**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Научный руководитель  Приглашенный преподаватель департамента программной инженерии | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Поповкин  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

**МНОГОПОЛЬЗОВАТЕЛЬСКАЯ ИГРА «КОЛОНИЗАТОРЫ» ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 81 01-1-ЛУ**

**Исполнитель**

Студент группы БПИ173

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / И.Н. Руденко /

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

**Москва 2018**

**УТВЕРЖДЕН**

**RU.17701729.04.01-01 81 01-1-ЛУ**

**МНОГОПОЛЬЗОВАТЕЛЬСКАЯ ИГРА «КОЛОНИЗАТОРЫ» ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ**

**Пояснительная записка**

**RU.17701729.04.01-01 81 01-1**

**Листов 32**

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл |  |

**Москва 2018**

**СОДЕРЖАНИЕ**

[1. ВВЕДЕНИЕ 3](#_Toc514358798)

[1.1 Наименование программы 3](#_Toc514358799)

[1.2 Документы, на основании которых ведется разработка 3](#_Toc514358800)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 4](#_Toc514358801)

[2.1 Функциональное назначение 4](#_Toc514358802)

[2.2 Эксплуатационное назначение 4](#_Toc514358803)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 5](#_Toc514358804)

[3.1 Постановка задачи на разработку программы 5](#_Toc514358805)

[3.2 Описание функционирования игры 5](#_Toc514358806)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 9](#_Toc514358807)

[4.1 Ориентировочная экономическая эффективность 9](#_Toc514358808)

[4.2 Предполагаемая потребность 9](#_Toc514358809)

[4.3 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 9](#_Toc514358810)

[5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 10](#_Toc514358811)

[Приложение 1 ПРАВИЛА НАСТОЛЬНОЙ ИГРЫ «КОЛОНИЗАТОРЫ 11](#_Toc514358812)

[Приложение 2 ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ 14](#_Toc514358814)

[Приложение 3 ОПИСАНИЕ И ФУКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ,  
ПОЛЕЙ И СВОЙСТВ 15](#_Toc514358815)

**1. ВВЕДЕНИЕ**

**1.1 Наименование программы**

Наименование программы: «Colonizers Game».

**1.2 Документы, на основании которых ведется разработка**

Разработка программы ведется в рамках курсовой работы на тему «Многопользовательская игра “Колонизаторы” для мобильных устройств», в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия».

Основанием для разработки является приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» № 2.3-02/2604-01 от 26.04.2018 "Об изменении тем курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук".

**2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ**

**2.1 Функциональное назначение**

Программа предназначена для использования в развлекательных целях. Также назначением программы является улучшение навыков общения и коммуникации между людьми.

**2.2 Эксплуатационное назначение**

Программа предназначена для использования на устройствах под управлением ОС «Android».

**3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**

**3.1 Постановка задачи на разработку программы**

Настольная игра «Колонизаторы» - одна из самых популярных настольных игр, она известна во всем мире. Задачей являлось разработать приложение для мобильных устройств, которые бы реализовывало основные функции оригинальной игры «Колонизаторы», такие как сбор ресурсов с клеток игрового поля, строительство различных объектов и обмен полученными ресурсами между игроками, и позволяло играть нескольким пользователям на одном устройстве одновременно.

**3.2 Описание функционирования игры**

Основной игровой процесс можно условно разделить на несколько частей (более подробно о них вы можете прочитать в приложении 1 «Правила игры»):

1. Сбор ресурсов с клеток игрового поля, рядом с которыми есть поселение или город игрока. При этом то, с какой именно клетки можно собрать ресурс на данном ходу определяется случайным числом, полученным с помощью броска игровых кубиков, совершаемого в начале хода каждого игрока. На каждой клетке игрового поля есть цифра, которая соответствует числу на кубиках, при котором эта клетка принесет ресурс игрокам, построившим рядом свои поселения или города. Также каждая клетка приносит игрокам ресурс только одного определенного вида.

