**Министерство профессионального образования и занятости**

**населения Приморского края**

**краевое государственное автономное**

**профессиональное образовательное учреждение**

**«ПРИМОРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

**(КГА ПОУ «ППК»)**

Отделение Экономики и сервиса

Специальность *09.02.07*

«Информационные системы и программирование»

Работа допущена к защите:

Завотделением \_\_\_\_\_\_\_ А.А. Савина

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

**КУРСОВОЙ ПРОЕКТ**

**по теме:**

**РАЗРАБОТКА ИГРОВОГО ПРИЛОЖЕНИЯ НА UNITY 3D**

|  |  |
| --- | --- |
| **Выполнил**  студент группы 1125 | **Руководитель**  преподаватель |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_А.М. Куранаков | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Пушкин |

**Владивосток 2023**

**Министерство профессионального образования и занятости населения Приморского края**

**краевое государственное автономное профессиональное**

**образовательное учреждение**

**«ПРИМОРСКИЙ ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ»**

**(КГА ПОУ «ППК»)**

**Отделение Экономики и сервиса**

**З А Д А Н И Е**

на курсовой проект

Заведующий отделением \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Савина

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

|  |
| --- |
| *студенту Куранаков Артем Михайлович 1125 группы* |

(фамилия, имя, отчество)

на тему «Разработка игрового приложения на Unity 3D»

|  |
| --- |
|  |
|  |

Вопросы, подлежащие разработке (исследованию):

|  |
| --- |
| 1. Изучение предметной области. Выделение объектов и связей между ними. |
| 1. Разработка схемы базы данных |
| 1. Построение диаграммы объектов. |
| 1. Написание технического задания на разработку системы |
| 1. Разработка функциональной части приложения |
| 1. Разработка пользовательского интерфейса приложения |
|  |

Основные источники информации и прочее, используемые для разработки темы

|  |
| --- |
| 1. Гаурав Арораа, Джефри Чилберто – Паттерны проектирования для C# и платформы .Net Core, 2021 |
| 1. Эндрю Лок – Asp.Net Core в действии, 2021 |
| 1. Прайс Марк – C# 10 и .Net 6. Современная кросс-платформенная разработка, 2023 |
|  |

Срок предоставления работы «17» мая 2023 г.

Дата выдачи задания «01» апреля 2023 г.

Руководитель КП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.А. Пушкин

(подпись) (и.о. фамилия)

Задание получил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.М. Куранаков

(подпись) (и.о. фамилия)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc135718543)

[1 Разработка системы работы приложения 6](#_Toc135718544)

[1.1. Используемые технологии 6](#_Toc135718545)

[1.2. План приложения 8](#_Toc135718546)

[2 Описание разработанной системы 10](#_Toc135718547)

[2.1. Описание интерфейса пользователя 10](#_Toc135718548)

[2.1.1. Верхняя панель 10](#_Toc135718549)

[2.1.2. Нижняя панель 12](#_Toc135718550)

[2.2. Игровой опыт 12](#_Toc135718551)

[2.2.1. Разработка модели поведения пользователя 13](#_Toc135718552)

[2.2.2. Финансы 16](#_Toc135718553)

[2.2.3. Система строительства 18](#_Toc135718554)

[2.2.4. Логика поведения учащихся 20](#_Toc135718555)

[2.3. Код приложения 22](#_Toc135718556)

[2.3.1. Система текущего действия 22](#_Toc135718557)

[2.3.2. Система строительства 23](#_Toc135718558)

[2.3.3. Система заданий пользователя 24](#_Toc135718559)

[2.3.4. Управление камерой 25](#_Toc135718560)

[2.3.5 Магазин строений 25](#_Toc135718561)

[2.3.6. Система представления средств. 26](#_Toc135718562)

[2.3.7. Система ускорения времени 28](#_Toc135718563)

[ЗАКЛЮЧЕНИЕ 29](#_Toc135718564)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 30](#_Toc135718565)

# ВВЕДЕНИЕ

Колледж — среднее специальное учебное заведение, реализующее основные профессиональные образовательные программы среднего профессионального образования базовой подготовки и программы среднего профессионального образования углублённой подготовки.

В эру развития информационных технологий появляется все больший спрос на обучение в колледже. Современные технологии и доступ к интернету привели к значительному увеличению информации и возможностей для самообразования. Однако, несмотря на это, обучение в колледже остается востребованным и значимым.

Объектом данной курсовой работы является профессиональный колледж «ДВФУ», а предметом – моделирование системы управления современным колледжем.

Целью данного курсового проекта является разработка симулятора управляющего данным учебным заведением. Из этого вытекают следующие задачи:

1. Выявить основные сущности и их взаимосвязи,
2. Сформировать требования к разрабатываемой системе,
3. Определить основные алгоритмы, описывающие работу учебного заведения,
4. Разработать приложение, которое моделирует поведение системы обучения,
5. Протестировать разработанное приложение.

# 1 Разработка системы работы приложения

**1.1. Используемые технологии**

Вот лишь несколько функций языка C#, которые позволяют создавать надежные и устойчивые приложения. Сборка мусора автоматически освобождает память, занятую недостижимыми неиспользуемыми объектами. Типы, допускающие значение null, обеспечивают защиту от переменных, которые не ссылаются на выделенные объекты. Обработка исключений предоставляет структурированный и расширяемый подход к обнаружению ошибок и восстановлению после них. Лямбда–выражения поддерживают приемы функционального программирования. Поддержка языков для асинхронных операций предоставляет синтаксис для создания распределенных систем. В C# имеется Единая система типов. Все типы C#, включая типы–примитивы, такие как int и double, наследуют от одного корневого типа object. Все типы используют общий набор операций, а значения любого типа можно хранить, передавать и обрабатывать схожим образом. Более того, C# поддерживает как определяемые пользователями ссылочные типы, так и типы значений. C# позволяет динамически выделять объекты и хранить упрощенные структуры в стеке. C# поддерживает универсальные методы и типы, обеспечивающие повышенную безопасность типов и производительность. C# предоставляет итераторы, которые позволяют разработчикам классов коллекций определять пользовательские варианты поведения для клиентского кода.

