**Министерство науки и высшего образования Российской Федерации**

**Федеральное государственное автономное образовательное учреждение**

**высшего образования**

**«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

ИНСТИТУТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И

ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

КАФЕДРА АНАЛИЗА ДАННЫХ И

ТЕХНОЛОГИЙ ПРОГРАММИРОВАНИЯ

Направление: 09.03.03 – Прикладная информатика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**РАЗРАБОТКА МОБИЛЬНОЙ ИГРЫ С ЭЛЕМЕНТАМИ ОБУЧЕНИЯ ФИНАНСОВОЙ ГРАМОНОСТИ ДЛЯ ДЕТЕЙ**

**Работа завершена:**

Студент 4 курса

группы 09-151

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Иванов Е.А.

**Работа допущена к защите:**

Научный руководитель

к.п.н., ст. преподаватель

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ахмедова А.М.

Заведующий КАДиТП

к.ф.-м.н.

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г.    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Бандеров В.В.

Напишите тему

Казань-2025

Содержание

[Введение 3](#_Toc198674442)

[1. Исследование предметной области разработки игр с элементами обучения финансовой грамотности и выбор средств разработки 6](#_Toc198674443)

[1.1. Анализ предметной области обучения через геймификацию 6](#_Toc198674444)

[1.2. Обзор мобильных игр, обучающих финансовой грамотности 7](#_Toc198674445)

[1.3. Выбор средств разработки 10](#_Toc198674446)

[2. Проектирование мобильной игры с элементами обучения финансовой грамотности 13](#_Toc198674447)

[2.1. Составление технического задания 13](#_Toc198674448)

[2.2. Проектирование взаимодействия игрока с игровыми событиями 16](#_Toc198674449)

[2.3. Проектирование базы данных для мобильной игры с элементами обучения финансовой грамотности для детей 22](#_Toc198674450)

[2.4. Проектирование интерфейса для мобильной игры с элементами обучения финансовой грамотности для детей 24](#_Toc198674451)

[3. Разработка мобильной игры с элементами обучения финансовой грамотности для детей 26](#_Toc198674452)

[3.1. Реализация механик управления процессом игры с элементами обучения финансовой грамотности для детей 26](#_Toc198674453)

[3.2. Реализация пользовательского интерфейса игры с элементами обучения финансовой грамотности для детей 30](#_Toc198674454)

[3.3. Разработка системы взаимодействия с игровыми данными 33](#_Toc198674455)

[4. Тестирование мобильной игры с элементами обучения финансовой грамотности для детей 35](#_Toc198674456)

[4.1. Тестирование пользовательского интерфейса мобильной игры 35](#_Toc198674457)

[4.2. Тестирование механик взаимодействия игрока с игровыми событиями 35](#_Toc198674458)

[Заключение 38](#_Toc198674459)

[Список использованных источников 39](#_Toc198674460)

Приложения

# Введение

Современное общество стремительно развивается в условиях цифровой трансформации, что оказывает значительное влияние на все сферы жизни, включая образование. Финансовая грамотность становится неотъемлемой частью успешной социализации личности, особенно в условиях растущей сложности экономических отношений. Однако уровень финансовой грамотности среди детей и подростков остается недостаточно высоким. Традиционные методы обучения зачастую не учитывают особенности восприятия информации современным поколением, которое активно взаимодействует с цифровыми технологиями. В связи с этим возникает потребность в новых подходах к обучению, способных эффективно вовлекать детей в процесс освоения финансовых знаний.

Несмотря на осознание важности финансовой грамотности, существующие методы ее преподавания часто оказываются малоэффективными для детей. Традиционные формы обучения (лекции, учебники) не всегда учитывают игровую и интерактивную природу восприятия информации современными школьниками. Это приводит к снижению мотивации и слабому усвоению материала.

В связи с этим возникает необходимость в разработке новых подходов к обучению финансовой грамотности, которые были бы не только эффективными, но и увлекательными для детей. Одним из таких подходов является использование мобильных игр, которые сочетают в себе элементы развлечения и обучения. Игровые технологии позволяют вовлечь ребенка в процесс обучения, сделать его интерактивным и мотивирующим, что способствует лучшему усвоению знаний и формированию практических навыков.

Целью данной выпускной квалификационной работы является разработка мобильной игры с элементами обучения финансовой грамотности для детей.

Задачи исследования:

1. Исследовать современные подходы к созданию игр с элементами обучения финансовой грамотности, изучить аналоги, определить оптимальные средства разработки.
2. Разработать концепцию игры, включая сюжет, механики, систему обучения финансовой грамотности, а также спроектировать пользовательские сценарии, интерфейсы и логику взаимодействия игрока с контентом.
3. Разработать функциональный прототип мобильной игры, реализовав основные игровые механики, систему прогресса, интерактивные задания по финансовой грамотности и адаптивный интерфейс, подходящий для детского восприятия.
4. Провести проверку корректности работы всех игровых систем, протестировать интерактивные элементы на удобство взаимодействия и соответствие задуманному UX/UI-дизайну.

Объектом исследования является разработка мобильных приложений с элементами геймификации для обучения детей финансовой грамотности, включая современные инструменты и технологии реализации подобных проектов.

Предметом исследования является процесс создания программной архитектуры и реализации мобильной игры на движке Unity, включая проектирование игровых механик, UI/UX-дизайна для детей и алгоритмов подачи обучающего контента по финансовой грамотности.

Результаты исследования могут быть использованы в образовательных учреждениях, а также для самостоятельного обучения детей основам финансовой грамотности. Разработанная игра станет инструментом, который сочетает обучение и развлечение, повышая интерес детей к финансовой грамотности.

Структура выпускной квалификационной работы включает: введение, четыре главы, заключение, список использованных источников и приложений.

Пока оставим, позже вернемся еще ко введению

# 1. Исследование предметной области разработки игр с элементами обучения финансовой грамотности и выбор средств разработки

## 1.1. Анализ предметной области обучения через геймификацию

Геймификация, или использование игровых элементов в неигровых контекстах, в последние годы приобретает всё большую актуальность. Этот подход активно применяется для повышения мотивации, вовлеченности и эффективности обучения, особенно среди детей и подростков. В контексте обучения финансовой грамотности геймификация представляет собой мощный инструмент, позволяющий превратить сложные экономические понятия в увлекательные и понятные игровые механики.

Геймификация базируется на принципах игрового дизайна, которые включают в себя использование таких элементов, как уровни, награды, рейтинги, задачи и обратная связь. Эти элементы стимулируют интерес пользователя, создают ощущение прогресса и достижений, что особенно важно для детей, которые часто испытывают трудности с концентрацией внимания при обучении в традиционной форме.

Исследования в области педагогики и психологии показывают, что игровые методы обучения способствуют:

* повышению мотивации – дети охотнее участвуют в процессе, если он воспринимается как игра;
* улучшению запоминания – интерактивные и визуальные элементы помогают лучше усваивать информацию;
* развитию критического мышления – игровые задачи часто требуют решения проблем и принятия решений;
* формированию практических навыков – игровые симуляции позволяют отработать действия в безопасной среде.