Данная система реализована с помощью классов Player и Hex, описывающих соответственно класс игрока и игровой клетки (далее - гекс). При строительстве нового поселения экземпляр класса игрока подписывается на событие сбора ресурсов, описанное в классе гекса, тех гексов, которые граничат с построенным поселением. В начале каждого хода (кроме ходов в стадии основания, когда игроки только основывают свои первые поселения), генерируется случайным образом число, далее вызывается метод, который для каждого объекта вызывает метод класса Hex, проверяющий, совпадает ли выпавшее число на кубиках с числом на гексе. В случае, если значения совпадают, вызывается событие, позволяющее объектам игроков, подписанным на него, получить ресурс с данной клетки.

2. Строительство объектов. Во время своего хода игрок может построить дорогу рядом с уже построенной ранее другой своей дорогой, построить поселение на любом перекрестке, к которому проложена его дорога, если на соседнем перекрестке не построено другое поселение, или построить город на месте любого своего поселения. При этом на момент постройки игрок должен иметь необходимое для постройки количество ресурсов каждого вида.

Система строительства реализована с помощью классов Road и Town, описывающих объекты дороги и поселения (непостроенные дороги и поселения также являются объектами этих классов, то есть данные классы реализуют, условно говоря, объект места, подходящего для строительства дороги или города). Каждый из объектов этих классов хранит в себе информацию о том, построен ли объект на этом месте, если нет, то кто из игроков на данный момент может построить здесь объект и проч. Выбор типа объекта для строительства осуществляется игроком в меню строительства. Перед тем, как активировать режим строительства определенного объекта, вызывается метод, проверяющий количество ресурса каждого вида у игрока, и, если количество ресурсов достаточно, активирующий кнопку в меню строительства, позволяющую построить объект определенного вида. При нажатии на кнопку строительства объекта в игровом меню вызывается метод, проверяющий для каждого объекта соответствующего типа, доступен ли он в данный момент для строительства. В том случае, если на данном месте можно построить объект, который игрок собирается построить, активируется элемент Collider2D, позволяющий обработать нажатие на эту часть игрового поля, и создать новый объект а так же занести данные в соответствующий экземпляр класса Road или Town о том, что объект был построен. При этом вызывается событие RoadBuilded или TownBuilded, обрабатываемое другими экземплярами класса Road или Town, что позволяет им правильно реагировать на постройку соседних объектов. Например, при строительстве игроком новой дороги, объекты дорог, расположенных рядом с построенной становятся также доступными для строительства дороги того же игрока.

3. Обмен ресурсами между игроками. Во время своего хода игрок может предложить любому из соперников обменяться ресурсами.

Возможность обмена ресурсами реализована с помощью отдельного игрового меню, в котором игрок может увидеть количество ресурсов других игроков и выбрать соперника для обмена, а так же настроить количество ресурсов каждого вида, которое он готов отдать или хочет получить от другого игрока. Методы, позволяющие выводить меню торговли и совершать сделки описаны в классе TradingScript. Сам обмен ресурсами совершается при помощи метода MakeDeal, описанного в классе GameMethods, который изменяет количество ресурсов, получая доступ к объектам класса Player, реализующим объект игрока, совершающего ход и объект соперника, с которым производится обмен.

Другие классы содержат в себе описания методов, позволяющих приложению корректно отображать игровой интерфейс, управлять различными игровыми объектами и загружать сцены приложения.

**3.3 Описание и обоснование выбора метода организации входных и выходных данных**

Входные данные пользователь передает с помощью графического интерфейса путем нажатия различных кнопок. В меню создания игры реализованы текстовые поля, с помощью которых пользователь программы может ввести имена игроков, которые далее будут отображаться в игре. Выходные данные также представлены с помощью различных частей графического интерфейса. Таким образом для управления функциями программы пользователям не требуется каких-либо специальных навыков, что позволяет пользоваться ей широкому кругу лиц.

**3.4 Описание и обоснование выбора технических и программных средств**

Для работы приложения требуется устройство под управлением ОС «Android». Данная система была выбрана в силу своей широкой распространенности среди пользователей.