Чтобы обеспечить совместимость программ и библиотек C# при дальнейшем развитии, при разработке C# много внимания было уделено управлению версиями. Многие языки программирования обходят вниманием этот вопрос, и в результате программы на этих языках ломаются чаще, чем хотелось бы, при выходе новых версий зависимых библиотек. Вопросы управления версиями существенно повлияли на такие аспекты разработки C#, как раздельные модификаторы virtual и override, правила разрешения перегрузки методов и поддержка явного объявления членов интерфейса.

Unity — это мощный игровой движок и интегрированная среда разработки (IDE), созданная компанией Unity Technologies. Он используется для разработки различных типов интерактивных приложений, в основном игр, для разных платформ, включая компьютеры, мобильные устройства и игровые консоли.

Unity предоставляет разработчикам широкий набор инструментов и ресурсов для создания игрового контента, таких как трехмерная графика, анимация, физика, звук, искусственный интеллект и многое другое. Этот движок позволяет разработчикам создавать игры и визуальные симуляции, а также взаимодействовать с игровым миром через скрипты и программирование.

Unity имеет много преимуществ и причин его выбрать. Вот несколько из них:

* Кроссплатформенная разработка: Unity позволяет создавать игры для разных платформ, таких как Windows, macOS, Android, iOS, Xbox, PlayStation и многих других, что делает его очень удобным для разработчиков, желающих достичь широкой аудитории,
* Интуитивный интерфейс: Unity имеет простой и интуитивно понятный интерфейс, что делает его доступным для начинающих разработчиков и облегчает работу со сложными системами и компонентами игры,
* Обширный экосистема: Unity обладает обширной экосистемой, которая включает в себя магазин активов (Asset Store), где можно найти готовые модели, текстуры, анимации, скрипты и другие ресурсы для использования в проектах. Также есть активные сообщества разработчиков, где можно получить советы, обменяться опытом и найти решения проблем,
* Скриптовый язык: Unity поддерживает несколько языков программирования, включая C#, JavaScript (теперь устаревший) и Boo. Это дает разработчикам свободу выбора языка, который им наиболее удобен,
* Расширяемость: Unity предоставляет возможность создавать собственные инструменты и плагины для улучшения рабочего процесса и функциональности. Это позволяет разработчикам интегрировать собственные системы и развивать проекты согласно своим потребностям.

Так же стоит упомянуть о Unity–ассетах, или же пакетах, которые были использованы в процессе разработки приложения:

* A\* Pathfinding Project Pro: это мощная и простая версия популярной библиотеки A\* Pathfinding Project для разработки игр. Она предоставляет расширенные функциональные возможности для реализации алгоритма поиска пути A\* в игровых средах,
* Low Poly Ultimate Pack: это набор ресурсов и моделей, созданных в стиле Low Poly, предназначенный для использования в различных проектах, включая игры, визуализации и анимации,
* Basic Motions Pack: это набор анимаций персонажей, который может включать базовые движения, такие как ходьба, бег, прыжки, атаки и другие. Этот набор анимаций разработан с целью предоставить разработчикам игр готовые анимации, которые могут быть легко импортированы и использованы в их проектах.

**1.2. План приложения**

После некоторого локального опроса, темой приложения стала разработка игры в сеттинге “Tycoon”, в котором можно будет управлять своей собственной учебной инфраструктурой.

Tycoon игра — это игра-симулятор, в которой игрок управляет своим бизнесом или империей, стремясь увеличить прибыль и стать богатым. В таких играх игрок может создавать и управлять различными предприятиями, такими как парки развлечений, торговые центры, гостиничные комплексы, фермы и т. д. Он может нанимать персонал, устанавливать цены на товары и услуги, расширять свой бизнес и многое другое.

Для реализации приложения, было решено составить план этого приложения. После анализа и разработки системы был реализован план связей скриптов в приложении. План работы приложения можно посмотреть на рисунке 1.