Финансовая грамотность – это сложная тема, которая включает в себя понимание таких понятий, как бюджет, сбережения, инвестиции, кредиты и риски. Для детей эти темы могут казаться абстрактными и скучными, если они подаются в традиционной форме. Геймификация позволяет сделать их более доступными и интересными.

Примеры успешного применения геймификации в обучении финансовой грамотности:

Игры-симуляторы – например, игры, где ребенок управляет виртуальным бюджетом, учится планировать расходы и принимать финансовые решения.

Квесты и миссии – задания, которые требуют от игрока применения финансовых знаний для достижения целей.

Системы наград – бонусы и достижения за выполнение задач, что стимулирует к дальнейшему обучению.

Анализ предметной области показывает, что геймификация представляет собой эффективный инструмент в процессе обучения детей финансовой грамотности. Данный подход позволяет преодолеть трудности, связанные с пониманием абстрактных экономических понятий, а также делает процесс обучения более интересным и доступным для восприятия. Однако для достижения наилучших результатов важно учитывать возрастные особенности целевой аудитории, разрабатывать качественный и адаптированный образовательный контент, а также обеспечивать высокий уровень интерактивности в процессе обучения. Это позволяет не только удерживать внимание детей, но и способствует более глубокому усвоению материала.

## 1.2. Обзор мобильных игр, обучающих финансовой грамотности

В настоящее время на рынке мобильных приложений существует множество игр, направленных на обучение детей финансовой грамотности. Эти игры используют различные подходы к подаче материала, сочетая образовательные элементы с игровыми механиками. В данном разделе будут рассмотрены две игры: «Барбоскины: Супермаркет» и «Три кота: Финансы для детей», а также отмечены их сильные и слабые стороны.

1. Барбоскины: Супермаркет

Игра «Барбоскины: Супермаркет» – это красочная и увлекательная игра для детей, основанная на популярном мультсериале "Барбоскины". В ней игроки помогают персонажам семьи Барбоскиных совершать покупки в супермаркете, следуя списку, составленному мамой.

Цель игры – помочь персонажам собрать все товары из списка покупок, который мама подготовила для похода в супермаркет. Игрок должен внимательно следить за витринами, находить нужные товары и класть их в корзину, избегая при этом лишних покупок, которые могут добавлять другие члены семьи.

Положительные стороны:

* узнаваемые персонажи: использование героев мультфильма делает игру привлекательной для детей, так как они уже знакомы с этими персонажами;
* простота: механики игры понятны даже для детей младшего возраста, что делает ее доступной для широкой аудитории;
* обучение через практику: ребенок учится считать деньги, планировать покупки и принимать решения в игровой форме.

Слабые места:

* ограниченный образовательный контент: игра фокусируется на базовых навыках (счет, покупки), но не затрагивает более сложные темы, такие как сбережения, инвестиции или кредиты;
* отсутствие персонализации: игра не адаптируется под уровень знаний ребенка, что может снизить ее эффективность для детей с разным уровнем подготовки;
* повторяемость: со временем игровой процесс может стать однообразным, что снижает интерес к обучению.

1. «Три кота: Финансы для детей»

Игра создана по мотивам мультсериала «Три кота» и направлена на обучение детей основам финансовой грамотности. В игре представлены различные сценарии, связанные с управлением бюджетом, планированием расходов и принятием финансовых решений. Ребенок учится распределять ресурсы, копить деньги и избегать ненужных трат.

Цель игры – помочь детям освоить базовые понятия финансовой грамотности, научиться разумно обращаться с деньгами и развить навыки предпринимательства. Всё это подается в легкой и увлекательной форме, чтобы обучение было интересным и понятным для ребенка.

Положительные стороны:

* образовательный уклон: игра охватывает более широкий спектр тем, чем «Барбоскины: Супермаркет», включая сбережения и планирование бюджета;
* яркий дизайн: красочная графика и анимация, характерные для мультсериала, привлекают внимание детей;
* мини-игры: разнообразные задания помогают закрепить полученные знания в игровой форме;
* социальный аспект: игра учит детей делиться и помогать другим, что способствует формированию социальной ответственности.

Слабые места:

* сложность для младших детей: некоторые задания могут быть сложными для детей младшего возраста, что требует участия родителей;
* недостаток глубины: несмотря на более широкий охват тем, игра все же не затрагивает такие важные аспекты, как инвестиции или риски;
* ограниченная интерактивность: некоторые сценарии могут показаться линейными и недостаточно вовлекающими.

Проведенный обзор показал, что современные мобильные игры для обучения финансовой грамотности, такие как «Барбоскины: Супермаркет» и «Три кота: Финансы для детей», успешно используют геймификацию для вовлечения детей. Они предлагают интерактивные механики, знакомые персонажи и базовые финансовые концепции, что делает обучение более доступным и увлекательным. Однако у этих решений есть ряд ограничений:

* узкий образовательный охват – игры фокусируются на простых темах (счет, покупки), но игнорируют сложные аспекты (инвестиции, кредиты, риски);
* отсутствие адаптивности – контент не подстраивается под уровень знаний ребенка, что снижает эффективность обучения;
* повторяемость и линейность – со временем игровой процесс становится предсказуемым, уменьшая мотивацию.

Учитывая выявленные в результате анализа ограничения, было решено разработать мобильное игровое приложение с расширенным образовательным контентом, которое будет включать не только базовые навыки (счет, бюджет), но и продвинутые темы (сбережения, инвестиции, финансовые риски).

Выше написанное относится к 1 выводу, а отсутствие адаптивности и повторяемость и линейность будут решены в Вашем приложении?

## 1.3. Выбор средств разработки

Для реализации мобильной игры с элементами обучения финансовой грамотности для детей был проведен анализ современных инструментов разработки, учитывая такие критерии, как кроссплатформенность, простота использования, производительность и доступность документации. На основе проведенного анализа были выбраны следующие средства разработки: игровой движок Unity, язык программирования C#, среда разработки Visual Studio и СУБД SQLite.

Игровой движок Unity был выбран в качестве основного инструмента для разработки игры. Unity является одним из наиболее популярных движков для создания 2D- и 3D-игр, что обусловлено следующими преимуществами:

* кроссплатформенность: Unity позволяет разрабатывать игры для различных платформ, включая iOS, Android, Windows и другие, с минимальными изменениями в коде. Это особенно важно для мобильной игры, которая должна поддерживаться на разных устройствах;
* гибкость: движок поддерживает как 2D, так и 3D-графику, что позволяет создавать визуально привлекательные и интерактивные проекты;
* большое сообщество и документация: Unity имеет обширную базу знаний, множество обучающих материалов и активное сообщество разработчиков, что упрощает процесс разработки и поиск решений для возникающих задач;
* интеграция с другими инструментами: Unity легко интегрируется с Visual Studio, что упрощает процесс написания и отладки кода.