Для нормального функционирования приложения на устройстве требуется не менее 25МБ свободной памяти на диске и не менее 500МБ оперативной памяти.

При разработке была использована среда Unity, так как она предоставляет все необходимые возможности для создания подобного приложения. Скрипты, реализующие игровые функции написаны на языке C#. Данный язык был выбран в силу того, что он поддерживается средой Unity, и к тому же является основным языком для изучения на первом курсе образовательной программы «Программная инженерия».

**4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ**

**4.1 Ориентировочная экономическая эффективность**

В рамках данной темы не предусмотрен расчет экономических показателей.

**4.2 Предполагаемая потребность**

Данный продукт предполагается к использованию всеми категориями пользователей устройств под управлением ОС «Android» для развлекательных целей.

**4.3 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами**

Расчет экономических преимуществ не входит в данную работу.

**5. ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ**

1. Хабр (электронный ресурс): https://habr.com
2. stackoverflow (электронный ресурс): https://stackoverflow.com
3. Unity user manual электронный ресурс): https://docs.unity3d.com/Manual/index.html
4. В.В.Подбельский. (2011). *Язык C# Базовый курс.* Москва: "Финансы и статистика".
5. Джозеф Албахари, Б. А. (2013). *С# 5.0 Справочник.* Москва: "И.Д. Вильямс".
6. Хокинг, Д. (2016). *Unity в действии. Мультиплатформенная разработка на C#.* Санкт Петербург: "Питер".
7. Шилдт, Г. (2014). *Полное руководство C# 4.0.* Москва: Издательский дом "Вильямс".

**Приложение 1**

**Правила настольной игры «Колонизаторы»**

**Подготовка к игре Колонизаторы**

Вначале создаётся карта острова, путём произвольного расположения карточек “суши” (имеется шесть типов “суши”). Готовый остров со всех сторон обносится карточками “моря”.

Затем, всем карточкам “Суши” (кроме «Пустыни») присваивается в любом порядке жетон с количественным числом (от 2 до 12).

Каждому игроку выдаётся по две дороги и два поселения одного цвета.

Выкладываются колоды с карточками ресурсов. Первоочерёдность хода определяется броском двух кубиков и сравнением выпавшей суммы.

**Ход игры**

Вначале все игроки, начиная с первого, по очередности выставляют на карту острова по одному поселению и одной примыкающей дороге. Последний игрок поочерёдно выставляет две пары “поселение-дорога”. Далее, против часовой стрелки остальными игроками также ставятся ещё по одному поселению и дороге. От каждого поселения игрок получает по карточке каждого ресурса, местность с которым примыкает к данному поселению.

Сбор сырья. Первым действием каждого игрока является выбрасывание двух кубиков. Сумма выпавших очков указывает на местности с данным номером на жетоне. От этих местностей все игроки, чьи поселения построены на этой же территории, получают по одной карточке, а за город – две карточки соответствующих ресурсов (если в колоде этого ресурса закончились карточки, то игроку не достаются ресурсы).

Строительство. Далее ходящий игрок может построить объект: «Поселение», «Город» или же «Дорогу». Для постройки игроку необходимо иметь соответствующее количество ресурсов.

Для постройки необходимо:

* «Дорога» = «Древесина» + «Кирпичи»;
* «Поселение» = «Древесина» + «Кирпичи» + «Шерсть» + «Зерно»;
* «Город» = две карточки «Зерно» + три карточки «Руда»;

Также у игрока на руках должна быть фишка соответствующая планируемой постройки.

Местоположение на карте, подходящее под условия для возведения конкретного объекта.

Для каждой постройки карточка местности должна удовлетворять следующим требованиям:

* «Дорога». Должна быть построена на свободной грани карточки «Суша». Она должна примыкать к поселению, городу или уже имеющейся дороге.
* «Поселение». Возводятся на перекрёстке 3-х шестигранников «Суша», при наличии связи с другими поселениями игрока через дорожную систему. Нужно наличие на карте свободной от других городов и поселений местности. Также необходимо, чтобы на соседних гранях карточек «Суши» также отсутствовали какие-либо постройки.
* «Город». Имеющиеся Поселения игрока можно развить до Города.

