

Введение

На учебную практику была поставлена задача Разработка мобильного игрового приложения в жанре кликер «Meme Cat».

Цель учебной практики заключается в разработке программного продукта, который позволит вам расслабиться и провести хорошо свободное время.

Создаваемая программа будет рассчитана на любого рода пользователей. Применить данную программу могут не только дети, которые хотят занять свое свободное время, но также взрослые, которые хотят просто снять стресс.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы можете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи. Также в этом разделе вы можете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже описаны в первом разделе. В подразделе “Инструменты разработки” рассматривается среда, в которой создается данный курсовой проект. Здесь также установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” рассмотрены основные аспекты разработки программного продукта. Здесь можно узнать об организации данных в контексте среды разработки. В данном разделе четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе четко описаны функции пользователя и их структура. Здесь можно найти таблицу, в которой будет представлена полная аннотация файлов, используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем описано полное и функциональное тестирование данной программы, т.е. оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” описано назначение, область применения, среда функционирования курсовой программы.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

“Заключение” содержит краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Литературе” приведен список используемых при разработке источников.

В приложениях к пояснительной записке приведен листинг программы с необходимыми комментариями.

Схема работы системы представлена в графической части.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

1 Анализ задачи

1.1 Постановка задачи

1.1.1 Организационно-экономическая сущность задачи

Темой данного проекта является мобильное приложение «Meme Cat».

В наше время все люди довольно загружены повседневной и скучной рутиной, для них каждый день проходит как день сурка. Они хотят как-то разбавить свои скучные дни каким-то занятием, которое будет интересным для них. Так же у этих людей накапливается стресс, который они хотят где-то выпустить, но не каждый знает где и как это сделать.

Мое игровое приложение как раз является выходом из этого порочного круга суеты и стресса. В ней вы сможете накормить милого и смешного котика, зарабатывая при этом игровую валюту «MeowCoin», с помощью которой вы сможете покупать улучшения, которые будут приносить больше монет и больше радовать вашего голодного и милого котёнка.

Цель проекта заключается в создании мобильного приложения «Meme Cat».

За основу программы следует взять из соответствующих источников (сайтов, статей, энциклопедий) наиболее полную, достоверную и точную информацию. При реализации проекта разработчик должен стараться сделать его простым и максимально доступным для пользователей. Приложение должно располагать своим интерфейсом и функционалом к пользователю, чтобы ему было комфортно.

Периодичность использования данного программного продукта не ограничена. Пользователь может в любое время установить, а также удалить приложение.

Аналогов для такого приложения можно привести довольно много. SPACEPLAN, Doomsday Clicker, Realm Grinder. Но эти все игры сделаны со смыслом чего-то добиться, победить босса и так далее. Но в моем приложении нету какого-то большого смысла. Оно больше для душевного умиротворения и расслабления.

1.1.2 Функциональные требования

Разрабатываемый программный продукт позволит выполнить следующие действия:

- заработок монет за нажатие;
- открытие магазина;

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		5

- покупка улучшений для заработка;
- покупка трофеев;
- открытие панели с настройками;
- выход из игры;
- открытие панели настройки звука;
- возможность регулировать музыку;
- возможность заглушить музыку.

1.1.3 Эксплуатационные требования

Требования к реализации: Для полной реализации будет использоваться язык C#, с игровым движком Unity.

Требования к надежности: alfa- и beta-тестирования перед выпуском новых обновлений.

Требования к интерфейсу: интерфейс должен быть с приятной цветовой гаммой и понятной для пользователя. Следовательно, каждое окно должно иметь ясную визуальную иерархию своих элементов. Фрагменты текста должны располагаться на экране так, чтобы пользователя было просто и понятно принимать информацию.

1.2 Диаграмма вариантов использования

Диаграмма вариантов использования – диаграмма, отражающая отношения между актерами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне.

Суть данной диаграммы состоит в следующем: проектируемая система представляется в виде множества сущностей или актеров, взаимодействующих с системой с помощью так называемых вариантов использования представлена на рисунке 1

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

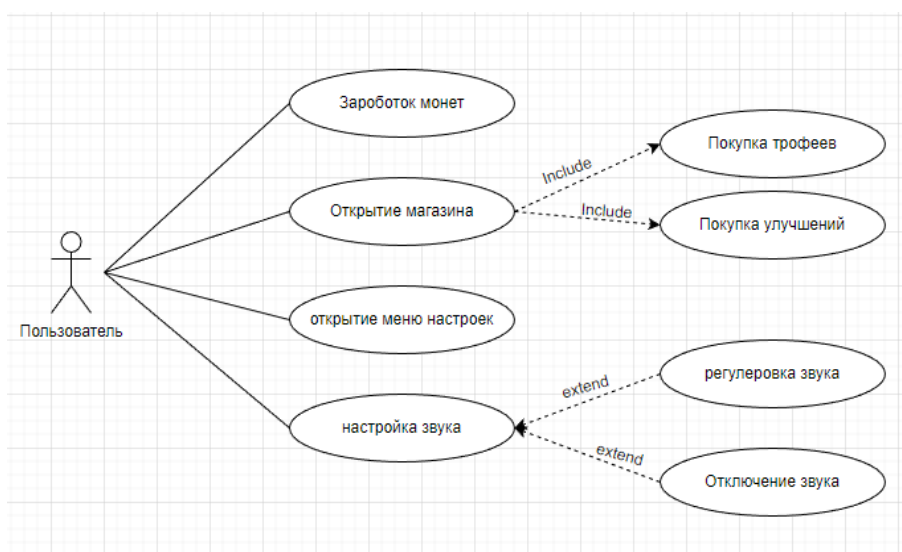


Рисунок 1 – Диаграмма вариантов использования для пользователя

1.3 Выбор стратегии разработки и модели жизненного цикла

Для разработки мобильной игры «Meme Cat» следует выбрать стратегию разработки и модель жизненного цикла. Осуществляем выбор посредством составления таблиц:

Таблица 1 – Выбор модели жизненного цикла на основе характеристик требований

№ критерия	Критерии категории требований	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	2	3	4	5	6	7	8
1.	Являются ли требования к проекту легко определяемыми и реализуемыми?	Да	Да	Да	Нет	Нет	Нет
2.	Могут ли требования быть сформулированы в начале ЖЦ?	Да	Да	Да	Да	Нет	Нет
3.	Часто ли будут изменяться требования на протяжении ЖЦ?	Нет	Нет	Нет	Нет	Да	Да
4.	Нужно ли демонстрировать требования с целью их определения?	Нет	Нет	Да	Нет	Да	Да

Продолжение таблицы 1

1	2	3	4	5	6	7	8
5.	Требуется ли проверка концепции программного средства или системы?	Нет	Нет	Да	Нет	Да	Да
6.	Будут ли требования изменяться или уточняться с ростом сложности системы (программного средства) в ЖЦ?	Нет	Нет	Нет	Да	Да	Да
7.	Нужно ли реализовать основные требования на ранних этапах разработки?	Нет	Нет	Да	Да	Да	Да

Вычисления: 2 за каскадную, за 2 V- образную, 5 за RAD, 3 за инкрементную, 5 за быстрого прототипирования и 5 за эволюционную.