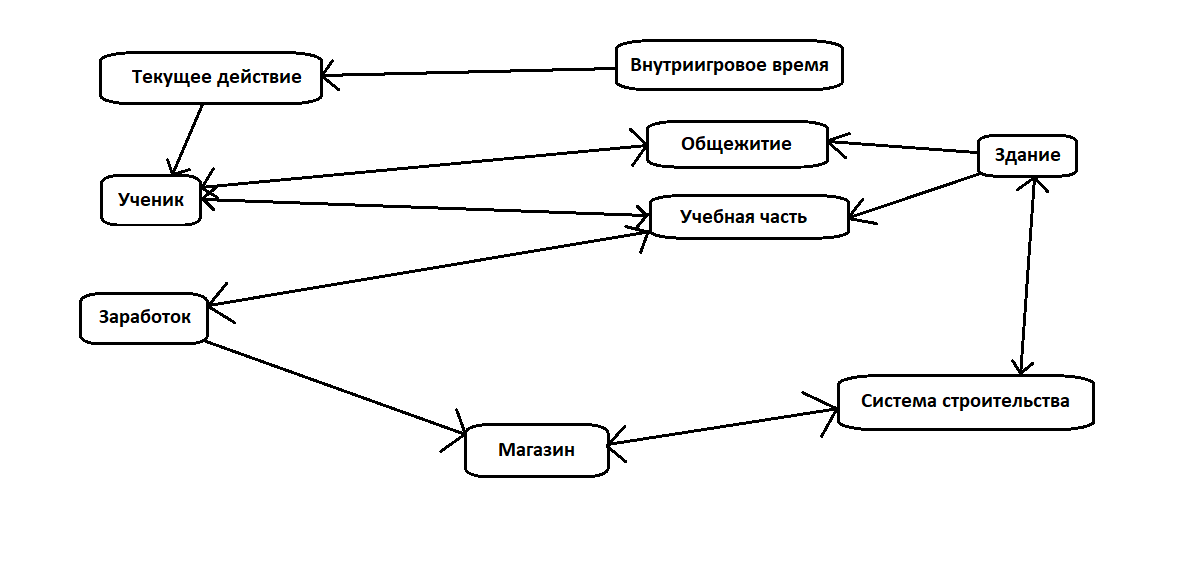


Рисунок 1 - План работы приложения

# 2 Описание разработанной системы

## 2.1. Описание интерфейса пользователя

GUI (Graphical User Interface) — это интерфейс пользователя с использованием графических элементов, таких как кнопки, текстовые поля, списки и окна, для взаимодействия с программой или системой.

GUI позволяет пользователям взаимодействовать с программным обеспечением с помощью интуитивно понятных и удобных визуальных элементов, что делает использование программы более простым и доступным. Он предоставляет пользовательский интерфейс, через который можно вводить данные, просматривать информацию и выполнять различные операции.

Для реализации GUI был использован стандартный набор GUI элементов, который является часть Unity.

### 2.1.1. Верхняя панель

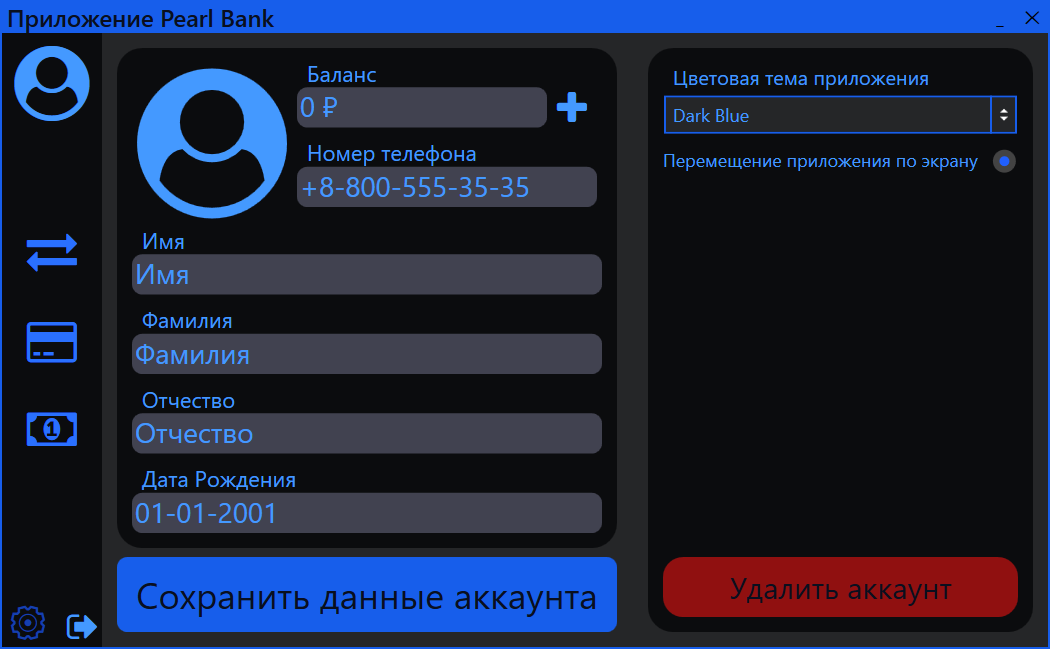
При входе в приложение пользователя встречает несколько панелей, одна из них – верхняя, или же интерактивная панель, которую можно наблюдать на рисунке 2.



Рисунок 2 - Интерактивная панель

Верхняя панель содержит две кнопки, располагающихся по центру, которые позволяют пользователю строить и удалять строения, текстовый блок с внутриигровым временем, который расположился слева, текущий игровой счёт игрока, находящийся справа и текстовый блок с описанием текущего действия, который определяет то, чем учащиеся занимаются в текущее время. Всего в игре есть 3 действия:

* Учёба,
* Сон,
* Свободное время

 При взаимодействии с кнопкой строительства откроется панель с доступными для игрока постройками, изображенной на рисунке 3. Каждая постройка имеет свою цену, название, картинку, а также цвет, который зависит от качества строения.

Качество строений разделяют на:

* Обычное (Зеленый цвет) вмещает в себя 5 учащихся,
* Хорошее (Синий цвет) вмещает в себя 10 учащихся,
* Эпическое (Фиолетовый цвет) вмещает в себя 10 учащихся,
* Легендарное (Оранжевый цвет) вмещает в себя 10 учащихся.

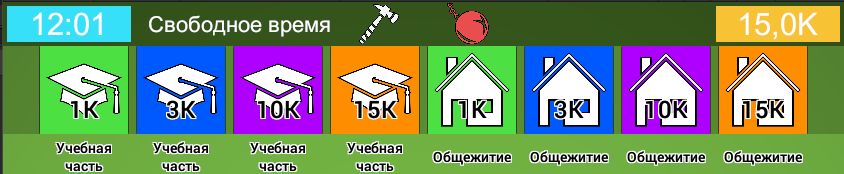


Рисунок 3 – Панель с постройками

### 2.1.2. Нижняя панель

Панель игрового интерфейса расположена в нижней части экрана (рисунок 4) и визуально разделена на три части. Давайте рассмотрим каждую часть по отдельности.

Левая панель, или панель состояния, содержит текущие данные об игровом мире. Здесь отображается:

* Количество учащихся,
* Количество свободных мест в общежитии и учебных частях,
* Текущее настроение и качество обучения учащихся.