Торговля. Следующее, что может сделать игрок, это торговля. Торговля представляет собой обмен ресурсами с другими игроками.

Предложение торговли, помимо ходящего игрока, могут делать все остальные игроки, но только играющему на данный момент участнику. Сам же игрок во время хода может торговать со всеми.

**Конец игры**

После набора одним из игроков 10 очко игра прекращается.

Очки начисляются за следующие достижения:

* Построенное поселение. 1 очко;
* Построенный город. 2 очка;

**Приложение 2**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ**

|  |  |
| --- | --- |
| **Класс** | **Описание** |
| MainMenu | Содержит описание методов, необходимых для работы главного меню приложения |
| CreatingGameMenu | Содержит описание методов, необходимых для работы меню создания игры |
| GameScript | Управляет основными функциями игры, вызывает методы других классов. |
| GameMethods | Содержит описание отдельных методов, необходимых для игрового процесса |
| RoadObjectScript | Управляет объектом дороги на игровом поле |
| TownObjectScript | Управляет объектом города на игровом поле |
| TradingScript | Управляет окном торговли и окном обмена ресурсами |
| LoadScene | Содержит описание метода Next, позволяющего переключаться между сценами |
| Hex | Представляет объект игровой клетки - гекса |
| Player | Представляет объект игрока |
| Road | Представляет объект места для строительства дороги |
| Town | Представляет объект места для строительства поселения или города |

**Приложение 3**

**ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ МЕТОДОВ, ПОЛЕЙ И СВОЙСТВ**

**Класс MainMenu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| ExitGameBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Выход» в главном меню игры. |

**Класс CreatingGameMenu**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| \_3players | public | GameObject | Ссылка на объект, представляющий интерефейс для ввода имен трех игроков |
| \_4players | public | GameObject | Ссылка на объект, представляющий интерфейс для ввода имен четырех игроков |
| player1input  player2input  player3input  player4input | public | GameObject | Ссылки на текстовые поля для ввода имен игроков |
| playerNames | public | string[] | Хранит введенные пользователем имена игроков |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| AddPlayer | public | void | - | Добавляет поле для ввода имени четвертого игрока |
| RemovePlayer | public | void | - | Убирает поле для ввода имени четвертого игрока |
| SavePlayerNames | public | void | - | Заносит введенные имена игроков в поле playerNames. Если имена не были введенs, заносит имена по умолчанию |