Итог: На основе результатов заполнения табл. 3 подходящей является модель быстрого прототипирования и эволюционная.

Таблица 2 – Выбор модели жизненного цикла на основе характеристик команды разработчиков

№ критерия	Критерии категории команды разработчиков проекта	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1	2	3	4	5	6	7	8
1.	Являются ли проблемы предметной области проекта новыми для большинства разработчиков?	Нет	Нет	Нет	Нет	Да	Да
2.	Являются ли инструментальные средства, используемые в проекте, новыми для большинства разработчиков?	Да	Да	Нет	Нет	Нет	Да
3.	Изменяются ли роли участников проекта на протяжении ЖЦ?	Нет	Нет	Нет	Да	Да	Да
4.	Является ли структура процесса разработки более значимой для разработчиков, чем гибкость?	Да	Да	Нет	Да	Нет	Нет
5.	Важна ли легкость распределения человеческих ресурсов проекта?	Да	Да	Да	Да	Нет	Нет

Продолжение таблицы 2

1	2	3	4	5	6	7	8
6.	Приемлет ли команда разработчиков оценки, проверки, стадии разработки?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>

Вычисления: 4 за каскадную, 4 за V-образную, 1 за RAD, 4 за инкрементную, 3 за быстрого прототипирования и 4 за эволюционную.

Итог: На основе результатов заполнения табл. 4 подходящими являются каскадная, V-образная, инкрементная и эволюционная модели.

Таблица 3 – Выбор модели жизненного цикла на основе характеристик коллектива пользователей

№ критерия	Критерии категории коллектива пользователей	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1.	Будет ли присутствие пользователей ограничено в ЖЦ разработки?	<u>Да</u>	<u>Да</u>	Нет	<u>Да</u>	Нет	<u>Да</u>
2.	Будут ли пользователи оценивать текущее состояние программного продукта (системы) в процессе разработки?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
3.	Будут ли пользователи вовлечены во все фазы ЖЦ разработки?	<u>Нет</u>	<u>Нет</u>	Да	<u>Нет</u>	Да	<u>Нет</u>
4.	Будет ли заказчик отслеживать ход выполнения проекта?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да

Вычисления: 1 за каскадную, 1 за V-образную, 1 за RAD, 2 за инкрементную, 3 за быстрого прототипирования и 3 за эволюционную.

Итог: На основе результатов заполнения табл. 5 подходящей является модель быстрого прототипирования и эволюционная.

Таблица 6 – Выбор модели жизненного цикла на основе характеристик типа проектов и рисков

№ критерия	Критерии категории типов проекта и рисков	Каскадная	V-образная	RAD	Инкрементная	Быстрого прототипирования	Эволюционная
1.	Разрабатывается ли в проекте продукт нового для организации направления?	<u>Нет</u>	<u>Нет</u>	<u>Нет</u>	Да	Да	Да
2.	Будет ли проект являться расширением существующей системы?	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>	Нет	Нет
3.	Будет ли проект крупно- или среднемасштабным?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
4.	Ожидается ли длительная эксплуатация продукта?	Да	Да	<u>Нет</u>	Да	<u>Нет</u>	Да
5.	Необходим ли высокий уровень надежности продукта проекта?	<u>Нет</u>	Да	<u>Нет</u>	Да	<u>Нет</u>	Да
6.	Предполагается ли эволюция продукта проекта в течение ЖЦ?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
7.	Велика ли вероятность изменения системы (продукта) на этапе сопровождения?	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>
8.	Является ли график сжатым?	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>
9.	Предполагается ли повторное использование компонентов?	Нет	Нет	<u>Да</u>	<u>Да</u>	<u>Да</u>	<u>Да</u>
10.	Являются ли достаточными ресурсы (время, деньги, инструменты, персонал)?	Нет	Нет	Нет	Нет	<u>Да</u>	<u>Да</u>

Вычисления: 3 за каскадную, 2 за V-образную, 6 за RAD, 6 за инкрементную, 7 за быстрого прототипирования и 6 за эволюционную.

Итог: На основе результатов заполнения табл. 6 подходящей является модель быстрого прототипирования.

Общий итог: в итоге заполнения табл. 3 – 6 наиболее подходящей является модель быстрого прототипирования и эволюционная.

1.4 Инструменты разработки

Для разработки данного проекта будет выбрана среда Unity, так как это самая удобная и доступная среда разработки на данный момент. Unity – больше,

чем движок, это среда для разработки компьютерных игр, в которой объединены различные программные средства, используемые при создании ПО – текстовый редактор, компилятор, отладчик и так далее. При этом, благодаря удобству использования, Unity делает создание игр максимально простым и комфортным, а мультиплатформенность движка позволяет игроделам охватить как можно большее количество игровых платформ и операционных систем. В первую очередь, как мы уже упоминали, движок Unity дает возможность разрабатывать игры, не требуя для этого каких-то особых знаний. Здесь используется компонентно-ориентированный подход, в рамках которого разработчик создает объекты (например, главного героя) и к ним добавляет различные компоненты (например, визуальное отображение персонажа и способы управления им). Благодаря удобному Drag & Drop интерфейсу и функциональному графическому редактору движок позволяет рисовать карты и расставлять объекты в реальном времени и сразу же тестировать получившийся результат.

Второе преимущество движка – наличие огромной библиотеки ассетов и плагинов, с помощью которых можно значительно ускорить процесс разработки игры. Их можно импортировать и экспортировать, добавлять в игру целые заготовки – уровни, врагов, паттерны поведения ИИ и так далее. Никакой возни с программированием.

Третья сильная сторона Unity – поддержка огромного количества платформ, технологий, API. Созданные на движке игры можно легко портировать между ОС Windows, Linux, OS X, Android, iOS, на консоли семейств PlayStation, Xbox, Nintendo, на VR- и AR-устройства.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		11

2 Проектирование задачи

2.1 Разработка UML-диаграмм

2.1.1 Диаграмма последовательности

Диаграмма последовательности UML — такая диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления. Основные элементы диаграммы последовательности это: обозначения объектов (прямоугольники), вертикальные линии, отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами.