Правая часть содержит панель заданий для игрока. Здесь появляются задания в качестве подсказок или предупреждений, чтобы игрок сразу знал, на что следует обратить внимание.

По центру экрана располагается меню ускорения игрового времени. Оно состоит из:

* Текстового блока с отображением текущего ускорения,
* Кнопка паузы,
* Кнопка ускорения,
* Кнопка отключения ускорения.

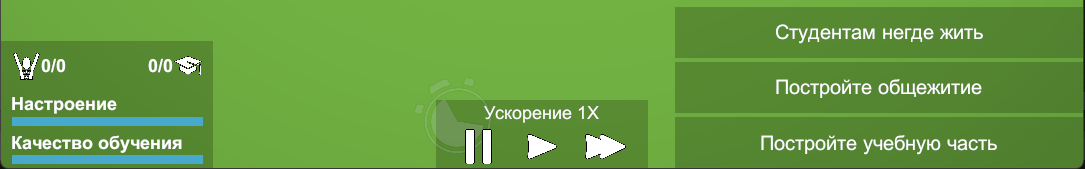


Рисунок 4 – Панель состояния

**2.2. Игровой опыт**

Игровой опыт — это совокупность знаний, навыков, эмоций и впечатлений, получаемых путем участия в игровых активностях. Он охватывает все аспекты игры, включая правила, задачи, взаимодействие с другими игроками, стратегии, решение проблем, а также эстетический и эмоциональный аспекты.

Игровой опыт имеет ряд особенностей, которые делают его ценным и полезным для развития человека:

* Активное участие: Игровой опыт требует активного участия игрока, что способствует его вовлеченности и позволяет применять и развивать различные навыки и способности,
* Учебная среда: Игры могут создавать специальные учебные среды, где игроки могут экспериментировать, исследовать, пробовать разные стратегии и изучать новые концепции. Это способствует активному обучению и запоминанию информации,
* Мотивация и эмоциональная привлекательность: Игровой опыт обычно связан с чувством удовлетворения, достижения целей и преодоления препятствий, что повышает мотивацию игрока. Игры также могут вызывать различные эмоции, такие как радость, волнение, азарт, что делает опыт более интересным и запоминающимся,
* Социальное взаимодействие: Многие игры предоставляют возможность для социального взаимодействия с другими игроками, будь то соревнование или сотрудничество. Это помогает развивать коммуникативные навыки, умение работать в команде и решать проблемы в группе,
* Развитие навыков: Игры могут способствовать развитию различных навыков, включая критическое мышление, проблемное решение, логику, стратегическое планирование, координацию движений, реакцию и другие компетенции.

Игровой опыт имеет широкий спектр применений, от развлечения до образования и профессионального развития. Многие образовательные программы и тренинги используют игровые методики

**2.2.1. Разработка модели поведения пользователя**

Модель поведения игрока (player behavior model) — это абстрактная концептуальная модель, которая описывает и предсказывает типичные действия, решения и взаимодействия игрока в игровой среде. Она основана на изучении и анализе поведения реальных игроков, а также на исследованиях и эмпирических данных, полученных из игровых сессий и экспериментов.

Модель поведения игрока может включать следующие аспекты:

* Игровые предпочтения: Модель учитывает предпочтения игрока в отношении определенных игровых элементов, таких как стиль игры, предпочитаемые стратегии, предпочтительные роли или классы персонажей. Например, некоторые игроки могут предпочитать агрессивную игру и сосредоточиться на сражениях, в то время как другие могут быть более склонны к исследованию и развитию персонажа,
* Решения и стратегии: Модель может описывать типичные решения и стратегии, принимаемые игроком в различных ситуациях игры. Это может включать выбор определенных действий, тактик, путей развития персонажа, распределение ресурсов и принятие решений в условиях неопределенности,
* Социальное взаимодействие: Модель может учитывать взаимодействие игрока с другими игроками и социальные аспекты игры. Это может включать предпочтение сотрудничества или соревнования, выбор командных тактик, степень активного участия в групповых заданиях или влияние на игровое сообщество,
* Прогресс и достижения: Модель может учитывать мотивацию игрока, связанную с прогрессом, достижениями и получением наград в игре. Это может включать стремление к прокачке персонажа, разблокировке новых возможностей, достижению определенных целей, уровней сложности или занятию лидирующих позиций в рейтингах,
* Интересы и мотивация: Модель может учитывать интересы и мотивацию игрока, которые могут варьироваться в зависимости от игрового контекста. Некоторые игроки могут быть мотивированы социальным взаимодействием и соревнованием, в то время как другие могут искать расслабление и удовольствие от исследования виртуальных миров.

Модель поведения игрока является важным инструментом для разработчиков игр, позволяющим лучше понять потребности и предпочтения игроков, а также адаптировать игровой контент и механики для достижения более удовлетворительного игрового опыта. Кроме того, модель поведения игрока может использоваться для создания искусственного интеллекта в играх, чтобы соперники или союзники в игре могли симулировать реалистичные реакции и действия, соответствующие типичному поведению игрока.

При запуске приложения перед нами сразу запускается окно с открытой сценой игры. Первым делом игрок обращает внимание на панель заданий, в которой выведено несколько блоков:

* Студентам негде жить, что дает игроку справку о том, что в его инфраструктуре недостаточно жилых помещений,
* Постройте общежитие, что сигнализирует игроку о том, что у него не построено ни одного общежития,
* Постройте учебную часть, что сигнализирует игроку о том, что у него не построено ни одной учебной части.

После увиденных подсказок игроку будет интуитивно понятно, что ему нужно нажать на строительный молот (Рисунок 5), после чего перед ним откроется панель с постройками (Рисунок 3) и ему нужно будет построить учебную часть и общежитие.