**Класс Game Script**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| playerNames  playerNames\_4players | public | Text[] | Текстовые поля, отображающие имена игроков |
| resLabels | public | Text[] | Текстовые поля, отображающие количество ресурсов у игрока в данный момент |
| players | private | Player | Ссылки на объекты игроков |
| currentPlayerNum | private | int | Номер игрока, который совершает ход в данный момент |
| roads | private | List<Road> | Массив объектов дорог |
| towns | private | List<Town> | Массив объектов городов |
| playerColors | private | Color[] | Массив объектов типа Color, задающих цвет игровых построек игроков |
| mountains  lake forest grassland fields | private | Sprite | Спрайты для разных местностей, из которых состоит игровое поле |
| hexObjects | public | SpriteRenderer[] | Объекты для отображения местностей на игровом поле |
| hexes | private | Hex[] | Объекты местностей |
| townObjArray | public | GameObject[] | Массив мест для строительства поселений и городов |
| roadObjArray | public | GameObject[] | Массив мест для строительства дорог |
| infoScreen | public | GameObject | Экран уведомлений о смене хода в начальной фазе игры |
| infoScreenText | public | Text | Текстовое поле для вывода уведомления о смене хода в начальной фазе игры |
| infoScreenButton | public | Button | Кнопка продолжения игры на экране уведомлений о смене хода в начальной фазе игры |
| infoScreenBtnText | public | Text | Текст кнопки продолжения в начальной фазе игры |
| changeTurnScreen | public | GameObject | Экран уведомлений о смене хода в основной фазе игры |
| changeTurnPlayerTxt | public | Text | Поле для вывода имени игрока, которому передается ход в основной фазе игры |
| changeTurnCubesTxt | public | Text | Поле для вывода значения, выпавшего на кубиках |
| buildingScreen | public | GameObject | Меню строительства |
| buildingScreenBtns | public | Button[] | Кнопки в меню строительства |
| finalScreen | public | GameObject | Экран подсчета очков игроков |
| finalScreen\_player4Obj | public | GameObject | Поле для вывода количества очков четвертого игрока |
| quitBtn | public | Button | Кнопка выхода из игры на экране подсчета очков |
| continueGameBtn | public | Button | Кнопка продолжения игры на экране подсчета очков |
| endGamePlayerNames | public | Text[] | Текстовые поля для вывода имен игроков на экране подсчета очков |
| endGamePlayerScores | public | Text[] | Текстовые поля для вывода количества очков у игроков на экране подсчета очков |
| standartInterface  standartInterface\_3Players  standartInterface\_4Players | public | GameObject | Стандартный интерфейс верхней части экрана |
| buildingInterface | public | GameObject | Интерфейс режима строительства |
| buildingInterface\_CanelBtn | public | GameObject | Кнопка отмены строительства |
| endGameBtn  tradeBtn  buildBtn  endTurnBtn | public | Button | Кнопки игрового меню |
| phase | private | int | Номер текущей фаза игры |
| playersCount | public | int | Количество игроков |
| rand | private | Random | Генератор случайных значений |
| cubesValue | private | int | Последнее выпавшее число на кубиках |

|  |  |  |  |
| --- | --- | --- | --- |
| Свойства | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| Players | public | Player[] | Доступ на чтение к полю players |
| CurrentPlayerNum | public | int | Доступ на чтение к полю currentPlayerNum |
| Roads | public | List<Road> | Доступ на чтение к полю roads |
| Towns | public | List<Town> | Доступ на чтение к полю towns |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| TownBuildedHandler\_SettlePhase | public | void | string townID  bool townBuided | Обрабатывает постройку поселения в фазе основания |
| RoadBuildedHandler\_SettlePhase | public | void | string roadID | Обрабатывает постройку дороги в фазе основания |
| TownBuildedHandler\_MainPhase | public | void | string townID  bool townBuilded | Обрабатывает постройку города в основной фазе |
| RoadBuildedHandler\_MainPhase | public | void | string roadID | Обрабатывает постройку дороги в основной фазе |
| Start | public | void | - | Загружает сцену |
| InfoScreenBtn\_OnClick\_SettlePhase | public | void | - | Обрабатывает нажатие кнопки «Продолжить» на экране уведомлений о смене хода в начальной фазе игры |
| StartMainPhase | public | void | - | Начинает основную фазу игры |
| ChangeTurnScreenBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Продолжить» на экране уведомлений о смене хода в основной фазе игры |
| EndTurnBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Закончить ход» в игровом меню в фазе основания |
| EndTurnBtn\_ObClick\_MainPhase | public | void | - | Обрабатывает нажатие кнопки «Закончить ход» в игровом меню в основной фазе |
| BuildBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Строительство» в игровом меню |
| BuildRoadBtn\_OnClick  BuildVillageBtn\_OnClick  BuildTownBtn\_OnClick | public | void | - | Обрабатывают нажатие соответствующих кнопок в меню строительства |
| BuildingInterfaceCancelBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Отмена» на интерфейсе строительства в верхней части экрана |
| EndGameBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Закончить игру» в игровом меню |
| FinalScreen\_ContinueBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Продолжить» на экране подсчета очков |
| QuitBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Выйти из игры» на экране подсчета очков |