Диаграмма последовательности для проекта показана на рисунке 2.

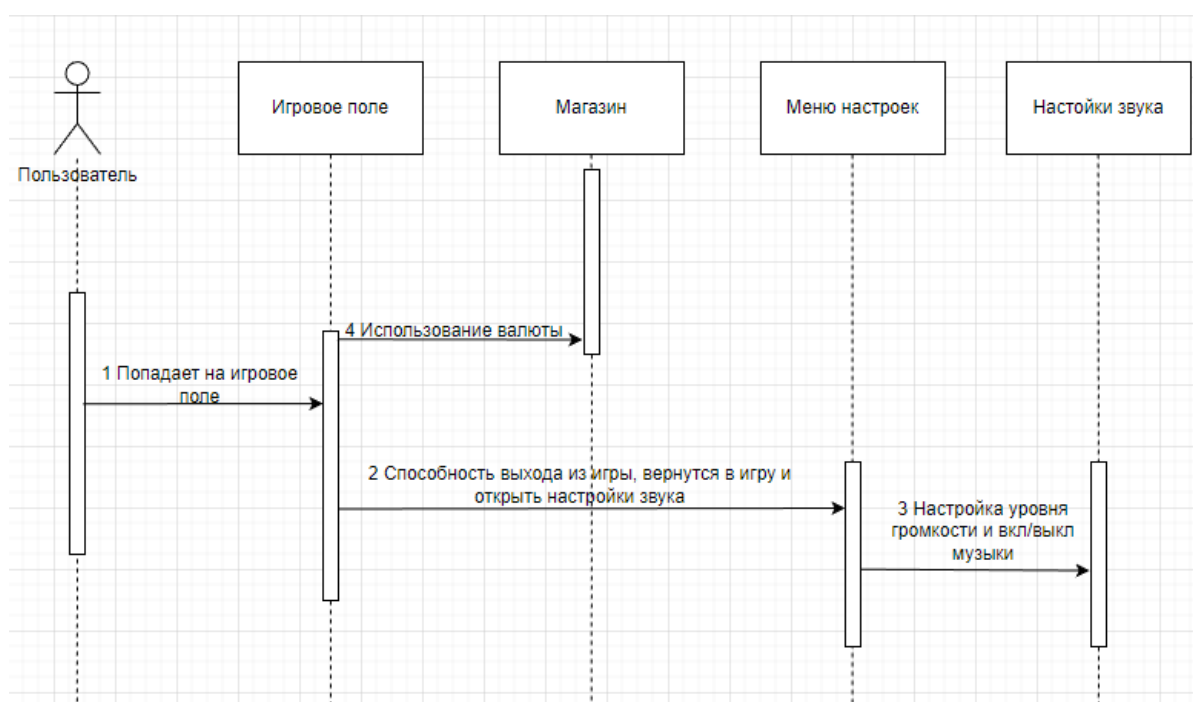


Рисунок 2 - Диаграмма последовательности

2.1.2 Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей (отношений) между ними. Широко применяется не только для документирования и визуализации, но также для конструирования посредством прямого или обратного проектирования.

Диаграмма классов для проекта показана на рисунке 3.

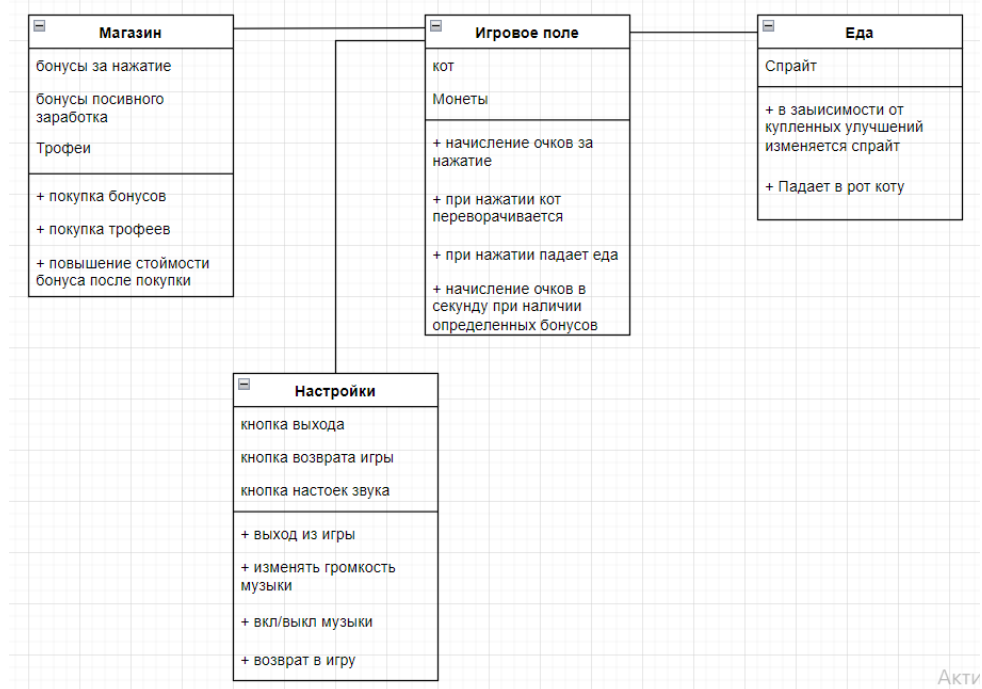


Рисунок 3 - Диаграмма классов

2.2 Разработка пользовательского интерфейса

Важным элементом проектирования данного программного продукта является описание внешнего интерфейса разрабатываемого мобильного игрового приложения.

Для разработки визуального дизайн использовались сдержанные, мягкие цвета для удобства использования программного продукта.

Для удобного использования данного приложения, все кнопки расположены так что бы не мешать игровому процессу. Также приобретенные трофеи располагаются так, чтобы не перекрывать важные аспекты игрового геймплея. Дизайн магазина и настроек выполнен в приятных и спокойных глазу тонах, для приятной игры.

С UX/UI можно ознакомиться по ссылке ниже.

<https://www.figma.com/file/BlmbiVDTm3Ye8HMsAhKREw/Untitled?type=design&node-id=0%3A1&mode=design&t=uqOqbqpsmHQxpnN5-1>

2.3 Разработка плана работы над проектом

Для разработки плана работы над проектом была создана диаграмма Ганта. Для разработки диаграммы был использован программный продукт

Microsoft Excel. Данная диаграмма дает визуальный план работ со сроками выполнения и всеми задачами. Данная диаграмма Ганта представлена на рисунке 4.