Для написания скриптов и реализации игровой логики был выбран язык программирования C#. Этот выбор обусловлен следующими причинами:

* интеграция с Unity: C# является основным языком программирования для Unity, что обеспечивает максимальную совместимость и производительность;
* простота и читаемость: C# обладает понятным синтаксисом, что делает его подходящим для разработчиков разного уровня подготовки;
* богатая стандартная библиотека: C# предоставляет обширный набор инструментов для работы с различными функциями, такими как работа с файлами, сетью и базами данных;
* поддержка сообщества: C# активно развивается, а его сообщество предоставляет множество ресурсов для обучения и решения проблем.

Для написания и отладки кода была выбрана интегрированная среда разработки Visual Studio. Выбор обусловлен следующими преимуществами:

* интеграция с Unity: Visual Studio является рекомендуемой средой разработки для Unity, что обеспечивает удобную отладку, автодополнение кода и другие полезные функции;
* мощные инструменты отладки: Visual Studio предоставляет продвинутые инструменты для поиска и исправления ошибок, что ускоряет процесс разработки;
* поддержка расширений: в Visual Studio доступно множество расширений, которые могут упростить работу с кодом, базами данных и другими аспектами разработки;
* кроссплатформенность: Visual Studio поддерживает разработку для различных платформ, включая Windows, macOS и Linux.

Для хранения данных игры, таких как прогресс пользователя, настройки и другая информация, была выбрана встраиваемая реляционная система управления базами данных SQLite. Выбор SQLite обусловлен следующими преимуществами:

* легкость и компактность: SQLite не требует установки и настройки сервера, что делает ее идеальной для мобильных приложений;
* простота использования: SQLite поддерживает стандартный SQL, что упрощает работу с базой данных;
* высокая производительность: SQLite обеспечивает быстрый доступ к данным, что важно для мобильных игр;
* кроссплатформенность: SQLite поддерживается на всех основных платформах, включая iOS и Android.

# 2. Проектирование мобильной игры с элементами обучения финансовой грамотности

## 2.1. Составление технического задания

Целью данного проекта является создание прототипа мобильной игры, в которой будут содержаться элементы обучения финансовой грамотности, которые будут понятны людям, не имеющих знаний в данной сфере, и особенно детям. На примере простых, понятных обыденному пользователю моментах, в игре будут показаны базовые элементы взаимодействия с некоторыми инструментами инвестиций и заработка. Пользователи, проводя время в игре, будут получать навыки работы с данными инструментами, которые в будущем смогут применять уже в реальных условиях. Основной аудиторией этой игры являются дети, для которых обучение сложным и скучным вещам в более располагающей для них обстановке будет иметь большую пользу.

Игра будет представлять собой симулятор цветочного магазина, владельцем которого будет являться игрок. Основной целью игры является развитие своего магазина: расширение ассортимента, увеличение статуса магазина и заработок денег. Заработанные деньги игроку предстоит тратить на улучшение состояния магазина. Чтобы заработать, игрок должен покупать цветы для последующей перепродажи, анализ популярности цветов будет помогать игроку в выборе тех, которые принесут ему больше прибыли.

Для реализации описанной задачи были предъявлены следующие требования:

Требования к функционалу:

* покупка цветов по рыночной цене и последующая перепродажа;
* просмотр графика популярности цветов, для анализа и принятия решения по поводу вложения в конкретный вид;
* самостоятельное выставление цен на цветы при перепродаже;
* наём и увольнение сотрудников и указание условий работы (ставка в час);
* просмотр статистики магазина, для принятия дальнейших решений по развитию магазина;
* возможность брать деньги в кредит.

Ниже приведены визуализации взаимодействий пользователя с разрабатываемой системой. На рисунке 1 изображена диаграмма вариантов использования, на рисунке 2 – диаграмма последовательности действий при организации рабочего дня.

