — С. 1.).
11. «Формирование функциональной грамотности школьников: Новые дидактические решения» (Формирование функциональной грамотности школьников: Новые дидактические решения : монография / научный редактор И. Ю. Тарханова. — Ярославль : , 2021. — ISBN 978-5-00089-466-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/283373> (дата обращения: 21.05.2023). — Режим доступа: для авториз. пользователей. — С. 232.).
12. «Есин Р. В., Вайнштейн Ю. В. Формирование математической компетентности на основе построения индивидуальной образовательной траектории в электронной среде» (Есин, Р. В. Формирование математической компетентности на основе построения индивидуальной образовательной траектории в электронной среде : монография / Р. В. Есин, Ю. В. Вайнштейн. — Красноярск : СФУ, 2021. — ISBN 978-5-7638-4376-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL:

<https://e.lanbook.com/book/181647> (дата обращения: 21.05.2023). — Режим доступа: для авториз. пользователей. — С. 32.).

13. Документация Unreal Engine 5 : сайт. — URL: <https://docs.unrealengine.com/5.0/en-US/> (дата обращения: 27.04.2023)

14. «Актуальные вопросы совершенствования среднего профессионального и высшего образования в современных условиях» (Актуальные вопросы совершенствования среднего профессионального и высшего образования в современных условиях : материалы конференции / RU. — Новосибирск : СибГУТИ, 2022. — ISBN 978-5-91434-073-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/257324> (дата обращения: 21.05.2023). — Режим доступа: для авториз. пользователей. — С. 53.).

ПРИЛОЖЕНИЕ А:

1. Назначение разработки

Дипломная работа является комплексным проектом, который охватывает все стадии разработки ПО.

2. Стадии и этапы разработки

2.1. Создание и проработка идеи приложения

- продумывание персонажа и его окружения (продумывание графической части проекта);
- продумывание музыкальной составляющей проекта (фоновая музыка, звуки).
- продумывание основных игровых механик
- продумывание дизайна уровней, которые будут отвечать обучающим целям игрового приложения

2.2. Выбор движка

- Приложение будет создано на базе игрового движка Unreal Engine 5

2.3. Реализация графического компонента игры

- разработка окружения героя (локации, объекты для взаимодействия);
- создание уровней

2.4. Разработка программного кода игры

- изучение необходимой для разработки документации;
- разработка непосредственно программного кода игры;
- Создание базовых механик, отвечающим целям игрового обучающего приложения

2.5. Реализация музыкальной/звуковой составляющей проекта

- Поиск или создание основной фоновой музыки;
- Сбор основных фоновых звуков
- Поиск или написание музыки для главного меню и для титров;
- Внедрение музыкального компонента игры в проект;

2.6. Тестирование

- юзабилити-тестирование:
тестирование главного меню;
тестирование базовых внутри игровых механик;

3. Требования к программе или программному изделию

3.1. Требования к функциональным характеристикам

- звуковой функционал:
 - а) регулировка громкости;

- внутриигровой функционал:
 - а) система взаимодействия игровых объектов;

- интерфейс пользователя:
 - а) главное меню;

3.2. Требования к информационной и программной совместимости

- игра должна работать под ОС семейства Windows

3.3. Минимальные системные требования

- ОС: Windows
- оперативная память: 1 ГБ;
- клавиатура, мышь.