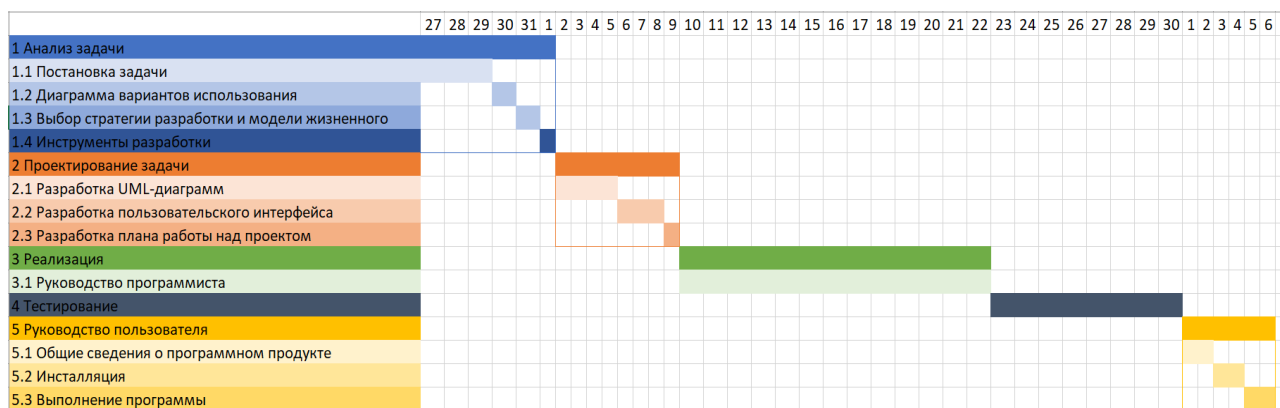


Рисунок 4 - Диаграмма Ганта

3 Реализация

3.1 Руководство программиста

Данный программный продукт был разработан с помощью игрового движка Unity с использованием языка программирования C#. Так как Unity прост в освоении для начинающих программистов, он имеет удобный и понятный интерфейс, много различных библиотек, ассетов и так далее.

3.1.1 Реализация игрового поля

Для создания игрового поля необходимо создать сцену и прикрепить к ней камеру. Далее добавить фон, кнопки магазина, настроек, кота и счет монет и добавить на них соответствующие спрайты. Написать коды для кнопок, и для зачисления монет за нажатие по экрану.

Игровое поле представлено на рисунке 5.



Рисунок 5 – Игровое поле

3.1.2 Реализация Aim панели

Для создания Aim панели необходимо создать новую панель и добавить на нее кнопку «ОК» и текст с приветствием и целью данной игры, далее написать код для кнопки и добавить на нее соответствующие спрайты.

Aim панель представлена на рисунке 6.

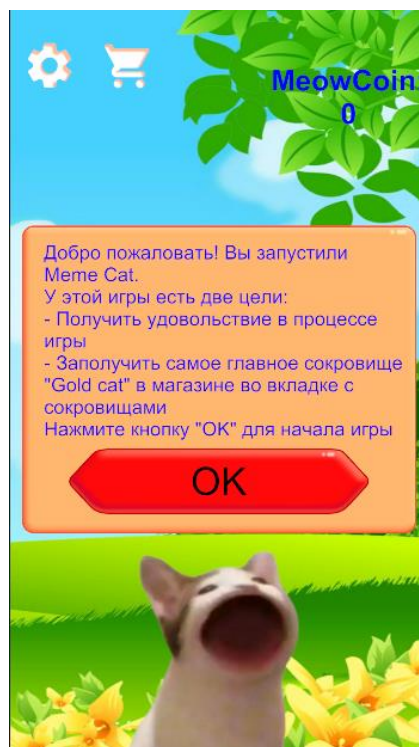


Рисунок 6 – Aim панель

3.1.3 Реализация магазина

Для создания магазина необходимо создать новую панель, на которую нужно разместить счет монет, количество получаемых монет за нажатие, количество получаемых монет за 1 сек, кнопку выхода. Также нужно создать еще две панели с тремя кнопками и ассортиментом покупаемых бонусов/трофеев и кнопки для их приобретения. Далее нужно написать коды для кнопки выхода, трех кнопок переключения между разделами магазина, для кнопки покупки бонуса/трофея. И в конце добавить на них соответствующие спрайты.

Магазин представлен на рисунке 7.



Рисунок 7 – Магазин

3.1.4 Реализация меню настроек

Для создания меню настроек необходимо создать новую панель, на которой необходимо разместить четыре кнопки: вернуться в игру, закрыть панель, настройки звука, выход из игры. Далее нужно написать код для этих кнопок и добавить на них соответствующие спрайты.

Меню настроек представлено на рисунке 8.



Рисунок 8 – Меню настроек

3.1.5 Реализация меню настроек звука

Для создания настроек звука необходимо создать новую панель и добавить на нее две кнопки: вкл/выкл музыки и закрыть меню, ползунок и два текста: On/Off. Далее написать коды к кнопкам вкл/выкл музыки и закрыть меню, и ползунку. И в конце добавить соответствующие спрайты.

Меню настроек звука представлено на рисунке 9.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

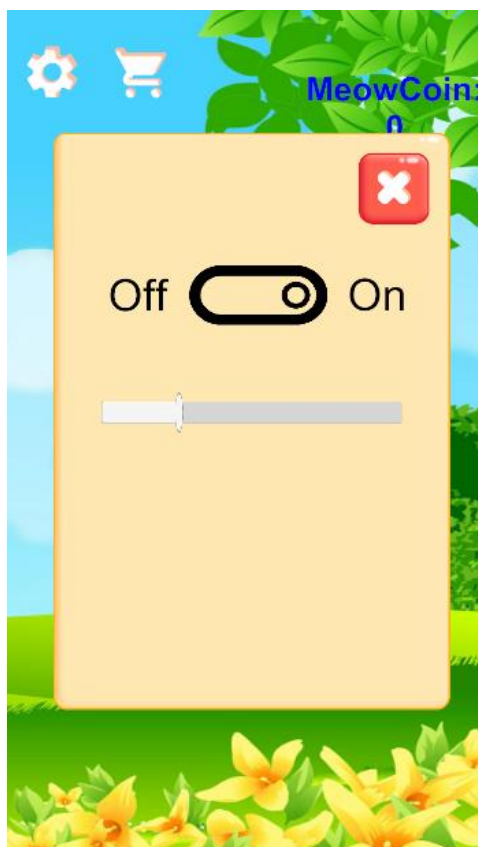


Рисунок 9 – Меню настроек звука

3.1.4 Создание Win панели

Для создания Win панели необходимо создать новую панель, и добавить на нее текст и кнопку «ОК», далее добавить код для кнопки и соответствующие спрайты.