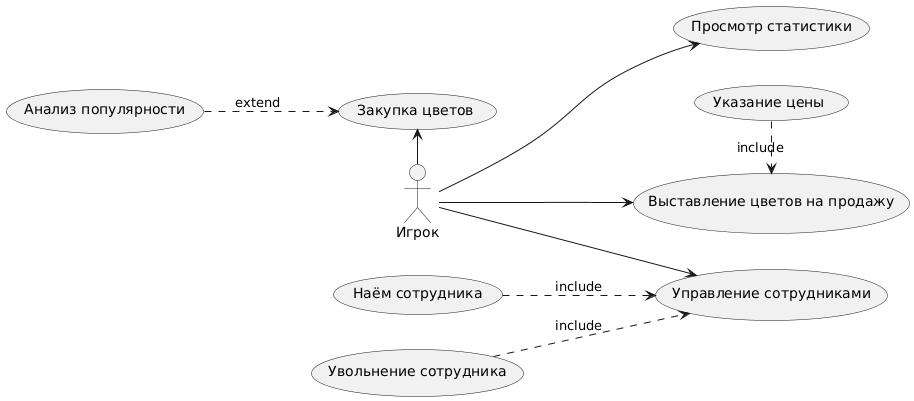


Рисунок 1. Диаграмма вариантов использования

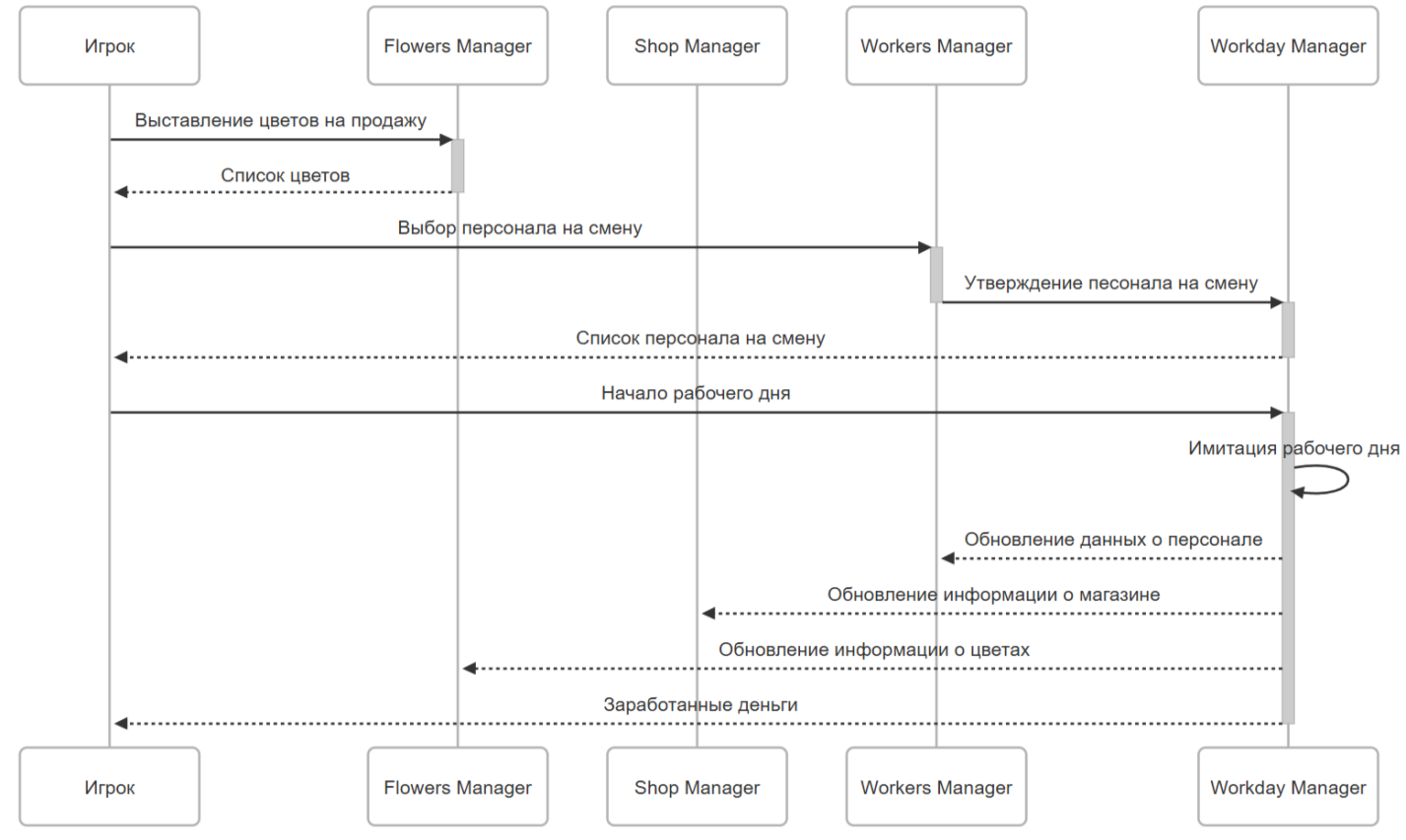


Рисунок 2. Диаграмма последовательности

Технические требования:

* игровой движок: Unity. Данный движок прост в изучении и позволяет разрабатывать кроссплатформенные игры, что позволит увеличить аудиторию игры;
* язык программирования: C#. Основной язык программирования, использующийся для разработки на Unity;
* база данных: SQLite. Позволяет создавать локальную базу данных для хранения прогресса игрока. Также может развёртываться на разных типах устройств.

Дизайн и интерфейс:

* интерфейс должен быть простым и понятным, чтобы пользователи с лёгкостью могли найти необходимы им раздел;
* адаптивность – интерфейс должен подстраиваться под размеры экрана устройства, на котором запускается игра, чтобы он был одинаково удобен на всех устройствах.

В результате должен получиться прототип игры, реализующий все функции, и выполняющий свою основную цель – помощь в изучении инструментов финансовой грамотности.

Добавьте диаграмму

## 2.2. Проектирование взаимодействия игрока с игровыми событиями

Проектирование архитектуры игры является важным этапом разработки, влияющим на проект. Правильно спроектированная архитектура позволяет упростить разработку и избежать различных проблем в процессе.

В процессе анализа требований к функциональной составляющей игры, было принято решение сделать следующую архитектуру.

Проект будет разбит на модули-менеджеры, каждый из которых будет отвечать за свою часть функционала. Все они будут обращаться к главному модулю – менеджеру базы данных, а также модулю для работы с пользовательским интерфейсом.

Database Manager – управляет сохранением и загрузкой данных (бюджет, персонал, цветы, прогресс).

Shop Manager – отслеживает состояние магазина (деньги, дни, рейтинг).

Flowers Manager – управляет ассортиментом, ценами и популярностью цветов.

Workers Manager – отвечает за наём, увольнение и зарплаты сотрудников.

Work Day Manager – координирует игровой цикл (начало/конец дня, события).

UI Manager – отображает игроку все изменения в режиме реального времени.

На рисунке 3 представлена схема, на которой показано, как взаимодействуют между собой модули-менеджеры.

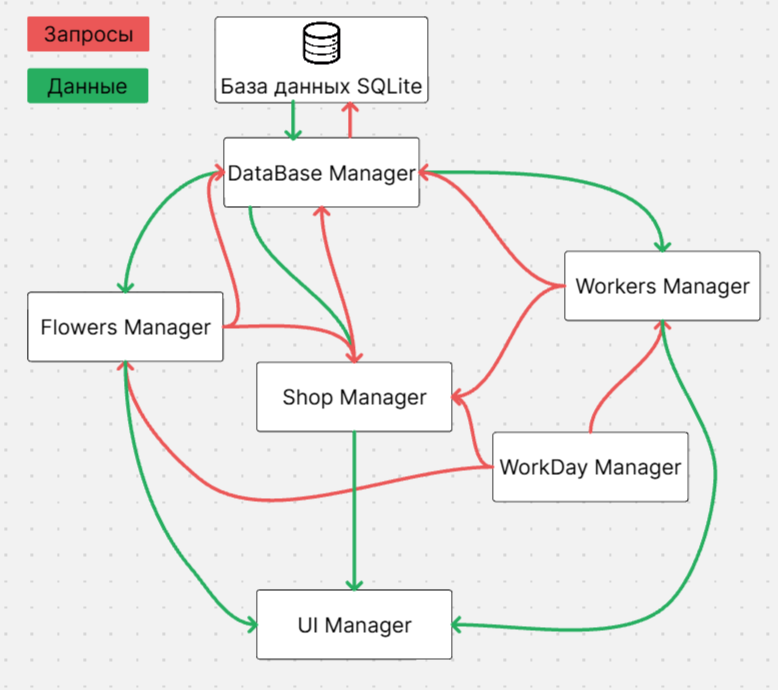


Рисунок 3. Диаграмма взаимодействия модулей-менеджеров

Данный подход с разделением на компоненты, каждый из которых отвечает за конкретную игровую механику, обеспечивает проект следующими преимуществами:

* четкое разделение ответственности (логика цветов, персонала, магазина и UI изолированы друг от друга);
* гибкость – добавление новых функций (например, системы скидок) не требует переписывания существующего кода;
* производительность – менеджеры кэшируют данные и обновляют только нужные части игры;
* простоту тестирования – каждый модуль проверяется отдельно.

Каждый из специализированных менеджеров инкапсулируют бизнес-логику своей предметной области, а также методы обработки игровых событий. Менеджер работы с базой данных, отвечает только за сохранение и изменение игровой информации. Менеджер работы с пользовательским интерфейсом получает уведомления об изменениях состояний и автоматически отображает новую информацию.

Имеется следующий список основных функций:

1. Покупка и выставление цветов на продажу.
2. Просмотр графика популярности цветов на рынке.
3. Наём и увольнение сотрудников.
4. Старт рабочего дня.
5. Указание часовой ставки в течение рабочего дня.
6. Просмотр статистики.

Далее рассмотрим по отдельности каждую из приведённых функций:

1. Покупка и выставление цветов на продажу. На рынке будут отображаться цветы с указанием их рыночной стоимостью. Игрок вводит количество цветов, которое хочет приобрести и со счёта магазина снимаются деньги. На рисунке 4 показано, как цветы отображаются на рынке.