**Класс GameMethods**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| CreateIsland | public | void | SpriteRenderer[] HexObjects  Hex[] hexes  Sprite mountains  Sprite lake  Sprite forest  Sprite grassland  Sprite fields | Случайным образом генерирует игровое поле |
| ActivateRoadBuildingMode | public | void | List<Road> roads | Активирует объекты дорог на игровом поле, чтобы пользователь мог выбрать объект для строительства дороги |
| ActivateVillageBuildingMode | public | void | List<Town> towns | Активирует объекты городов на игровом поле, чтобы пользователь мог выбрать объект для строительства поселения |
| ActivateTownBuildingMode | public | void | List<Town> towns | Активирует объекты городов на игровом поле, так чтобы пользователь смог основать город на месте построенного поселения |
| DeleteBorderingTownObjects | public | void | Town buildedTown  List<Town> towns | Блокирует для строительства объекты городов, граничащие с построенным поселением |
| GetResources | public | void | Hex[] hexes  int cubesValue | Позволяет игрокам получить ресурсы с клеток, номер на которых совпадает с числом, выпавшим на кубиках |
| AddHexListeners | public | void | string townID  Hex[] hexes  Player player | Подписывает методы обработчики класса Player на событие сбора ресурсов с гексов, граничащих с построенным поселением |
| LoadResValues | public | void | Text[] resLabels  Int playerNum | Загружает информацию о количестве ресурсов каждого вида у игрока |
| LoadPlayerInterface | public | void | Text[] playerNames  Text[] resLabels | Загружает интерфейс пользователя |
| DeactivateRoadObjects | public | void | List<Road> roads | Деактивирует объекты дорог |
| DeactivateTownObjects | public | void | List<Town> towns | Деактивирует объекты городов |
| MakeDeal | public | void | int[] resValues | Меняет количество ресурсов у игроков, реализуя обмен ресурсами |

**Класс TradingScript**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| player1Text  player2Text | public | Text | Текстовые поля, отображающие имена игроков в меню торговли |
| player1Text\_4players  player2Text\_4players  player3Text\_4players | public | Text | Текстовые поля, отображающие имена игроков в меню торговли при игре вчетвером |
| player1resLabels  player2resLabels | public | Text[] | Текстовые поля, отображающие количество ресурсов каждого вида у игроков в меню торговли |
| player1resLabels  player2resLabels  player3resLabels | public | Text[] | Текстовые поля, отображающие количество ресурсов каждого вида у игроков в меню торговли |
| tradingScreen3players  tradingScreen4players | public | GameObject | Меню торговли для 3-х и 4-х игроков |
| exchangeScreen | public | GameObject | Экран обмена ресурсами между игроками |
| player1Txt\_ExchangeScreen  player2txt\_ExchangeScreen | public | Text | Текстовые поля, отображающие имена игроков на экране обмена ресурсами |
| player1ResLabels\_ExchangeScreen  player2ResLabels\_ExchangeScreen | public | Text[] | Текстовые поля, отображающие количество обмениваемых игроками ресурсов каждого вида |
| exchangingPlayerNum | private | int | Номер игрока, с которым производится обмен ресурсами |
| playersCount | private | int | Количество игроков |
| exchangingPlayer  currentPlayer | private | Player | Объекты игроков, участвующих в сделке |
| exchangingResValues | private | int[] | Количество обмениваемых ресурсов каждого вида |
| resLabels | public | Text[] | Текстовые поля, отображающие количество ресурсов у игрока в основном игровом интерфейсе |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| TradeBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Торговля» в игровом меню |
| CancelBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Отмена» в меню торговли |
| LoadExchangeScreen | public | void | int playerNum | Загружает экран торговли |
| CurrentPlayerAddBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки добавления ресурса текущему игроку на экране обмена ресурсами |
| ExchangingPlayerAddBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки добавления ресурса игроку, совершающему обмен на экране обмена ресурсами |
| ConfirmBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Подтвердить» на экране обмена ресурсами |
| ExchangeScreenCancelBtn\_OnClick | public | void | - | Обрабатывает нажатие кнопки «Отмена» в меню торговли |