Win панель представлено на рисунке 10.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		



Рисунок 10 – Win панель

3.1.4 Создание установщика ПП

Для того что бы создать установщик необходимо зайти: File > Build Settings... и выбрать платформу Android, потом добавить название приложения и его иконку. Далее нажимаем на кнопку Build и выбираем место куда будет создан ваш установщик.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		20

4 Тестирование

При разработке мобильного игрового приложения «Meme Cat» многие возникающие ошибки и недоработки были исправлены на этапе реализации программного продукта. После завершения испытания реализации мобильного игрового приложения было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программного продукта в автономном режиме. Тестирование ПП представлено в Таблице 1.

Таблица 1 – Отчёт о результатах тестирования функций для пользователя.

Название теста	Действие	Ожидаемый результат	Физический результат	Результат тестирования
1	2	3	4	5
Появление Aim панели	Запуск игры	Появится Win панель	Появилась Win панель	Выполнено
Заккрытие Aim панели	Нажать на кнопку «ОК»	Заккрытие Win панели	Закрылась Win панели	Выполнено
Зачисление монет и переворот кота при нажатии	Нажать на экран	Зачисление монет и переворот кота	Зачисляются монеты и кот переворачивается	Выполнено
Открытие магазина	Нажать на кнопку магазина	Открытие магазина	Открытие магазина	Выполнено
Переход на различные разделы магазина	Нажать на кнопку раздела	Переход на различные разделы магазина	Переход на различные разделы магазина	Выполнено
Покупка бонуса/трофея	Нажать на кнопку «Buy»	Приобретение бонуса/трофея	Приобретение бонуса/трофея	Выполнено
Обновление счета монет, прибавления монет за 1 нажатие и прибавление монет за 1 сек	Находится в игре	Обновление счета монет, прибавления монет за 1 нажатие и прибавление монет за 1 сек	Обновление счета монет, прибавления монет за 1 нажатие и прибавление монет за 1 сек	Выполнено
Заккрытие магазина	Нажать на кнопку закрытия	Заккрытие магазина	Заккрытие магазина	Выполнено
Открытие меню настроек	Нажать на кнопку настроек	Открытие меню настроек	Открытие меню настроек	Выполнено
Возврат в игру	Нажать на кнопку «Play»	Возврат в игру	Возврат в игру	Выполнено
Заккрытие меню настроек	Нажать на кнопку закрытия меню настроек	Заккрытие меню настроек	Заккрытие меню настроек	Выполнено

Продолжение таблицы 1

1	2	3	4	5
Выход из игры	Нажать на кнопку «Exit»	Выход из игры	Выход из игры	Выполнено
Открытие панели настройки звука	Нажать на кнопку «Volume»	Открытие панели настройки звука	Открытие панели настройки звука	Выполнено
Вкл/выкл музыки	Нажать на кнопку Вкл/выкл музыки	Вкл/выкл музыки	Вкл/выкл музыки	Выполнено
Изменение громкости музыки	Передвинуть ползунок	Изменение громкости музыки	Изменение громкости музыки	Выполнено
Воспроизведение музыки	Зайти в игру	Воспроизведение музыки	Воспроизведение музыки	Выполнено
Открытие Win панели	Приобрести сокровище «Gold Cat»	Открытие Win панели	Открытие Win панели	Выполнено
Закрытие Win панели	Нажать на кнопку «OK»	Закрытие Win панели	Закрытие Win панели	Выполнено

При тестировании программного продукта, в первую очередь нужно обратить внимание на правильную работу всего ПП. Показываются ли все изображения, виден ли текст, нет ли лишних пробелов и больших отступов, правильно ли производится подсчет монет и их зачисление, играет ли музыка и зациклена ли она и так далее.

В результате проведения тестирования выяснилось, что все ранее оговоренные функции и требования, были разработаны, а также протестированы. Тесты показали, что все функции работают правильно. Следовательно разработанный ПП можно опубликовать.

В ходе тестирования программного обеспечения продукта на разных устройствах не было выявлено каких-либо ошибок, так как адаптивность ПП была проведена на всех стадиях разработки.

5 Руководство пользователя

5.1 Общие сведения о программном продукте

Разрабатываемое игровое мобильное приложение будет носить название «Meme Cat».

Данный программный продукт является мобильной игрой, предназначенной для веселого проведения времени. От предназначен для лиц всех возрастов.

Данный ПП занимает не много памяти на диске и является весьма простой программой, так что даже слабый телефон сможет установить и запустить его.

5.2 Инсталляция

Инсталляция проходит очень быстро и просто. Все что нужно сделать это открыть арк файл и на всплывающем окне нажать сначала кнопку «Установить», подождать установки, а затем нажать на кнопку «Открыть». Окно установки показано на рисунках 11, 12.



Рисунок 11 - Окно установки

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		23



Рисунок 12 - Окно установки

5.3 Выполнение программы

Запустить игру можно двумя способами:

- 1) При установке игры после ее загрузки нажать открыть;
- 2) Нажать на иконку игры на главном экране вашего устройства.

После запуска игры вас встречает панель на которой написана цель данной игры. Панель представлена на рисунке 13

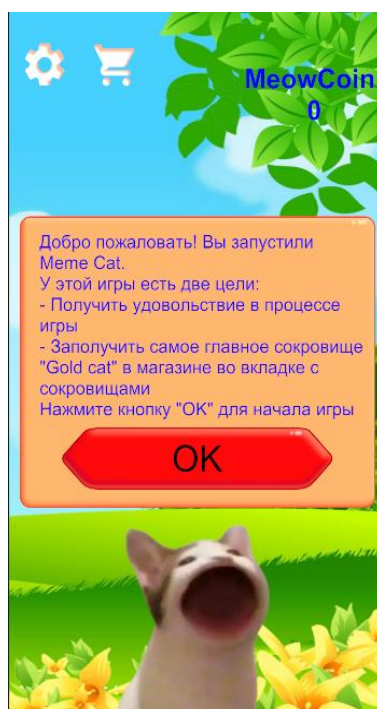


Рисунок 13 – Aim панель

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		24

Далее после нажатия кнопки «ОК» панель закрывается. На главном экране мы видим кота, счетчик монет, кнопки магазина и настроек. Главный экран представлен на рисунке 14.