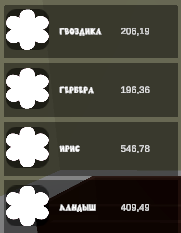


Рисунок 4. Отображение цветов на рынке

На рисунке 5 показан процесс покупки цветов для последующей перепродажи.

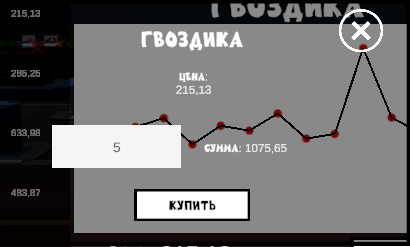


Рисунок 5. Ввод количества цветов

1. Просмотр графика популярности цветов на рынке. Игрок может посмотреть как со временем менялся уровень спроса на каждый цветок в виде графика (рис.6) и, отталкиваясь от этого, принимать решения по покупке цветков. (а расписывать не будешь?)

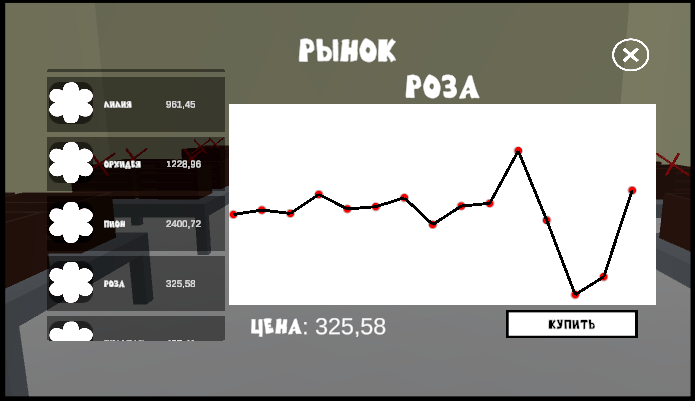


Рисунок 6. Просмотр графика популярности цветов

1. Наём и увольнение сотрудников. На бирже труда игрок может выбрать любого сотрудника и нанять его к себе на работу. Также в списке уже работающих сотрудников игрок может выбрать любого и уволить его. На рисунках 7 и 8 показаны биржа труда, на которой игрок может найти себе сотрудника и нанять его на работу в магазин и список имеющихся рабочих соответственно.



Рисунок 7. Биржа труда сотрудников



Рисунок 8. Нанятые на работу сотрудники

1. Старт рабочего дня. При начале рабочего дня выбираются сотрудники, которые будут обслуживать клиентов (рис. 9).



Рисунок 9. Начало рабочего дня

1. Указание часовой ставки в течение рабочего дня. Каждому сотруднику устанавливается ставка в час, которую он будет получать в течение рабочего дня, которая не может быть ниже минимальной ставки, зависящей от рейтинга магазина (рис. 10).

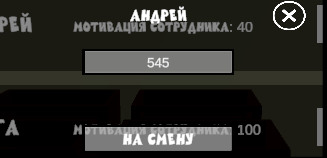


Рисунок 10. Указание часовой ставки для сотрудника

1. Просмотр статистики. После каждой смены можно будет просматривать статистику за рабочий день. Также можно просматривать статистику за всё время (рис.11).

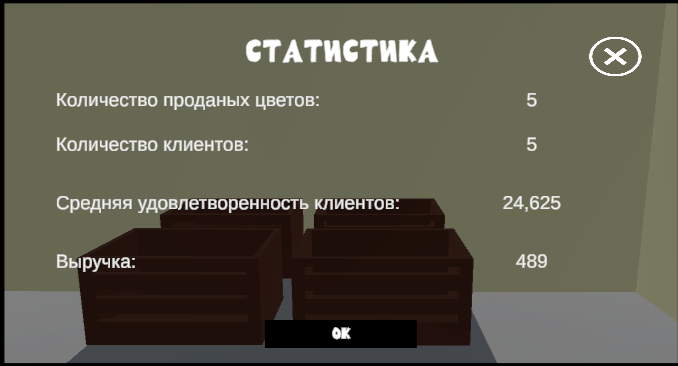


Рисунок 11. Просмотр статистики

## 2.3. Проектирование базы данных для мобильной игры с элементами обучения финансовой грамотности для детей

База данных в мобильной игре с элементами обучения финансовой грамотности для детей играет ключевую роль, обеспечивая сохранение прогресса игрока, управление игровыми сущностями (цветы, сотрудники, финансы) и поддержку сложных механик (динамика цен, статистика спроса, случайные события). Так как игра относится к жанру экономических симуляторов, где важна согласованность данных и возможность их быстрого анализа, была выбрана реляционная СУБД SQLite, оптимальная для мобильных платформ.

Думаю, надо убрать, т. к. внизу уже есть описание сущностей

На рисунке 12 представлена ER-диаграмма базы данных мобильного приложения со всеми таблицами и связями между ними

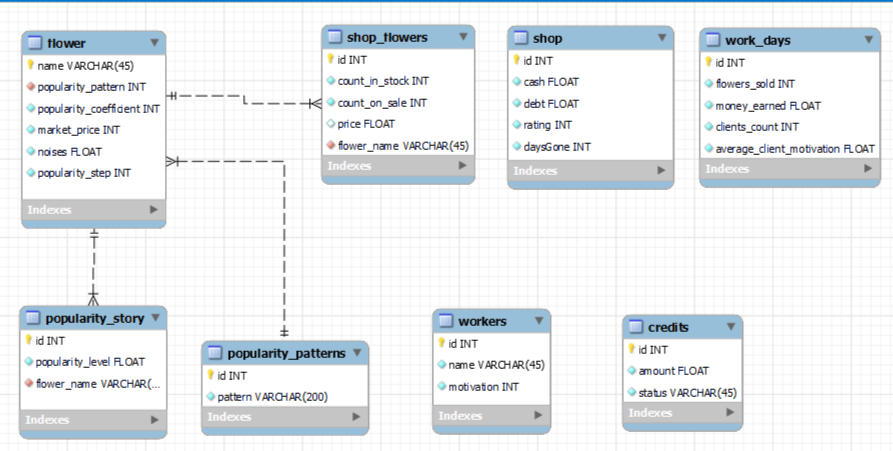
****

Рисунок 12. ER-диаграмма базы данных

Разберём каждую сущность:

1. Flower. Содержит название цветка, его рыночную стоимость, шумы, коэффициент популярности, историю популярности и шаг популярности.
2. Popularity story. На каждый цветок содержит значение его уровня популярности на каждом этапе.
3. Popularity patterns. Содержит в текстовом виде цепочку изменений уровня популярности.
4. Shop flowers. Хранит информацию о том, сколько цветов есть в магазине: сколько в продаже, а сколько на складе, цену указанную игроком.
5. Shop. Хранит информацию о количестве денег, количестве пройденных дней, кредитах и рейтинге магазина.
6. Workers. Информация о нанятых работниках, их имена и уровни мотивации.
7. Credits. Информация о заёмных деньгах и статусе их выплаты.
8. Work days. Количество проданных цветов и выручка за эти продажи. Количество обслуженных клиентов и средний уровень их удовлетворенности.