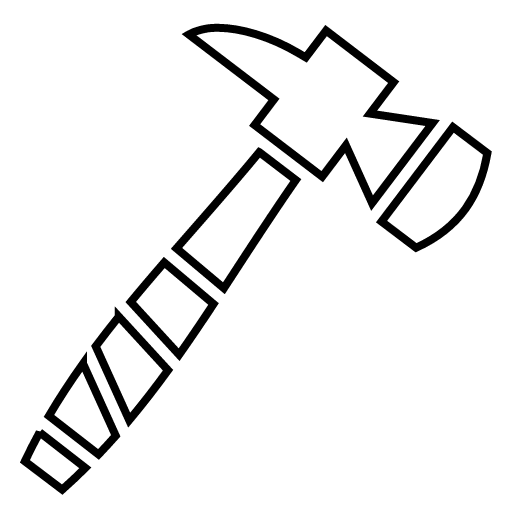


Рисунок 5 – Строительный молоток

После постройки сказанных выше сооружений в панели состояния обновится текущее количество мест в общежитиях и учебных частях. Главное, чтобы количество мест в учебных частях было больше или равно, количеству учащихся, или качество обучения будет падать в связи с тем, что люди просто не смогут поместиться в аудиториях, а вместе с качеством обучения будет снижаться настроение. Если настроение упадет до нуля, то учащиеся просто покинут инфраструктуру и игра закончится.

**2.2.2. Финансы**

Внутриигровая валюта — это средство обмена, используемое внутри

видеоигры для приобретения товаров, услуг и других игровых элементов. Она представляет собой цифровые единицы, которые игроки могут заработать или купить внутри игрового мира и использовать для достижения различных целей.

Внутриигровая валюта может иметь различные названия в разных играх, такие как "золото", "кредиты", "монеты" и так далее. Она может быть заработана путем выполнения заданий, прохождения уровней, победы в сражениях или продажи предметов другим игрокам. Также игроки могут пополнить свой запас валюты, покупая ее с реальными деньгами через внутриигровой магазин или другие механизмы микротранзакций.

В данном проекте у игровой валюты есть свой собственный формат представления, который выглядит так (Рисунок 6):



Рисунок 6 – Формат игровой валюты

Термин "тысяча" обычно обозначается буквой "K" в различных контекстах и ситуациях. Это происходит из-за латинского корня слова "kilo", который означает "тысячу". Буква "K" используется в международной системе единиц СИ для обозначения коэффициента, равного 1000. Таким образом, "K" в контексте числовых значений часто используется для сокращения слова "тысяча".

Например, если говорить о количестве денежных единиц, "10K" означает 10 000, "50K" - 50 000, и так далее. Это распространенное сокращение, которое используется во многих областях, включая финансы, статистику, социальные сети, игры и другие.

Важно отметить, что использование "K" для обозначения тысячи не является универсальным и может различаться в разных странах и культурах.

В данном проекте игровую валюта зарабатывается после окончании каждого занятия или при въезде учащегося в инфраструктуру. Тратить игровую валюту можно на строительство. Также вы получаете 50% от стоимости здания, которое вы снесли.

**2.2.3. Система строительства**

Система строительства по сетке (grid-based building system) в играх — это механика, которая позволяет игрокам строить и размещать объекты, здания, или элементы игрового мира на сетке с фиксированными ячейками. Каждая ячейка сетки представляет собой определенное пространство, куда можно поместить конкретный игровой элемент.

Преимущества системы строительства по сетке:

* Точность и регулярность: Сеточная система обеспечивает точное и регулярное размещение объектов в игровом мире. Это позволяет игрокам создавать симметричные и гармоничные конструкции, а также облегчает выравнивание элементов.
* Удобство планирования: Сетка предоставляет ясную и организованную основу для планирования строительства. Игроки могут предварительно определить местоположение и размеры объектов, что помогает им разрабатывать стратегии и создавать эффективные конструкции.
* Простота и интуитивность: Система построения по сетке обычно довольно проста в использовании и понимании. Она часто основана на простом нажатии или перетаскивании объектов на нужную ячейку, что делает процесс строительства интуитивным и доступным для различных типов игроков.

Система строительства по сетке широко используется во множестве игр различных жанров, таких как симуляторы городов, стратегии в реальном времени, выживание или песочницы. Метод генерации сетки можно увидеть на рисунке 8.

Чтобы начать строить, вам нужно нажать на кнопку строительного молота (Рисунок 4) и перед вами откроется список всех построек (Рисунок 2). После этого выберите строение, которое вам нужно и на которое у вас хватает средств и наведитесь на любое свободное место в игровом мире. В месте куда вы навелись появится предпросмотр здания красного или синего цвета.

Синий цвет обозначает, что все хорошо и вы можете построить здание.

Красный же цвет обозначает, что у вас недостаточно средств или это место уже занято.

Также строения можно вращать, нажатием кнопки R.

Каждое строение может иметь от 5 до 20 мест, в зависимости от качества строения. Это может сэкономить вам место, но вы потратите больше средств. К тому же, чем больше уровень учебной части, тем занятие стоит дороже и приносит вам больше средств.

Чтобы снести строение, вам нужно нажать на шар для сноса зданий, изображенный на рисунке 7, и навести курсор мыши на здание, которое вы хотите снести, оно подсветится красным цветом, после чего можно нажать на него, и оно будет снесено, и вы получите половину от стоимости этого здания.