Рисунок 14 – Главный экран

Нажимая на пустую область экрана, появляется еда, которая падает коту в рот, начисляется монеты и кот переворачивается. Нажав на кнопку настроек высветится панель настроек изображенная на рисунке 15

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		



Рисунок 15 – Меню настроек

В нем вы можете нажать на четыре кнопки: Кнопка с крестиком закрывает панель, кнопка «Play» возвращает нас на главный экран, кнопка «Exit» закрывает игру, кнопка «Volume» открывает панель настройки музыки, представленную на рисунке 16.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

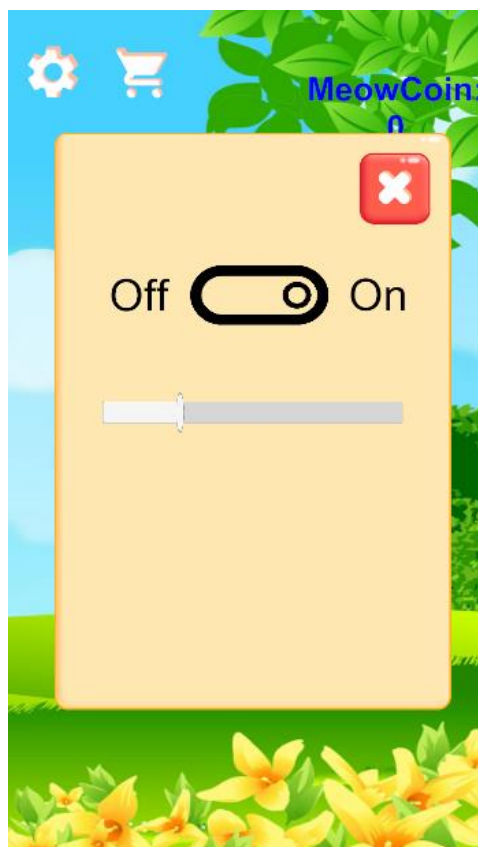


Рисунок 16 – Меню настроек звука

В этом меню вы можете выключить а также включить музыку, а также регулировать ее громкость перетягивая ползунок громкости, также можно закрыть эту панель, нажав на кнопку закрытия в правом верхнем углу панели.

Далее на главном экране нажав на кнопку с рисунком корзинки вы можете зайти в магазин, представленный на рисунке 17.



Рисунок 17 – Магазин

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

В магазине вы можете посмотреть сколько у вас монет, также как и не главном экране, сколько вы получаете монет за нажатие и сколько монет зачисляется в секунду. Также здесь есть три кнопки, с помощью которых можно открыть тот или иной раздел:

1) Кнопка с рисунком синего нажимающего пальца открывает раздел с улучшениями получением монет за нажатие;

2) Кнопка с рисунком завода открывает раздел с улучшениями заработка монет в секунду;

3) Кнопка с золотым котом открывает раздел с бонусами, покупая которые, появляются на фоне на главном экране.

Также можно закрыть магазин, нажав на кнопку закрытия в правом верхнем углу экрана.

Что бы пройти игру, нужно купить последний трофей, но это не так просто. Во-первых, он стоит 100 млн монет, а во-вторых, чтобы его купить, необходимо купить предыдущие трофеи. После покупки последнего трофея «Gold Cat» появится Win панель в которой будет написана за какое время была пройдена игра, а также общее количество заработанных монет за всю игру. Панель представлена на рисунке 18.



Рисунок 18 – Win панель

После нажатия на кнопку «ОК», в главном меню поменяется спрайт еды на золотую рыбу, а также спрайт кота на золотого кота, это будет означать пройденную игру.

Для выхода из игры в настройках необходимо нажать на кнопку «Exit»

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

Целью данного проекта на практику являлась разработка мобильного игрового приложения с названием «Meme Cat».

Нужно заметить, что в программном продукте был реализован простой и удобный интерфейс, который позволяет использовать данное мобильное приложение тому пользователю, который не обладает отличными навыками работы с программными продуктами данного типа. При раз работке были выполнены требования, поставленные в 1 и 2 разделе документации.

Данное приложение оптимизированно под большинство Android устройств. Благодаря удобному и интуитивно понятному интерфейсу в игру могут играть пользователи всех возрастов. Приложение, как и задумывалось, является интересным и расслабляющим способом провести свободное время.

После долгого и тщательного тестирования были выявлены небольшие ошибки и недоработки, которые в последствии были исправлены на стадии тестирования.

В целом при реализации программного продукта, были выполнены все условия, поставленные на начальном этапе разработки. Из чего мы можем сделать вывод, что данное мобильное приложение можно вводить в эксплуатацию.

Так же в процессе создания программного продукта была подготовлена программная документация. Я научился разрабатывать диаграмму Ганта и тесты на использование в процессе тестирования.

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

Список использованных источников

1. Курс обучения C# [Электронный ресурс]. – Режим доступа: https://www.youtube.com/watch?v=_8yZYhAkQjQ&list=PLDyJYA6aTY1laYPs6iS-SrYI9DZLVCUKr&index=1&ab_channel=%D0%A8%D0%BA%D0%BE%D0%BB%D0%B0itProger%2F%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5
2. Курс обучения Unity [Электронный ресурс]. – Режим доступа: https://www.youtube.com/watch?v=xAtou-FnC7o&list=PLDyJYA6aTY1k_-3fFiMVoYY04jCr-QY55&ab_channel=%D0%A8%D0%BA%D0%BE%D0%BB%D0%B0itProger%2F%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А
Листинг программы

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

```

using Cats.Core;
using System.Collections;
using System.Collections.Generic;
using Unity.VisualScripting;
using UnityEngine;
using UnityEngine.PlayerLoop;
using UnityEngine.UI;

public class Music : MonoBehaviour
{
    public GameDataContainer
dataContainer;
    private void Update() //обновление
музыки при ее завершении
    {
        dataContainer.fonMusic.volume
= dataContainer.slider.value;
        if( dataContainer.musicTime ==
136)
        {

dataContainer.fonMusic.PlayOneShot(dat
aContainer.clip);
            dataContainer.musicTime =
0;
        }
    }
    public void OfOnAudio() //функция
для кнопки вкл/выкл музыки
    {
        if(AudioListener.volume == 1)
        {
            AudioListener.volume = 0;

dataContainer.offOnButton.GetComponent
<Image>().sprite =
dataContainer.OffOn[1];
        }
        else
        {
            AudioListener.volume = 1;

dataContainer.offOnButton.GetComponent
<Image>().sprite =
dataContainer.OffOn[0];
        }
    }
    public void Start() //запуск
музыки
    {
        dataContainer.fonMusic.PlayOneShot(dat
aContainer.clip);
        StartCoroutine(MusicTime());
    }
    IEnumerator MusicTime()
    {
        yield return new
WaitForSeconds(1);
        dataContainer.musicTime++;
        StartCoroutine(MusicTime());
    }
}
using Cats.Core;