## 2.4. Проектирование интерфейса для мобильной игры с элементами обучения финансовой грамотности для детей

Проектирование пользовательского интерфейса является одним из ключевых этапов при проектировании мобильной игры. В Unity пользовательский интерфейс включает следующие ключевые элементы:

* canvas – контейнер для всех UI-элементов, определяющий их отрисовку и масштабирование;
* event System – система обработки ввода (касания, клики, наведение), обеспечивающая интерактивность интерфейса;
* UI-элементы (кнопки, текстовые поля, слайдеры, изображения) – визуальные компоненты, с которыми взаимодействует пользователь.

Использование данного подхода при проектировании UI имеет ряд преимуществ:

Гибкость и масштабируемость:

* canvas автоматически адаптирует UI под разные разрешения экранов благодаря Anchor и Pivot системам;
* позволяет легко перестраивать интерфейс без изменения исходного кода.

Оптимизация производительности:

* unity использует Batch-рендеринг для UI, объединяя похожие элементы в один Draw Call, что снижает нагрузку на GPU;
* event System оптимизирует обработку событий, минимизируя лишние вычисления.

Удобство разработки:

* встроенные компоненты (Button, ScrollView, InputField) ускоряют создание интерактивных элементов;
* поддержка анимаций и переходов через Animator и UI-эффекты (например, затемнение при наведении на кнопку).

Кроссплатформенность

* event System поддерживает различные устройства ввода (касание, мышь, геймпад), что упрощает портирование игры на разные платформы.

Поддержка локализации и динамического контента:

* легко интегрируется с системами локализации (TextMeshPro, Unity Localization);
* UI можно обновлять в реальном времени (например, отображение изменяющихся показателей здоровья или счёта).

Применение Canvas и Event System обеспечивает эффективное проектирование пользовательского интерфейса, сочетающее в себе удобство разработки, производительность и адаптивность. Такой подход позволяет создавать интуитивно понятный и отзывчивый UI, что критически важно для мобильных игр, где пользовательский опыт играет ключевую роль в успехе проекта.

# 3. Разработка мобильной игры с элементами обучения финансовой грамотности для детей

## 3.1. Реализация механик управления процессом игры с элементами обучения финансовой грамотности для детей

Управление игровым процессом происходит с помощью взаимодействия между модулями-менеджерами.

Flowers Manager. Обрабатывает запросы, связанные с цветами. В этом модуле реализованы механики для покупки цветов и выставления их на продажу и другие изменения информации о цветах. Также через этот модуль можно получить информацию об имеющихся цветах для отображения и взаимодействия. На рисунках 13 и 14 показаны методы для покупки цветка с рынка и получение информации о рыночной цене цветка соответственно.

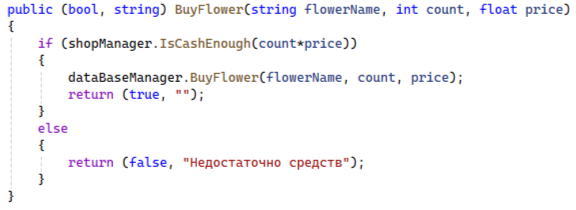


Рисунок 13. Метод покупки цветка

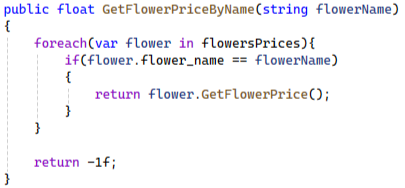


Рисунок 14. Получение информации о цене цветка

Workers Manager. Этот модуль отвечает за механику управления персоналом магазина. В этом модуле реализованы функции для найма (рис. 16) и увольнения сотрудников, а также отправления их на смену. Также в этом модуле содержится функция для генерации новых сотрудников для биржи труда (рис. 15).

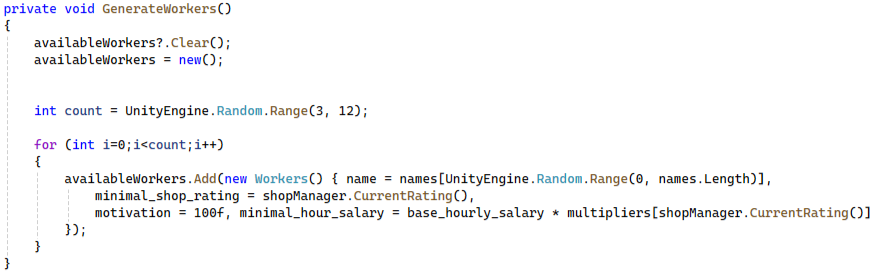


Рисунок 15. Функция для генерации работников

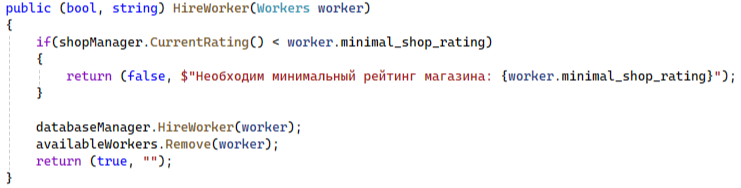


Рисунок 16. Функция для найма работника в магазин

Shop Manager. В данном модуле реализованы механики взаимодействия с данными магазина, такие как зачисления денег и расходы, проверка наличия денег, рейтинг, количество витрин и рабочих мест. На рисунке 17 показаны основные методы для получения информации о состоянии магазина. Функции для изменения состояний магазина представлены в Приложении А.

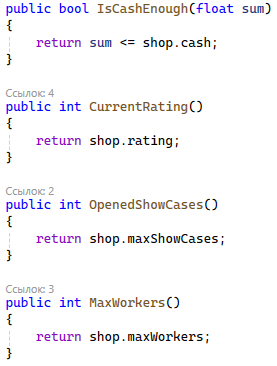


Рисунок 17. Функции для получения актуальной информации о состоянии магазина

Work Day Manager. Этот модуль отвечает за реализацию механики проведения рабочего дня. В этом модуле содержится информация о начале и конце рабочего дня, заработанных деньгах и потоке клиентов. В методе начала рабочего дня (рис. 19) задается время начала смены и запускается непрерывная генерация потока клиентов. В течении указанного времени генерируются клиенты и обслуживаются работниками. По истечению времени, равному длительности смены, выполняется функция окончания рабочего дня (рис. 20), в которой вся информация отправляется другим менеджерам и записывается в базу данных.