Рисунок 7 – Шар для сноса зданий

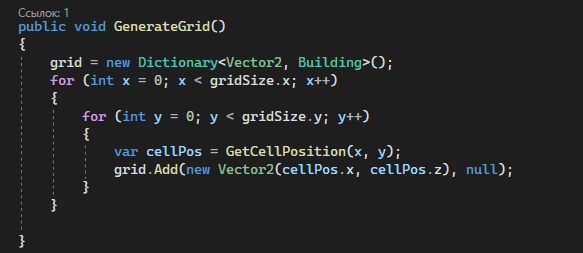


Рисунок - Метод генерации сетки

**2.2.4. Логика поведения учащихся**

Искусственный интеллект (Artificial Intelligence, AI) в играх относится к системам и алгоритмам, которые позволяют компьютерным персонажам или объектам в игре проявлять разумные и автономные действия, имитирующие поведение человека. Целью использования искусственного интеллекта в играх является создание более интересного, динамичного и реалистичного игрового опыта для игроков.

Искусственный интеллект в играх может включать различные компоненты и технологии, вот некоторые из них:

* Поведенческий интеллект: Этот аспект искусственного интеллекта отвечает за принятие решений и контроль над поведением компьютерных персонажей в игре. Он определяет, как персонажи реагируют на окружающую среду, на других персонажей и на действия игрока. Поведенческий интеллект может использовать алгоритмы и методы, такие как конечные автоматы, деревья решений, машинное обучение и т.д., чтобы принимать решения и определять оптимальное поведение в различных ситуациях,
* Интеллектуальный агент: Интеллектуальные агенты — это сущности в игре, которые обладают способностью воспринимать окружающую среду, анализировать информацию, принимать решения и выполнять действия в соответствии с поставленными целями. Это могут быть как непосредственно компьютерные персонажи, так и некие независимые сущности, например, управляющая система виртуального мира. Интеллектуальные агенты в играх часто используются для создания оппонентов, союзников или нейтральных персонажей, с которыми игрок может взаимодействовать,

В данной игре учащиеся полностью зависят от текущего действия, изображенного на рисунке 1.

Если текущим действием будет учёба, то все учащиеся пойдут к случайно выбранному учебному заведению и при достижении точки входа, войдут в него. Если же будет свободное время, то все учащиеся выйдут из зданий и начнут бродить по игровому миру, симулируя прогулку. Если же текущим действием будет сон, то все учащиеся пойду в случайное общежитие, и будут там, до переключения текущего действия на другое. Код выбора следующего места назначения представлен на рисунке 9.

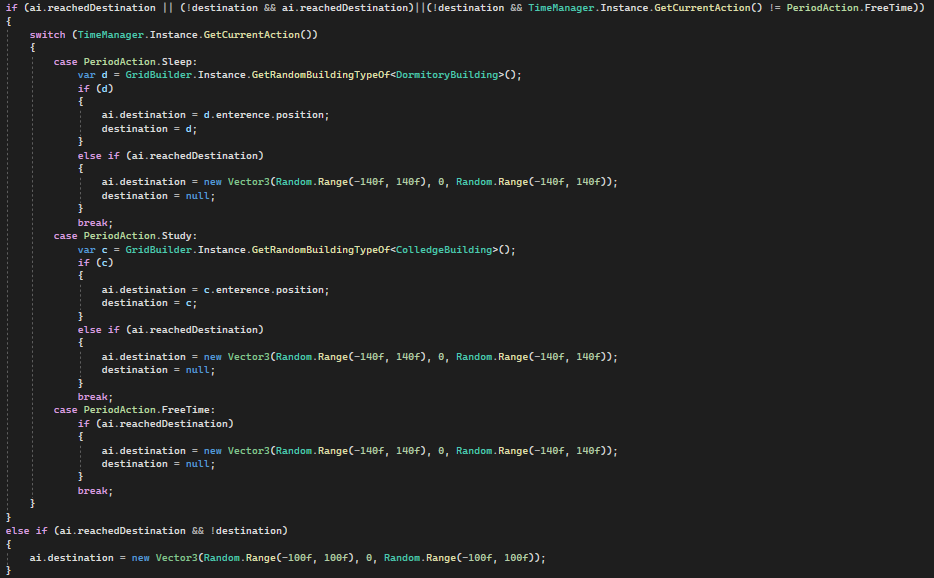


Рисунок 9 - Код выбора следующего места назначения

**2.3. Код приложения**

Для большего понимания работы приложения, нужно понимать, как оно работает изнутри, благодаря чему качество работы и опыт работы с приложением будет гораздо выше.

**2.3.1. Система текущего действия**

Чтобы ученики могли знать куда им идти, им нужно знать текущее действие. Для этого был реализован отдельный скрипт, который за это отвечает.

Для получения текущего действия, нужно получить элемент из массива, в соответствии с началом времени и концом времени этого действия. Метод получения текущего времени представлен на рисунке 10. Список всех действий представлен на рисунке 11.

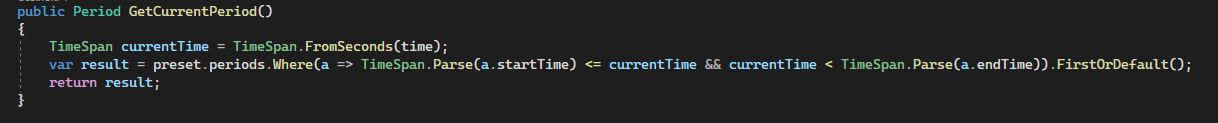


Рисунок - Метод получения текущего действия

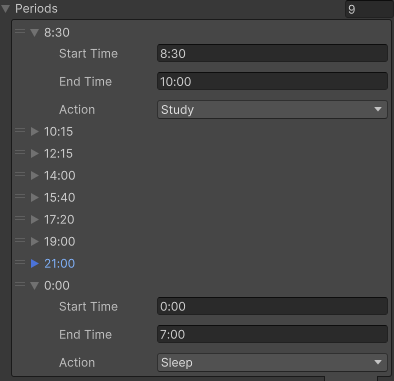


Рисунок - Список всех действий

**2.3.2. Система строительства**

Для строительства сооружений был разработан отдельный скрипт строительства по сетке, который с помощью луча получает точку, после чего высчитывает из этой точки ячейку для строительства. А после с помощью условий разрешает пользователю строить или удалять сооружения. Код системы строительства представлен на рисунке 12.