```

```

using System;
using System.Collections;
using System.Collections.Generic;
using Unity.VisualScripting;
using UnityEngine;
namespace Cats.Mechanics
{
    public sealed class ShopController
: MonoBehaviour
    {
        public GameDataContainer
dataContainer;
        public void BuyCatShop(int id)
//функция для кнопок покупки улучшения
монет за нажатие
        {
            if (dataContainer.MeowCoin
<= 0) { dataContainer.MeowCoin = 0;
return; }

            if (dataContainer.MeowCoin
>= dataContainer.costs[id] &&
dataContainer.MeowCoin != 0)
            {
                dataContainer.MeowCoin
-= Math.Round(dataContainer.costs[id],
2);
                dataContainer.MeowCoin
= Math.Round(dataContainer.MeowCoin,
2);

dataContainer.costs[id] *=
Math.Round(GameDataContainer.MYLTIPPLY_
VALUE, 2);

dataContainer.costs[id] =
Math.Round(dataContainer.costs[id],
2);

dataContainer.clickPower +=
dataContainer.clickPowers[id];
                if (id >
dataContainer.maxBuy)
                {

dataContainer.mainFood =
dataContainer.foods[id];

dataContainer.maxBuy = id;
                }
            }
        }
        public void
BuyCatShopSpeed(int id) //функция для
кнопок покупки улучшения монет за
секунду
        {
            if (dataContainer.MeowCoin
<= 0) { dataContainer.MeowCoin = 0;
return; }
            if (dataContainer.MeowCoin
>= dataContainer.costsSpeed[id] &&
dataContainer.MeowCoin != 0)

```

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		33

```

        {
            dataContainer.MeowCoin
            -=
            Math.Round(dataContainer.costsSpeed[id
            ],2);
            dataContainer.MeowCoin
            = Math.Round(dataContainer.MeowCoin,
            2);

            dataContainer.costsSpeed[id] *=
            Math.Round(GameDataContainer.MYLTIPPLY_
            VALUE,2);

            dataContainer.costsSpeed[id] =
            Math.Round(dataContainer.costsSpeed[id
            ]);

            dataContainer.clickSpeed +=
            dataContainer.clickSpeeds[id];
        }

    }
    public void
    BuyCatShopTreasure(int id) //функция
    для кнопок покупки сокровищ
    {
        if (dataContainer.MeowCoin
        <= 0) { dataContainer.MeowCoin = 0;
        return; }
        if (dataContainer.MeowCoin
        >= dataContainer.costsTreasure[id] &&
        dataContainer.MeowCoin != 0)
        {
            dataContainer.MeowCoin
            -=
            Math.Round(dataContainer.costsTreasure
            [id], 2);
            dataContainer.MeowCoin
            = Math.Round(dataContainer.MeowCoin,
            2);

            if (id == 9)
            {

            dataContainer.catSprite.gameObject.Set
            Active(false);

            dataContainer.clickPower += 99999;

            dataContainer.clickSpeed += 1000000;
            dataContainer.win
            = true;

            dataContainer.winPanel.SetActive(true)
            ;

            if
            (dataContainer.win == true)
            {

            dataContainer.endMeowCoin =
            dataContainer.allMeowCoin;
            }
        }
    }

```

```

        else
        {

            dataContainer.treasures[id].gameObject
            .SetActive(true);
        }

            dataContainer.treasureButtons[id].game
            Object.SetActive(false);

            dataContainer.allTreasure++;

            if
            (dataContainer.allTreasure == 9)
            {

            dataContainer.treasureButtons[9].gameO
            bject.SetActive(true);
            }
        }

    }
    private void Start()
    {

        StartCoroutine(MoneyFarm());

        StartCoroutine(PlayTime());

        dataContainer.treasureButtons[9].gameO
        bject.SetActive(false);

        dataContainer.GoldcatSprite.gameObject
        .SetActive(false);
    }
    IEnumerator
    MoneyFarm()//енумератор для для
    добавления монет в секунду
    {
        yield return new
        WaitForSeconds(1);
        dataContainer.allMeowCoin
        += dataContainer.clickSpeed;
        dataContainer.MeowCoin +=
        dataContainer.clickSpeed;

        StartCoroutine(MoneyFarm());
    }
    IEnumerator
    PlayTime()//енумератор для подсчета
    секунд геймплея
    {
        yield return new
        WaitForSeconds(1);
        if(dataContainer.min ==
        60)
        {
            dataContainer.min = 0;
            dataContainer.sec++;
            dataContainer.hour++;
        }
        else if (dataContainer.sec
        == 59)
    }

```

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		34


```

        {
            dataContainer.sec = 0;
            dataContainer.min++;
        }
        else
        {
            dataContainer.sec++;
        }
        if (dataContainer.win ==
true)
        {
            StopCoroutine(PlayTime());
        }
        else
        {
            StartCoroutine(PlayTime());
        }
    }
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

namespace Cats.Core
{
    public sealed class
GameDataContainer : MonoBehaviour
//все переменные проекта
    {
        public const double
MYLTIPLY_VALUE = 1.07;

        [Header("Sprites")]
        public SpriteRenderer
catSprite;
        public SpriteRenderer
GoldcatSprite;
        public SpriteRenderer[] foods;
        public SpriteRenderer
mainFood;
        public Image[] treasures;
        public Sprite[] OffOn;

        [Header("GameObjects")]
        public GameObject fish;
        public Transform spawnPoint;
        public Slider slider;
        public GameObject winPanel;

        [Header("Buttons")]
        public Button[]
treasureButtons;
        public GameObject offOnButton;
        [Header("Score")]
        public double MeowCoin = 0;
        public Text[] MeowCoinText;
        public double clickPower = 1;
        public Text clickPowerText;
        public double clickSpeed = 0;

        public Text clickSpeedText;
        public int allTreasure = 0;
        public int sec = 0;
        public int min = 0;
        public int hour = 0;
        public double allMeowCoin = 0;
        public double endMeowCoin = 0;
        public int musicTime=0;

        [Header("ShopController")]
        public double[] costs;
        public double[] costsSpeed;
        public double[] costsTreasure;
        public double[] clickPowers;
        public double[] clickSpeeds;
        public int maxBuy = -1;
        public bool win = false;
        [Space]
        public Text[] costTexts;
        public Text[]
costTreasureText;
        public Text[] costTextsSpeed;
        public Text[] clickPowerTexts;
        public Text[] clickSpeedTexts;
        public Text allScoreText;
        public Text allTimeText;
        public Text winText;
        [Header("Audio")]
        public AudioSource fonMusic;
        public AudioClip clip;
    }
}

using System.Collections;
using System.Collections.Generic;
using UnityEngine;

namespace Cat.Helpers
{
    public struct Tag // доп структура
для использования тегов
    {
        public const string Untagged
= "Untagged";
        public const string Respawn
= "Respawn";
        public const string Finish
= "Finish";
        public const string EditorOnly
= "EditorOnly";
        public const string MainCamera
= "MainCamera";
        public const string Player
= "Player";
        public const string
GameController = "GameController";
        public const string Food =
"Food";
    }
}

using Cats.Core;
using System.Collections;
using System.Collections.Generic;