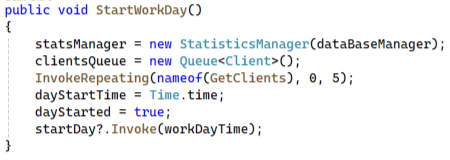


Рисунок 19. Функция начала рабочего дня

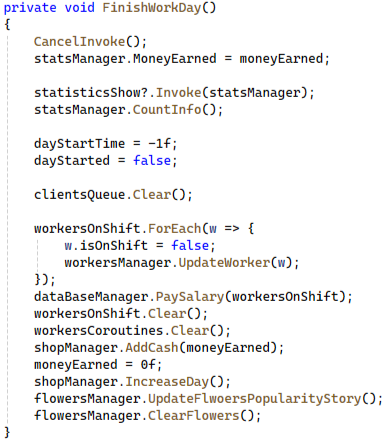


Рисунок 20. Функция окончания рабочего дня

Для реализации обслуживания клиентов, был создан класс, реализующий интерфейс IEnumerator. В методе Update проверяется наличие свободного работника, ему присваивается клиент и запускается корутина. Корутина выполняется в течении времени, которое зависит от базового времени обслуживания клиента и мотивации сотрудника. В зависимости от мотивации работника и ценовых ожиданий клиента рассчитывается уровень удовлетворенности обслуживанием покупателя, что в свою очередь влияет на дальнейшую мотивацию сотрудника. Метод для обслуживания клиентов представлен в Приложении А.

Для генерации клиентов был создан отдельный класс Clients Creator. В этом классе реализован метод GetClients (рис. 21), который вызывается раз в некоторое время и возвращает сгенерированных клиентов. Клиентам присваивается цветок, который они хотят купить, а также максимальную цену, которую они готовы заплатить. Эта цена будет зависеть от уровня рейтинга магазина, а количество клиентов будет зависеть от количества витрин, открытых в магазине.

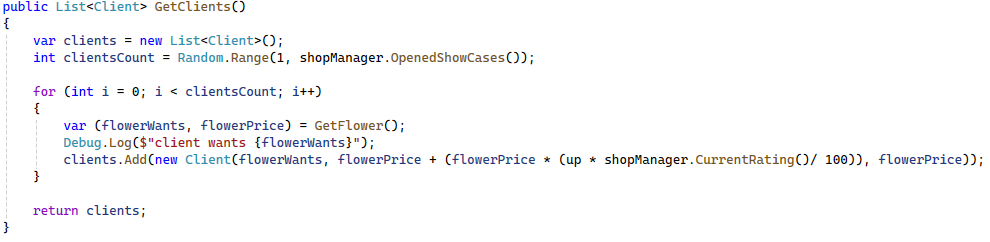


Рисунок 21. Функция генерации клиентов

Этот метод использует другой вспомогательный метод GetFlower (рис. 22), который присуждает клиенту цветок, который тот хочет купить. Цветок выбирается случайно в зависимости от уровня популярности всех цветов.

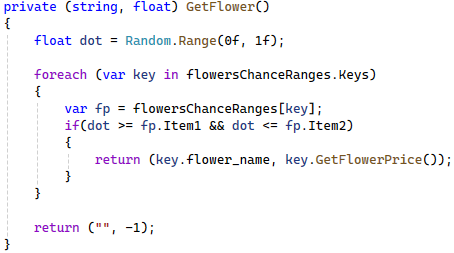


Рисунок 22. Функция для выбора цветка для клиента

## 3.2. Реализация пользовательского интерфейса игры с элементами обучения финансовой грамотности для детей

Разработка пользовательского интерфейса для игры осуществлялась с помощью системы Unity Canvas и Event System, которые предоставляют гибкие инструменты для создания интерактивных элементов. Как и в других современных игровых движках, интерфейс в Unity строится на основе иерархии объектов UI, включающих кнопки, текстовые поля, панели и другие компоненты. Перед непосредственной реализацией в Unity был подготовлен UI/UX-дизайн игры (рис. 24).



Рисунок 24. Заготовленный дизайн главного меню игры

Основные принципы, которых придерживались при разработке интерфейса:

* минимализм – интерфейс не должен перегружать экран и отвлекать от геймплея;
* интуитивность – все элементы (кнопки, иконки, шкалы) должны быть понятны без дополнительных пояснений;
* адаптивность – UI корректно отображается на разных разрешениях экрана.

В Unity пользовательский интерфейс строится на основе Canvas – основного объекта, который определяет область отрисовки UI-элементов. Для корректного отображения был выбран Render Mode = Screen Space – Overlay, чтобы интерфейс отображался поверх игрового мира.

Для обработки взаимодействия игрока с интерфейсом (нажатия кнопок, перетаскивания элементов) использовалась Event System. Эта система автоматически обрабатывает ввод с мыши, клавиатуры и сенсорных устройств, что упрощает реализацию интерактивных элементов. На рисунке 25 показана иерархия элементов пользовательского интерфейса в редакторе Unity.

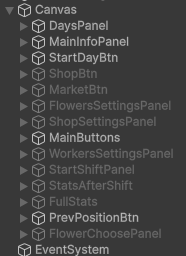


Рисунок 25. Иерархия объектов UI

В игре был выбран минималистичный подход к оформлению интерфейса, чтобы не перегружать экран лишними элементами и сохранить фокус на игровом процессе. Основу визуального стиля составили:

* полупрозрачные черные панели – используются для меню, HUD и диалоговых окон. Такой дизайн обеспечивает хорошую читаемость текста и иконок, не заслоняя при этом происходящее в игре;
* четкие контуры и простые формы – все элементы имеют строгую геометрию без избыточных декоративных деталей.

На рисунке 26 представлена панель, на которой отображается персонал магазина.



Рисунок 26. Панель с отображением персонала магазина

## 3.3. Разработка системы взаимодействия с игровыми данными

Взаимодействие с игровыми данными происходит с помощью библиотеки sqlite-net. Эта библиотека позволяет создавать и изменять базу данных, не прибегая к написанию SQL-кода. Для каждой таблицы был создан свой класс, который описывал структуру в виде полей класса. Пример класса для описания сущности цветов представлен на рисунке 27.

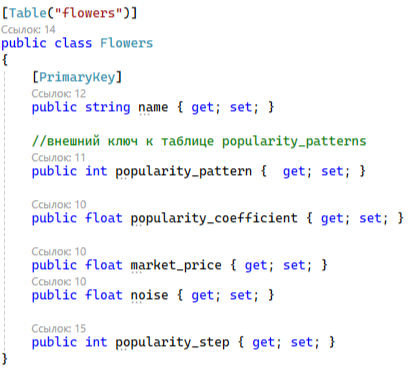


Рисунок 27. Пример класса, описывающего таблицу базы данных

Создание самой базы данных происходит посредством вызовов простой функции CreateTable (рис. 28).

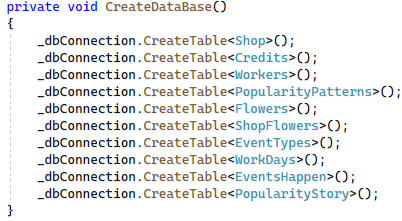


Рисунок 28. Функция для создания базы данных

Для выборки с объединением таблиц создаются дополнительные классы, которые содержат необходимые поля из объединяемых таблиц и создаётся строка, содержащая запрос на объединение таблиц. Пример такой выборке представлен на рисунке 29.