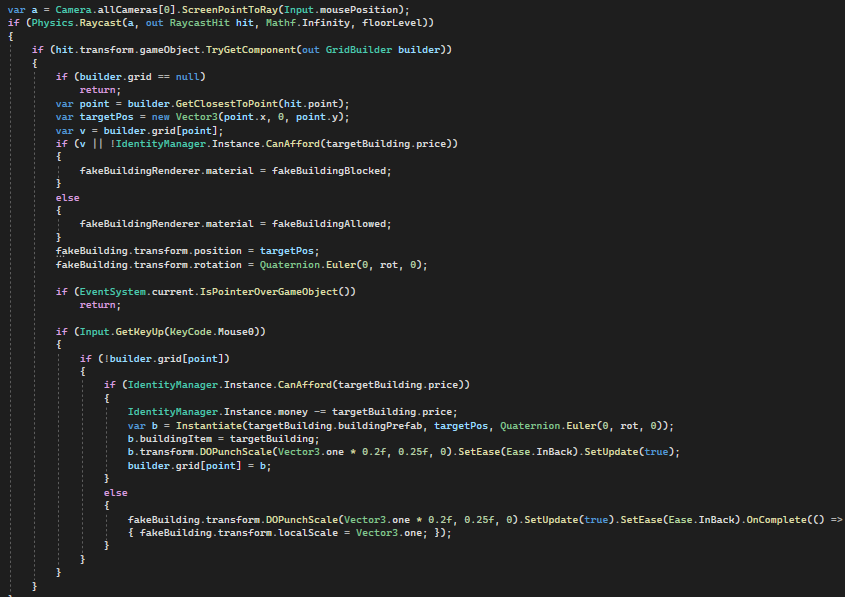


Рисунок - Код системы строительства

**2.3.3. Система заданий пользователя**

Чтобы пользователь не потерялся во время игры, ему представлена панель с заданиями, которые наводят игрока на некие действия.

Для реализации системы заданий, был разработан метод, позволяющий по условию добавлять и удалять задание из списка. А также и сам метод добавления заданий в список. Система заданий пользователя показана на рисунке 13.

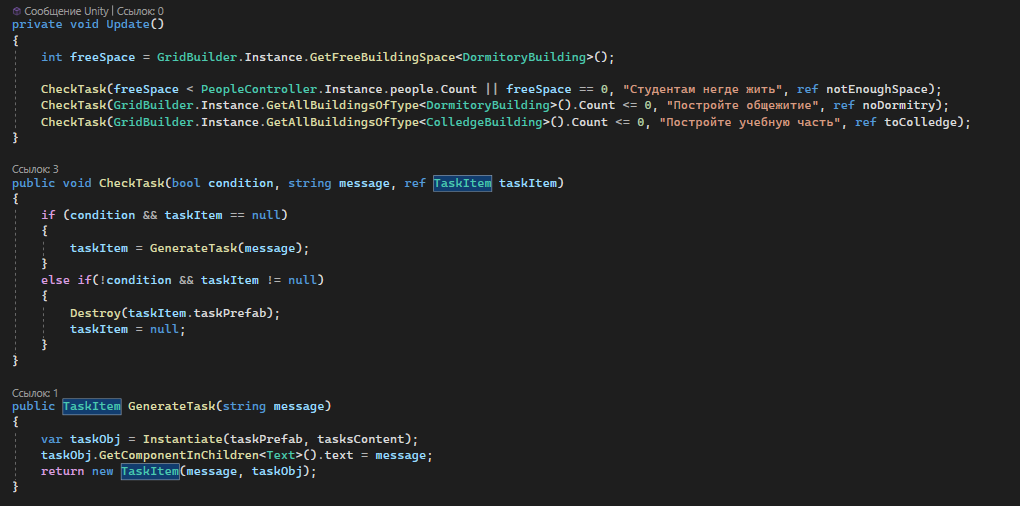


Рисунок - Система заданий пользователя

**2.3.4. Управление камерой**

Чтобы игрок мог свободно управлять своим положением в пространстве, был разработан скрипт, с помощью которого камера может перемещаться в ограниченном пространстве. Код управления камерой представлен на рисунке 14.