```

					УП ТРПО 2-40 01 01.33.37.14.23 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		35

```

using UnityEngine;

namespace Cats.Mechanics
{
    public sealed class ClickHandler :
    MonoBehaviour
    {
        [SerializeField] private
        GameDataContainer dataContainer;

        private void Awake()
        {
            if(dataContainer == null)

                dataContainer =
                GameObject.FindObjectOfType<GameDataCo
                nteiner>();
        }
        public void OnClick()
        //добавление монет при нажатии,
        //переворот кота и создание еды
        {
            dataContainer.MeowCoin +=
            dataContainer.clickPower;
            dataContainer.allMeowCoin
            += dataContainer.clickPower;

            dataContainer.catSprite.flipX
            = !dataContainer.catSprite.flipX;

            dataContainer.GoldcatSprite.flipX =
            !dataContainer.catSprite.flipX;

            if(dataContainer.allTreasure == 10)
            {
                dataContainer.mainFood
                = dataContainer.foods[10];
            }

            Instantiate(dataContainer.mainFood,
            dataContainer.spawnPoint.position,
            Quaternion.identity);
        }
    }

    using Cats.Core;
    using System.Collections;
    using System.Collections.Generic;
    using Unity.VisualScripting;
    using UnityEngine;
    namespace Cats.UI
    {
        public sealed class ButtonHandler
        : MonoBehaviour
        {
            public GameDataContainer
            dataContainer;
            public void
            OpenAndCloseUIPanel(GameObject panel)
            //функция для кнопок открытия и
            //закрытия панелей

```

```

        {
            panel.SetActive(!panel.activeInHierarc
            hy);
            if
            (dataContainer.allTreasure < 10)
            {
                dataContainer.catSprite.gameObject.Set
                Active(!dataContainer.catSprite.gameOb
                ject.active);
            }
            else
            {
                dataContainer.GoldcatSprite.gameObject
                .SetActive(!dataContainer.GoldcatSprit
                e.gameObject.active);
            }
        }
        public void
        SwitchShopLists1(GameObject
        panel)//функция для отображения панели
        {
            panel.SetActive(true);
        }
        public void
        SwitchShopLists2(GameObject
        panel)//функция для скрытия панели
        {
            panel.SetActive(false);
        }
        public void
        ExitButton()//функция для кнопки
        выхода из игры
        {
            Application.Quit();
        }
    }
}

using Cat.Helpers;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

namespace Cats.Helpers
{
    public sealed class
    CleansingTheWorld : MonoBehaviour
    {
        public string tagObjDestroy;
        private void Start()//проверка
        теров
        {
            if(tagObjDestroy ==
            string.Empty || tagObjDestroy == null)
            {
                tagObjDestroy =
                Tag.Food;
            }
        }
    }
}

```

```

        private void
OnCollisionEnter2D(Collision2D
collision)//если тег = Food, то этот
компонент уничтожается

{
    if
(collision.gameObject.CompareTag(tagOb
jDestroy))
    {
        Destroy(collision.gameObject);
    }
}
using Cats.Core;
using System;
using UnityEngine;
using
UnityEngine.SocialPlatforms.Impl;
using UnityEngine.UI;
namespace Cats.Mechanics
{
    public sealed class UIHandler :
MonoBehaviour
    {
        [SerializeField] private
GameDataContainer dataContainer;

        private void Awake()
        {
            if(dataContainer == null)
                dataContainer =
GameObject.FindObjectOfType<GameDataCo
ntainer>();
        }
        private void
Update()//обновление текстов
        {
            UpdateScoreTexsts();
            UpdateCostsTextCatShop();

            UpdateClickPowerTextsCatShop();
        }

        private void
UpdateClickPowerTextsCatShop()//функци
я обновления текста в магазине с
бонусами
        {
            for (int i = 0; i <
dataContainer.clickPowerTexts.Length;
i++)
            {
                dataContainer.clickPowerTexts[i].text
= $"Click +
{dataContainer.clickPowers[i]}";

                dataContainer.clickSpeedTexts[i].text
= "+" + dataContainer.clickSpeeds[i] +
" Coin/sec";
            }
        }
    }
}

```

```

        private void
UpdateCostsTextCatShop() //функция
обновления текста в магазине с
бонусами
        {
            for (int i = 0; i <
dataContainer.costTexts.Length; i++)
            {
                dataContainer.costTexts[i].text =
$"Cost: {dataContainer.costs[i]}";

                dataContainer.costTextsSpeed[i].text =
$"Cost:
{dataContainer.costsSpeed[i]}";

                dataContainer.costTreasureText[i].text
= $"Cost:
{dataContainer.costsTreasure[i]}";
            }
        }

        private void
UpdateScoreTexsts() //функция
обновления текста в магазине с
бонусами
        {
            for (var index = 0; index
< dataContainer.MeowCoinText.Length;
index++)
            {
                dataContainer.MeowCoinText[index].text
= $"MeowCoin:
{dataContainer.MeowCoin}";

                dataContainer.clickPowerText.text =
$"Click : {dataContainer.clickPower}";

                dataContainer.clickSpeedText.text =
dataContainer.clickSpeed + "
Coin/sec";

                dataContainer.allTimeText.text =
$"Play time: { dataContainer.hour}hour
{dataContainer.min}min
{dataContainer.sec}sec";

                dataContainer.allScoreText.text =
$"Score: {dataContainer.endMeowCoin}";
            }
        }
    }
}

```