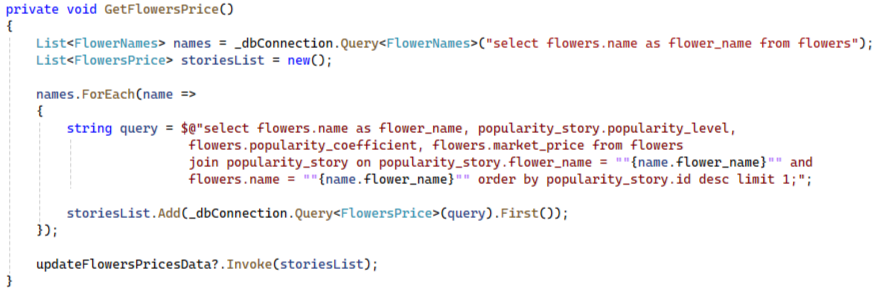


Рисунок 29. Пример функции с выборкой с объединением

В самом коде сделайте перенос длинной строки, чтоб можно было картинку крупнее шрифтом сделать

# 4. Тестирование мобильной игры с элементами обучения финансовой грамотности для детей

## 4.1. Тестирование пользовательского интерфейса мобильной игры

Тестирование пользовательского интерфейса будет проводиться в ручном режиме. Данный подход позволит наглядно оценить реализованный интерфейс, удобство использования, а также корректность отображения элементов.

Задачи тестирования пользовательского интерфейса:

1. Корректное отображение элементов интерфейса.

Результат: тест пройден.

1. Воспроизведение анимации при открытии панелей.

Результат: тест пройден.

1. Единый дизайн всех элементов интерфейса.

Результат: тест пройдет.

1. Интуитивно понятный интерфейс.

Результат: тест пройден.

1. Правильная смена элементов интерфейса. При открытии новой панели, предыдущая (если есть включенная панель) должна закрыться.

Результат: тест пройден.

## 4.2. Тестирование механик взаимодействия игрока с игровыми событиями

Для тестирования взаимодействий игрока с игровыми механиками были разработаны автоматизированные тесты. Написание тестов происходило с использованием фреймворка NUnit. На рисунке 30 представлены результаты прохождения тестов.

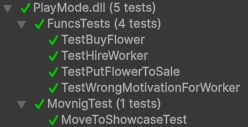


Рисунок 30. Результат прохождения функционального тестирования

Перед началом выполнения тестов производится загрузка игровой сцены. Данная операция представлена на рисунке 31.

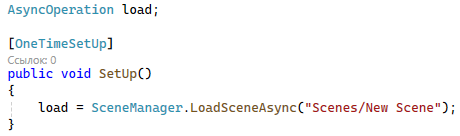


Рисунок 31. Функция для загрузки игровой сцены

На рисунке 32 представлена функция для тестирования выставления цветка на продажу. Для оценки прохождения теста сравнивается количество цветов на складе и на витринах.



Рисунок 32. Функция тестирования выставления цветка на продажу

На рисунке 33 показана функция тестирования передвижения к витринам. Происходит поиск первой витрины на сцене, далее камера поворачивается в сторону витрины и испускается луч, для того чтобы имитировать клик по объекту. Для того чтобы определить успешность прохождения теста сравниваются изначальная и конечная позиция камеры.



Рисунок 33. Функция тестирования передвижения камеры

# Заключение

# Список использованных источников

1. Unity и C# геймдев от идеи до реализации / Д.Г. Бонд. – 2-е изд. – Санкт-Петербург: Прогресс книга, 2019 – 1002 с.

2. Unity Learn [Электронный ресурс]. URL: <https://learn.unity.com/pathway/junior-programmer> (дата обращения: 12.02.2025).

3. Unity Documentation [Электронный ресурс]. URL: <https://docs.unity3d.com/ScriptReference/index.html> (дата обращения 15.02.2025).

4. Язык программирования C# и платформа .NET [Электронный ресурс]. URL: <https://metanit.com/sharp/> (дата обращения: 22.02.2025).

5. Любовь Карасева Геймификация: как игровой подход помогает в обучении и на работе // РБК Тренды [Электронный ресурс]. URL: <https://trends.rbc.ru/trends/education/605c6f2f9a79473a61646994> (дата обращения: 24.02.2025).

В работу желательно добавить различные диаграммы: взаимодействия, вариантов использования, алгоритмов, архитектуры ит.д…. Короче все что из этого и прочего у Вас есть)

**Приложения**

**Приложение А**

Фрагменты кода

На рисунке 18 представлены функции для обновления состояний магазина.

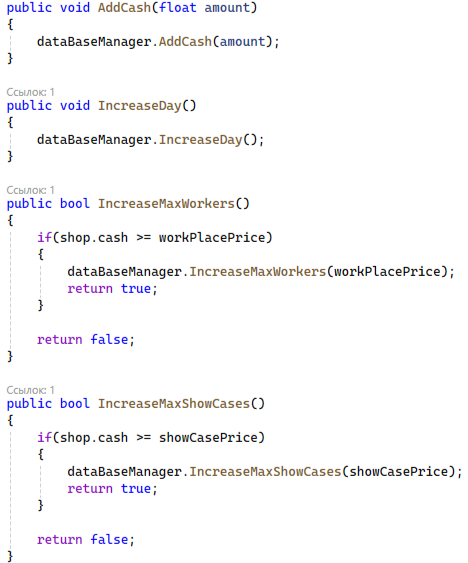


Рисунок 18. Функции для обновления состояний магазина

На рисунке 23 представлена функция для имитации обслуживания клиента.

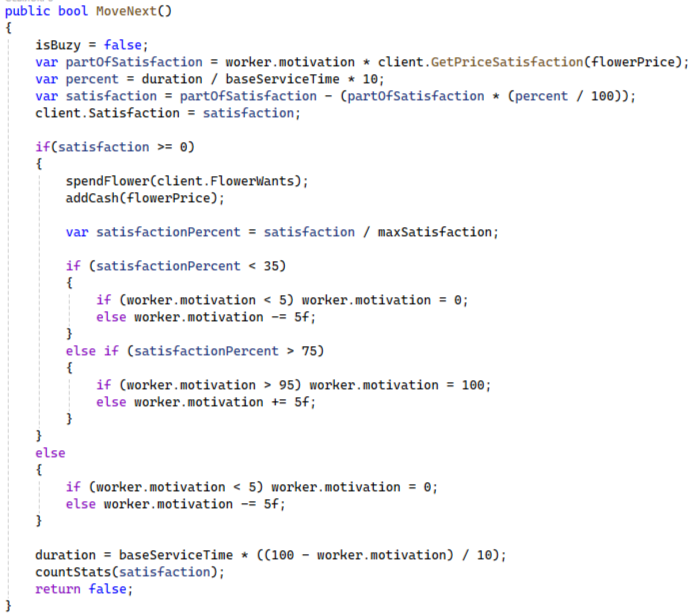


Рисунок 23. Функция для обслуживания клиента