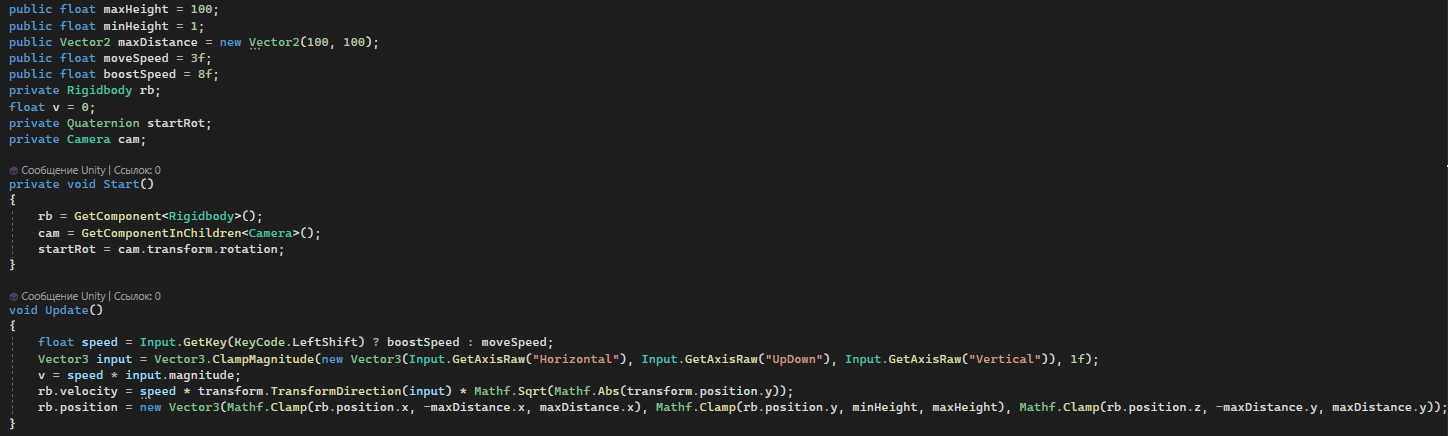


Рисунок - Код управления камерой

**2.3.5 Магазин строений**

Для системы строительств был реализован магазин, в котором можно покупать строения. Для реализации магазина, потребовалось добавить отдельный класс, в котором будут храниться настройки для каждого строения в магазине, такие как цена, название, иконка и т.д. Код магазина строений и предмета в магазине показан на рисуне 15.



Рисунок - Код магазина строений и предмета в магазине

**2.3.6. Система представления средств.**

Для перевода средств из формата 10000 в 10к потребовалось не мало усилий. Для этого нужно сначала найти подходящее окончание нашему числу из таблицы, с помощью представления этого числа в виде степени 10.

Код форматировния числа представлен на рисунке 16. Список всех степеней десятки представлен на рисунке 17.

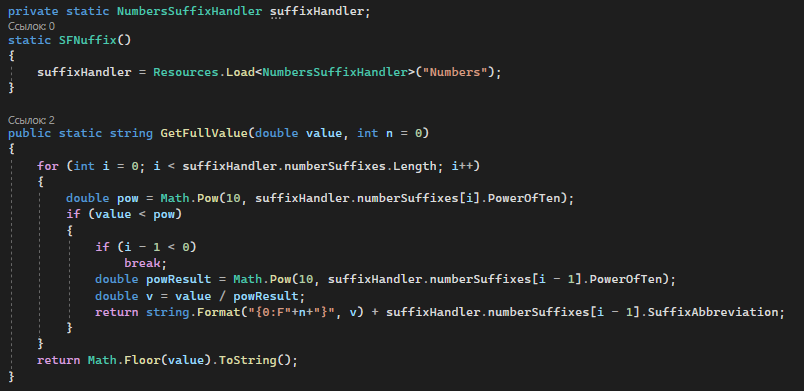


Рисунок - Код форматировния числа

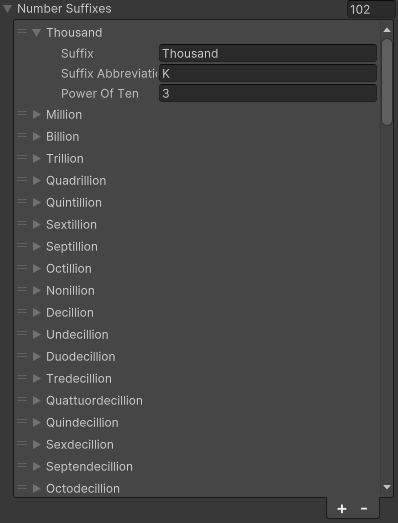


Рисунок - Список всех степеней десятки

**2.3.7. Система ускорения времени**

Для того, чтобы пользователю не стало скучно, была разработана система ускорения времени с помощью кнопок. В Unity уже встроена скорости времени, так что было решено её использовать. Код системы ускорения времени представлен на рисунке 18.



Рисунок - Система ускорения времени

# ЗАКЛЮЧЕНИЕ

Следует рассмотреть результаты проделанной работы по разработке игры, отметить цели, которые были достигнуты и которых можно достигнуть в будущем. Проблемой являлось создание оптимизированной и гибкой системы для имитации поведения учащихся, которая обеспечит быструю и удобную работу всей системы.

Для работы приложения была взята за основу удобная система поиска пути, в которой имеется множество настроек.

Из задач, которые не были достигнуты либо не реализованы должным образом, стоит вынести:

* Оптимизация отображения большого числа объектов,
* Оптимизация системы отслеживания возможных нарушений со стороны пользователей,
* Добавления большего количества настроек в приложение,
* Добавление мини игр,
* Разработка собственного решения поиска пути,
* Управление игрой с мобильного устройства.

В заключении можно отметить, что все основные требования к приложению были выполнены, что означает, что цель работы была достигнута.

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

**Учебники и учебные пособия**

1. Прайс. C#7 и .NET Core. Кросс–платформенная разработка для профессионалов, 2018
2. Скит. C# для профессионалов. Тонкости программирования, 2019
3. Албахари. C# 7.0. Справочник. Полное описание языка, 2018

**Ресурсы сети Интернет**

1. Сайт о программировании язык C# и платформа .NET <https://metanit.com/sharp/> [Электронный ресурс]
2. Руководство по языку C#, краткий обзор язык C# [https://docs.microsoft.com/ru–ru/dotnet/csharp/](https://docs.microsoft.com/ru-ru/dotnet/csharp/) [Электронный ресурс]
3. Электронный учебник по платформе Unity https://learn.unity.com/ [Электронный ресурс]
4. Полный электронный путеводитель по языку C# [https://csharp.net–tutorials.com/](https://csharp.net-tutorials.com/) [Электронный ресурс]
5. C# для начинающих Бесплатный курс по C# [https://ru.code–basics.com/languages/csharp](https://ru.code-basics.com/languages/csharp) [Электронный ресурс]
6. Stackoverflow – вопросы и ответы для программистов <https://ru.stackoverflow.com/> [Электронный ресурс]

**Лист нормоконтроля**

Автор работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.

Нормоконтроль

\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (и.о. фамилия)

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г.