**Класс RoadObjectScript**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| sprite | public | Sprite | Спрайт дороги |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| OnMouseDown | public | void | - | Позволяет объекту реагировать на нажатие |

**Класс TownObjectScript**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| villageSprite | public | Sprite | Спрайт поселения |
| townSprite | public | Sprite | Спрайт города |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| OnMouseDown | public | void | - | Позволяет объекту реагировать на нажатие |

**Класс LoadScene**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| Next | public | void | - | Загружает следующую сцену |

**Класс Hex**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| resType | private | string | Тип собираемого ресурса |
| gettingResValue | private | int | Значение на кубиках, при котором с гекса собирается ресурс |

|  |  |  |  |
| --- | --- | --- | --- |
| Свойства | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| ResType | public | string | Доступ на чтение к полю resType |
| GettingResValue | public | int | Доступ на чтение к полю gettingResValue |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| CheckGettingRes | public | void | int cubesValue | Проверяет, совпадает ли число на кубиках с числом на гексе |

**Класс Player**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| name | private | string | Имя игрока |
| color | private | Color | Цвет игровых объектов игрока |

|  |  |  |  |
| --- | --- | --- | --- |
| Свойства | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| Name | public | string | Доступ на чтение к полю name |
| Color | public | Color | Доступ на чтение к полю color |
| Wood  Bricks  Stone  Wool  Wheat | public | int | Количество ресурсов каждого вида у игрока |
| Score | public | int | Количество очков у игрока |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| GettingResHandler | public | void | string resType | Обрабатывает событие сбора ресурсов |
| Build | public | void | string objType | Отнимает у игрока ресурсы после постройки объекта |

**Класс Road**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| name | private | string | Название объекта |
| gameObject | private | GameObject | Ссылка на игровой объект дороги |
| hexes | private | string[] | Хранит названия граничащих гексов |

|  |  |  |  |
| --- | --- | --- | --- |
| Свойства | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| Name | public | string | Доступ на чтение к полю name |
| EnableFor | public | List<Player> | Возвращает список игроков, которые могут построить дорогу на этом месте |
| Builded | public | bool | Хранит информацию о том, построена ли на этом месте дорога |
| Hexes | public | string[] | Хранит названия граничащих гексов |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| RoadBuildedHandler | public | void | string buildedObjName | Обрабатывает событие, вызываемое при постройке новой дороги |
| SetActive | public | void | bool activate | Делает соответствующий игровой объект активным или неактивным |

**Класс Town**

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| name | private | string | Название объекта |
| gameObject | private | GameObject | Ссылка на игровой объект дороги |
| hexes | private | string[] | Хранит названия граничащих гексов |
| enableFor | private | List<Players> | Хранит список игроков, которые могут построить объект на этом месте |
| roads | private | List<Road> | Хранит список граничащих дорог |

|  |  |  |  |
| --- | --- | --- | --- |
| Свойства | | | |
| **Имя** | **Доступ** | **Тип** | **Назначение** |
| Name | public | string | Доступ на чтение к полю name |
| EnableFor | public | List<Player> | Доступ на чтение к полю enableFor |
| Builded | public | string | Хранит информацию о том, построен ли объект на этом месте |
| Hexes | public | string[] | Доступ на чтение к полю hexes |
| Roads | public | List<Road> | Доступ на чтение к полю roads |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| **Имя** | **Доступ** | **Тип** | **Аргументы** | **Назначение** |
| TownBuildedHandler | public | void | string buildedObjName | Обрабатывает событие, вызываемое при постройке нового объекта города или поселения |
| BorderingRoadBuildedHandler | public | void | Road buildedRoad | Обрабатывает событие, вызываемое при постройке новой дороги рядом с данным объектом |
| SetActive | public | void | bool activate | Делает соответствующий игровой объект активным или неактивным |

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Изм. | Номера листов (страниц) | | | | Всего листов (страниц) в документе | № документа | Входящий № сопроводительного документа и дата | Подпись | Дата |
| измененных | замененных | новых | аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |