МИНОБРНАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Институт архитектуры и строительства

Факультет архитектуры и градостроительного развития

Направление (специальность) 09.03.02 Информационные системы и технологии  
Кафедра цифровых технологий в урбанистике, архитектуре и строительстве

Дисциплина Качество и надежность ИС

|  |  |
| --- | --- |
|  | Утверждаю  зав. кафедрой Д.С. Парыгин |
|  | «\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20 \_\_\_ г. |

**ЗАДАНИЕ**

**на курсовую работу**

Групповая работа студентов: Алексеев А.Л., Савушкин А.Д., Шилин А.И., Кирюхин М.В.

Группа ИСТ-1-22

1. Тема: «Разработка информационной системы для заданной предметной области с учетом требований к качеству. Игра Stress Out.»

2. Срок представления работы к защите « » 20 г.

3. Содержание расчетно-пояснительной записки: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

4. Перечень графического материала: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

5. Дата выдачи задания « » 20 г.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рашевский Н.М.

подпись, дата

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексеев А.Л.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Савушкин А.Д.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шилин А.И.

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кирюхин М.В.

МИНОБРНАУКИ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Институт архитектуры и строительства

Факультет архитектуры и градостроительного развития

Кафедра цифровых технологий в урбанистике, архитектуре и строительстве

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе**

по дисциплине Качество и надежность ИС

на тему «Разработка информационной системы для заданной предметной области с учетом требований к качеству. Игра Stress Out.»

Студенты Алексеев Андрей Леонидович, Савушкин Александр Дмитриевич, Кирюхин Матвей Валерьевич, Шилин Арсений Игоревич.

(фамилия, имя, отчество)

Группа ИСТ-1-22

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

Члены комиссии:

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

(подпись и дата подписания) (инициалы и фамилия)

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

(подпись и дата подписания) (инициалы и фамилия)

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

(подпись и дата подписания) (инициалы и фамилия)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Рашевский Н.М.

(подпись, дата подписания) (инициалы и фамилия)

Волгоград 2023 г.

Содержание

[1. Исследование предметной области. 4](#_heading=h.30j0zll)

[2. Использование системы контроля версий 5](#_heading=h.1fob9te)

[3. Тестирование информационной системы 9](#_heading=h.3znysh7)

[4. Кодирование. 14](#_heading=h.2et92p0)

[5. Результат работы программы 17](#_heading=h.tyjcwt)

# ***1. Исследование предметной области.***

Наша программа является игрой для снижения уровня стресса и развлечения. Главной ее целью является помочь пользователю расслабиться и успокоить свои нервы.

На данном этапе разработки были разработаны следующие функции:

* Передвижение и телепортация игрока
* Подбор предметов
* Взаимодействие с окружением
* Частичное или полное разрушение окружения
* Разрушение различными способами
* Создание розочки
* РАЗРЕЗАНИЕ ОЛЕНЯ НА КУСКИ!

Также наша программа имеет следующие особенности:

* Интерактивность с окружающей средой
* РАЗРЕЗАНИЕ ОЛЕНЯ НА КУСКИ!
* Красивая визуальная составляющая

Работа над нашей программой совершается группой из четырех человек.

Задачи равномерно распределены между членами команды, что помогает добиться продуктивности и улучшения качества продукта.

Алексеев Андрей Леонидович – тим-лид, разработчик

Шилин Арсений Игоревич – тех-лид, разработчик, работа с документами

Кирюхин Матвей Валерьевич – 3D Modeler, разработчик

Савушкин Александр Дмитриевич– 3D Modeler, Map Designer, разработчик

В ходе работы наша команда столкнулась с необходимостью создания

макета программы. Данная проблема была решена успешно.

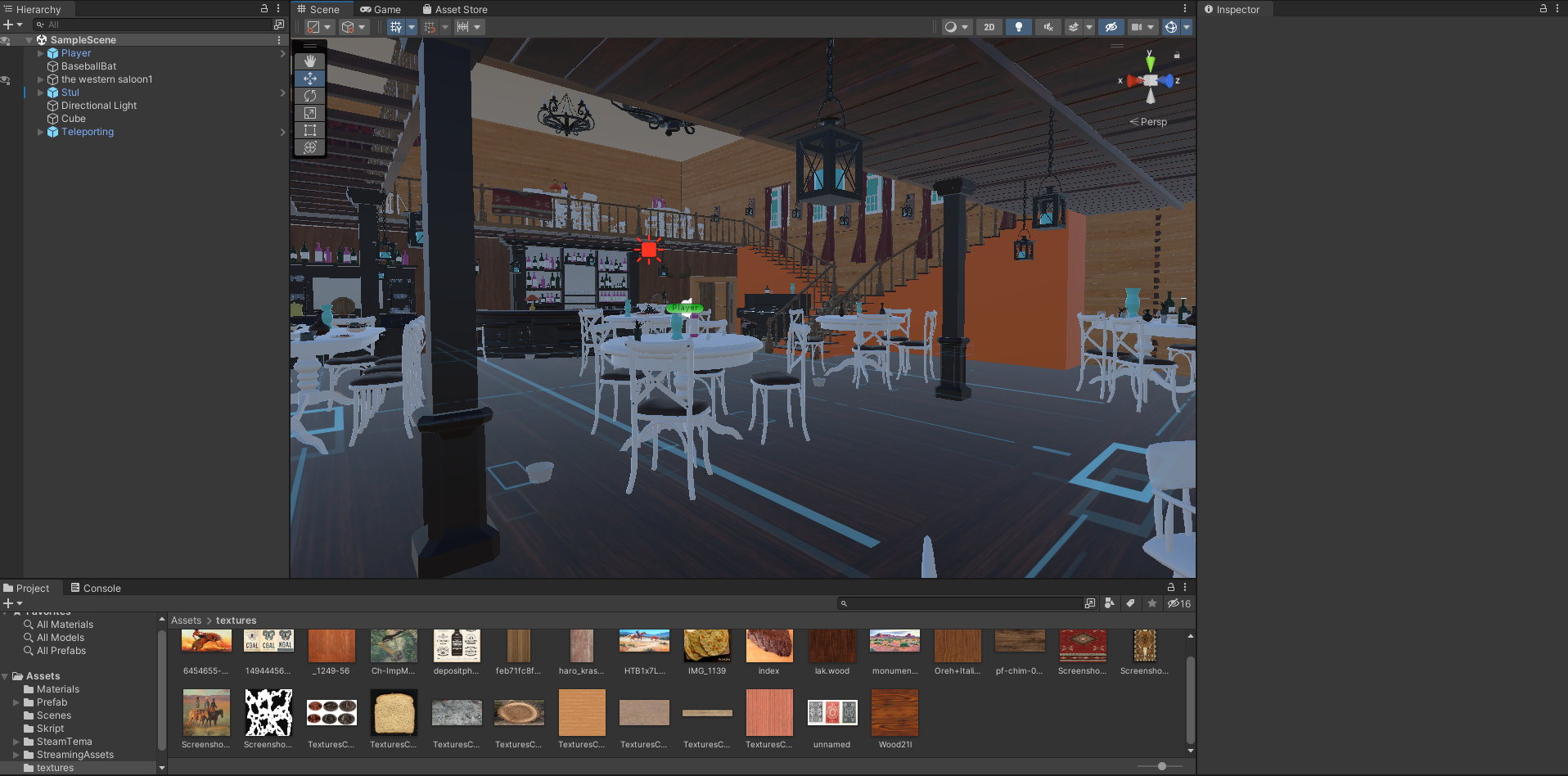


Рис.1. Макет программы.

Предложенный нами интерфейс удовлетворяет такому фактору качества

разработки ИС, как удобство использования, так как разработанная нами программа является интуитивно понятной и простой как для целевой аудитории, так и для других пользователей. Также важно отметить, что интерфейс нашего проекта не перегружен лишними объектами.

# ***2. Использование системы контроля версий***

Мы решили использовать распределенную систему контроля версий, и для хранения нашего репозитория мы выбрали веб-платформу GitHub. Этот выбор обоснован тем, что GitHub предоставляет удобный интерфейс, обширный функционал и удобную Desktop-версию, что делает его привлекательным решением для нашей команды.

Наша работа с GitHub началась с создания репозитория.



Рис.2 Название репозитория.

После чего тим-лид добавил всех разработчиков и загрузил наш проект на репозиторий.

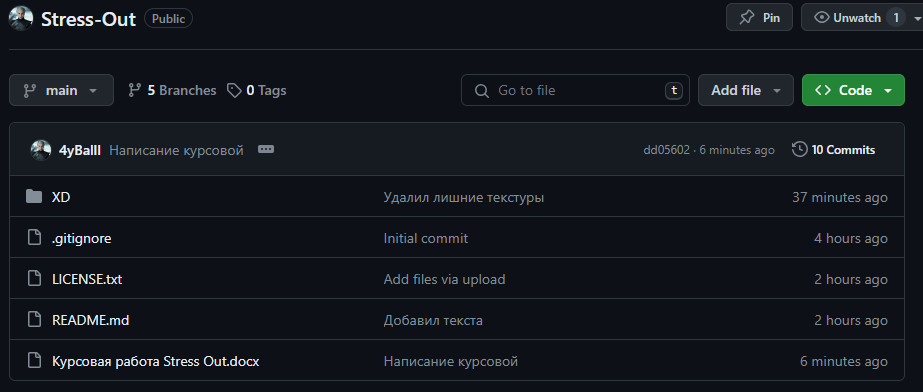


Рис.3 Содержимое репозитория

После участники создали себе ветви, чтобы вносить свои изменения и экспериментировать с кодом, не затрагивая основную ветку.

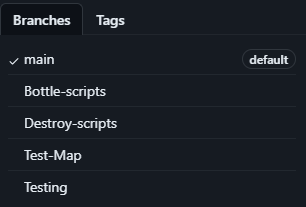


Рис.4 Пример веток проекта.

Main - главная ветка репозитория

Bottle-scripts - ветка для тестирования скрипта разрушения бутылок

Destroy-scripts - ветка для тестирования скрипта частичного разрушения объектов

Test-Map - Ветка для тестирование окружающей локации

Testing - ветка для экспериментов

Каждый из участников создал для себя собственную ветку, в которой вел работу над проектом. Данное решение обусловлено тем, что использование отдельной ветки позволяет разработчику делать свою работу и избежать прекращения работы всего проекта в случае возникновения ошибок или конфликтов.

Изменения в ветках происходили путем коммита в них изменений

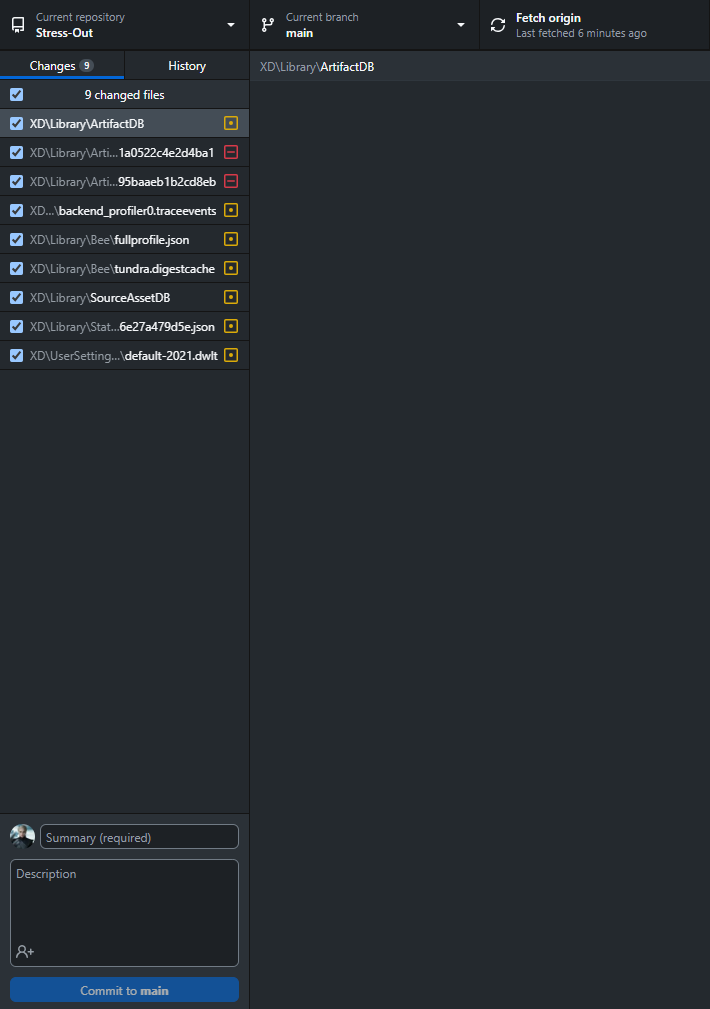


Рис.5 Пример создания коммита и выбора ветки.

После создания коммита, его необходимо выгрузить. Если все сделано правильно, то изменения в ветке сохранятся и это будет видно на GitHub.

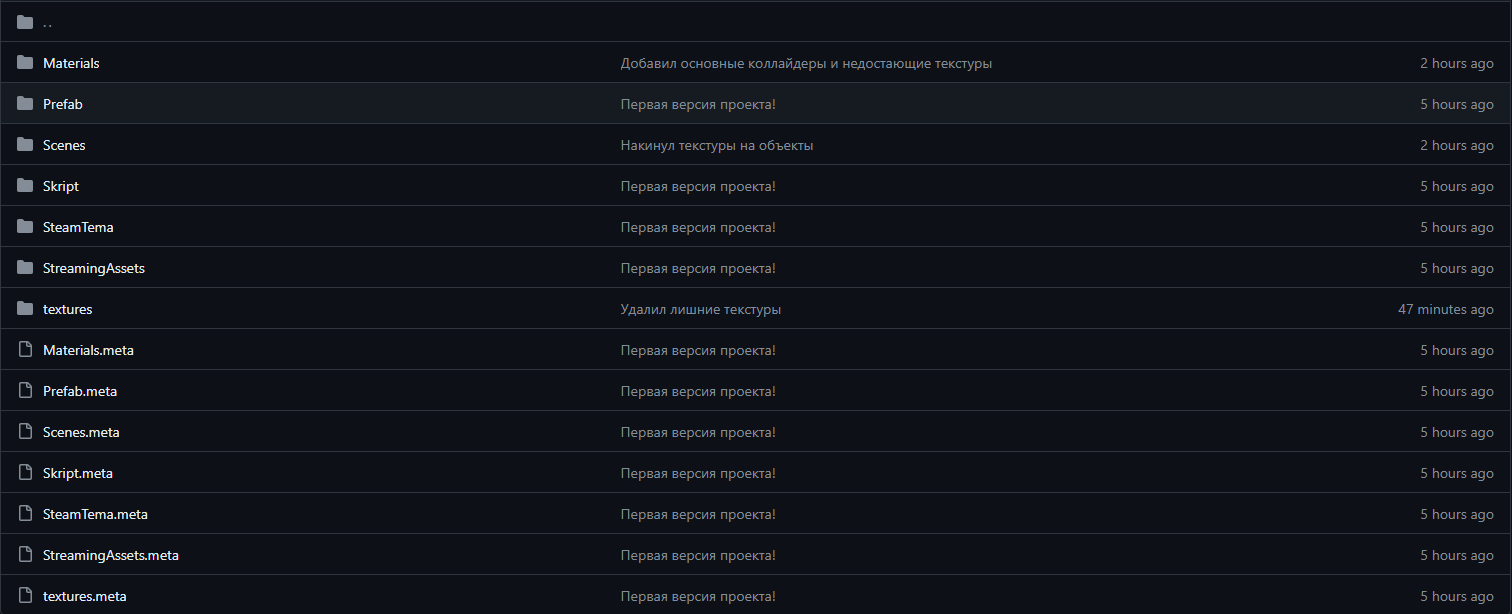


Рис.6 Пример примененного к ветке коммита.

Web-версия GitHub даёт понять другим членам команды, когда и где, были произведены изменения какой-либо ветки.

Изменения каждой ветки за время создания проекта образует дерево ведения проекта.

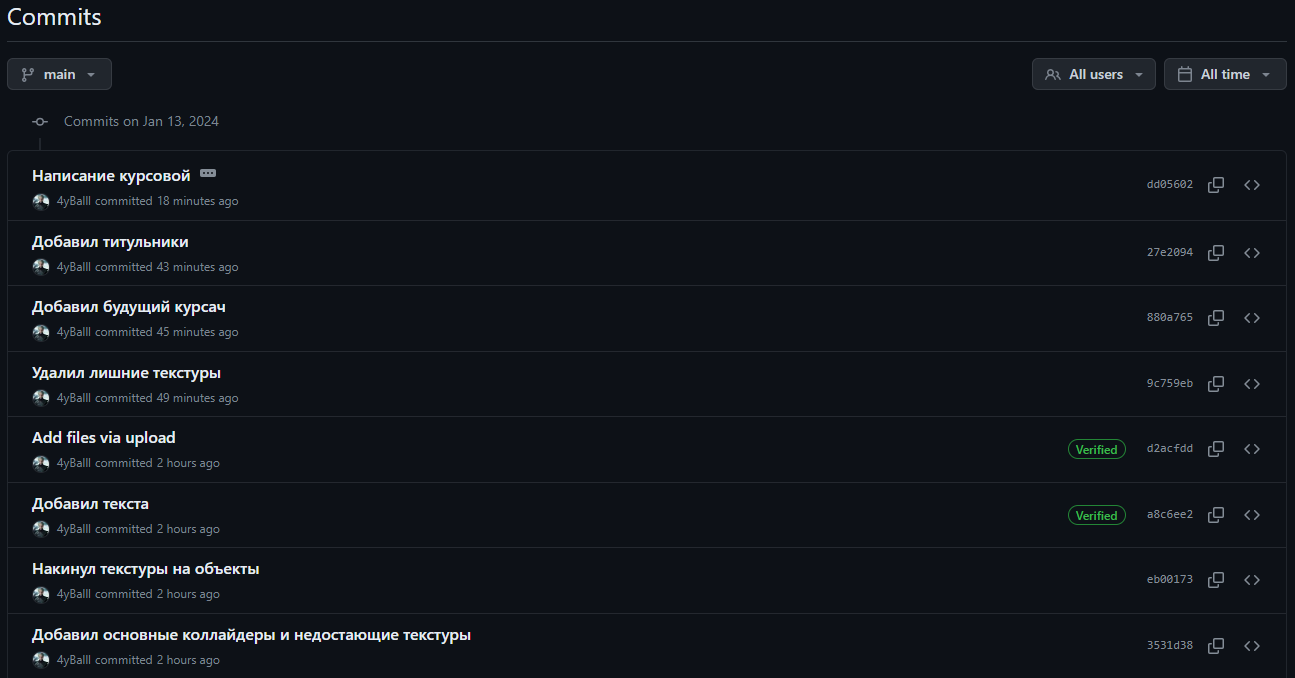


Рис.7 Дерево ведения проекта.

В дереве ведения проекта показаны коммиты главной ветки репозитория и дата их введения.

# ***3. Тестирование информационной системы***

Тестирование программы является неотъемлемой частью разработки продукта, так как нужно учитывать то, что не все пользователи могут корректно пользоваться приложением и с целью избегания прекращения работы, необходимо сделать так, чтобы программный код корректно отрабатывал во всех случаях.

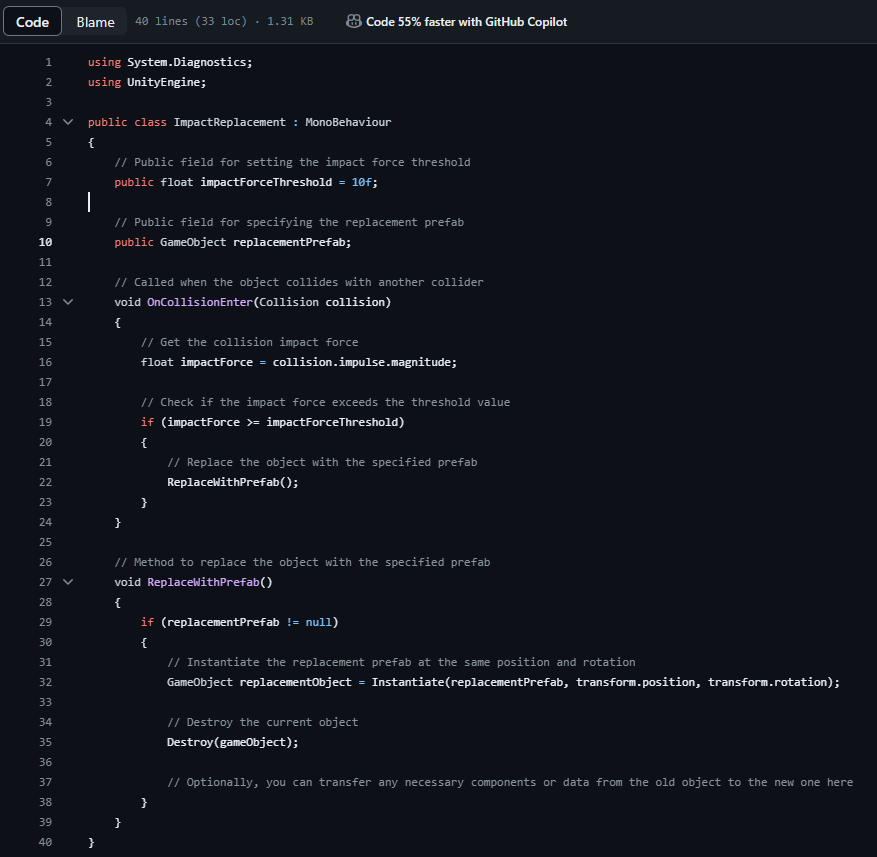


Рис. 8 Тестируемая функция

Данная функция позволяет поменять объект на заранее заготовленный префаб, если к этому объекту приложили достаточно силы. В нашем проекте данная функция используется для разбития бутылок.

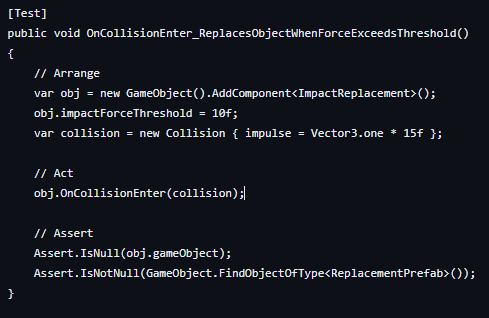


Рис.9 Первая часть теста.

Первая функция теста проверяет, что при вызове метода OnCollisionEnter объекта ImpactReplacement с импульсом столкновения, превышающим порог силы удара, текущий объект был уничтожен и экземпляр ReplacementPrefab был создан. Таким образом, тест проверяет, что замена объекта происходит корректно при превышении порогового значения силы удара.

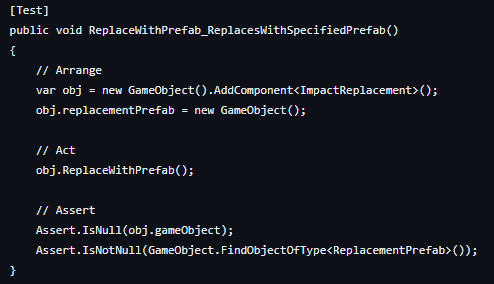


Рис. 10 Вторая часть теста.

Вторая функция теста проверяет, что при вызове метода ReplaceWithPrefab объекта ImpactReplacement, текущий объект был уничтожен и экземпляр ReplacementPrefab был создан. Таким образом, тест проверяет, что замена объекта указанным префабом происходит корректно.

# ***4. Кодирование.***

При разработке проекта, необходимо огласить требования к формату кода, чтобы он читался любым членом команды независимо от того, кто этот код писал.

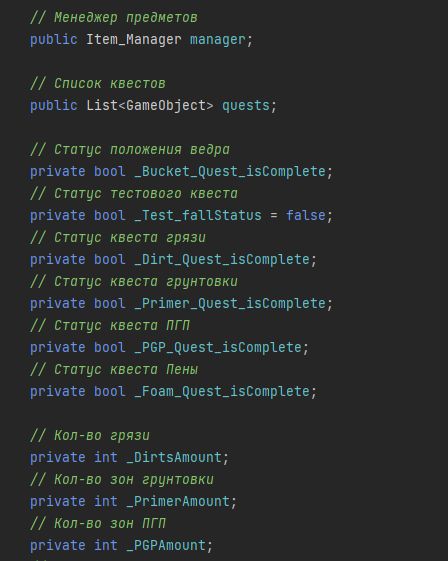


Рис.12.Пример оформления списка переменных.

Правила форматирования кода:

* В одном файле не может быть более 5 классов.
* Все названия классов и методов должны быть в PascalCase. Каждое

слово начинается с большой буквы.

* Все названия переменных/полей должны быть в camelCase.
* Если переменная/поле в одно слово, то необходимо писать с маленькой

буквы.

* Не имеет смысла дописывать Method, Attribute, Class и т.д.
* Если в коде есть аббревиатуры, то вы должны дать расшифровку в

документации класса и в документации метода/функции.

* Название метода должно быть глаголом, название класса -

существительным.

* В одном файле не должно быть более 500 строк кода. В методе не более

100 строк кода. Одна строка не более 120 символов.

* Закрытые (приватные) поля помечаются нижним подчеркиванием.
* Открывающаяся фигурная скобочка должна быть на уровне кода (при

объявлении класса, метода, функции, цикла и т.д.)

* Фигурные скобочки ставятся всегда. Даже если запись в одну строку.

Структура:

1) Base

1.1) Sound

1.1.1) Notify(string, anything)

1.2) BucketQuest

1.2.1) Notify(string, anything)

1.3) … Любой скрипт, наследующийся от MonoBehaviour

1.3.1) Notify(string, anything)

2) Managers

2.1) Item\_Manager

2.1.1) NotifyBucketQuestOpenFaucet(bool)

2.2) Quest\_Manager

2.2.1) NotifyQuestComplete(string)

2.3). … Остальные менеджеры

2.3.1) Любое кол-во методов начинающихся с Notify

3) Repository

3.1) Item\_Repository

3.1.1) Поля (атрибуты) связанные с предметами

3.2) Quest\_Repository

3.2.1) Поля (атрибуты) связанные с квестами

3.3) … Любое другое хранилище данных

3.3.1) Поля (атрибуты) связанные с этим хранилищем

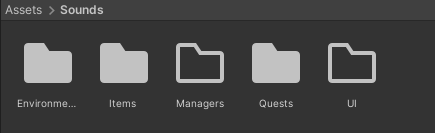


Рис.13. Пример директории в Unity.

Структура директорий в Unity:

1) Assets (Типы объектов)

Вложенные папки (далее тип):

1.1) Material

1.2) Script

1.3) Effect

1.4) Scene

1.5) Texture

1.6) Sound

1.7) Prefab

1.8) Plugin

Все файлы распределяются в директорию, в соответствии с их назначением.

# ***5. Результат работы программы***