**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Профессор департамента программной  инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Авдошин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **ПРИЛОЖЕНИЕ АКТИВНОСТИ И УВЕДОИЛЕНИЯ ВШЭ**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель  студент группы БПИ182  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /М.А. Шакура /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. | |
|  | | |
|  | |  |

**Москва 2019**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.04.01-01 81 01-1-ЛУ |  | |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **ПРИЛОЖЕНИЕ АКТИВНОСТИ И УВЕДОИЛЕНИЯ ВШЭ**  **Пояснительная записка**  **RU.17701729.04.01-01 81 01-1**  **Листов 31** | | | | |
|  | |  | | |
|  | | |
|  | | | | |
|  | | | |  |

**Москва 2018**

**АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программе «GravityScience» («Кроссплатформенная обучающая игра с моделированием реактивного движения и космических тел»).

В разделе «Введение» указано наименование программы, краткое наименование программы и документы, на основании которых ведется разработка.

В разделе «Назначение и область применения» указано функциональное назначение программы, эксплуатационное назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

− постановка задачи на разработку программы;

− описание алгоритма и функционирования программы;

− описание метода организации входных и выходных данных;

− описание и обоснование выбора состава технических и программных средств.

В разделе «Ожидаемые технико-экономические показатели» указана предполагаемая потребность и экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами.

Настоящий документ разработан в соответствии с требованиями:

1) ГОСТ 19.101-77 Виды программ и программных документов;

2) ГОСТ 19.102-77 Стадии разработки;

3) ГОСТ 19.103-77 Обозначения программ и программных документов;

4) ГОСТ 19.104-78 Основные надписи;

5) ГОСТ 19.105-78 Общие требования к программным документам;

6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом;

7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению.

Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78, ГОСТ 19.604-78.

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в Приложении 1 настоящей пояснительной записки.

ОГЛАВЛЕНИЕ

[1. ВВЕДЕНИЕ 5](#_Toc514095780)

[1.1. Наименование программы 5](#_Toc514095781)

[1.2. Документы, на основании которых ведется разработка 5](#_Toc514095782)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 6](#_Toc514095783)

[2.1. Назначение программы 6](#_Toc514095784)

[2.1.1. Функциональное назначение 6](#_Toc514095785)

[2.1.2. Эксплуатационное назначение 6](#_Toc514095786)

[2.2. Краткая характеристика области применения 6](#_Toc514095787)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 7](#_Toc514095788)

[3.1. Постановка задачи на разработку программы 7](#_Toc514095789)

[3.2. Описание алгоритма и функционирования программы 7](#_Toc514095790)

[3.2.1. Описания алгоритма работы программы при перенесении данных из Google Таблицы 7](#_Toc514095791)

[3.2.2. Описания алгоритма взаимодействия экранов программы 7](#_Toc514095792)

[3.2.3. Описание алгоритма заполнения меню уровней 7](#_Toc514095793)

[3.2.4. Описание алгоритма обновления и вытаскивания из меню планет 8](#_Toc514095794)

[3.2.5. Описания алгоритма сохранения данных 9](#_Toc514095795)

[3.2.6. Описание алгоритма удаления игровых объектов 9](#_Toc514095796)

[3.2.7. Реактивное движение 10](#_Toc514095797)

[3.2.8. Гравитация 10](#_Toc514095798)

[3.2.9. Астероид и черная дыра 10](#_Toc514095799)

[3.2.10. Оптимизация сравнения строк 10](#_Toc514095800)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных 10](#_Toc514095801)

[3.3.1. Описание метода организации входных и выходных данных 10](#_Toc514095802)

[3.3.2. Обоснования выбора метода организации входных и выходных данных 11](#_Toc514095803)

[3.4. Описание и обоснование выбора состава технических и программных средств 11](#_Toc514095804)

[3.4.1. Состав технических и программных средств 11](#_Toc514095805)

[3.4.2. Обоснование выбора технических и программных средств 11](#_Toc514095806)

[ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 12](#_Toc514095807)

[3.5. Предполагаемая потребность 12](#_Toc514095808)

[3.6. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 12](#_Toc514095809)

[ПРИЛОЖЕНИЕ 1 13](#_Toc514095810)

[ТЕРМИНОЛОГИЯ 13](#_Toc514095811)

[ПРИЛОЖЕНИЕ 2 14](#_Toc514095812)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 14](#_Toc514095813)

[ПРИЛОЖЕНИЕ 3 15](#_Toc514095814)

[ПРАВИЛА ИГРЫ 15](#_Toc514095815)

[ПРИЛОЖЕНИЕ 4 16](#_Toc514095816)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ 16](#_Toc514095817)

[ПРИЛОЖЕНИЕ 5 17](#_Toc514095818)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ 17](#_Toc514095819)

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы – «Кроссплатформенная обучающая игра с моделированием реактивного движения и космических тел» или «Cross-platform educational game with jet propulsion and space objects simulation».

Наименование программы для пользователя – «GravityScience».

## Документы, на основании которых ведется разработка

Приказ декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» №2.3-02/1212-01 от 12.12.2017 «Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук».

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

## Назначение программы

### Функциональное назначение

Программа предназначена для моделирования физических законов в игровой форме.

На каждом уровне пользователю задаются параметры ракеты: масса, направление движения, скорость газовой струи, масса выбрасываемых газов, ежесекундный расход газов, точка старта и успешного финиша. Игроку предлагается выбор из нескольких видов планет или других космических тел, характеристики которых он может прочитать с помощью дополнительного нажатия по ним (повторное нажатие по экрану закрывает описание). Игроку необходимо расставить космические тела на экране так, чтобы после запуска ракеты с данными характеристиками она оказалась в точке успешного финиша. Физика в игре приближена к реальной. После успешного прохождения уровня пользователю начисляются баллы (от 0 до 3), чем меньше космических тел игрок использовал, тем выше будут баллы.

Следующий уровень открывается только после успешного прохождения предыдущего. Для прохождения уровня игрок должен нажать на него.

Программа осуществляет возможность включения и выключения музыки в игре.

### Эксплуатационное назначение

Данная игра может применяться для проведения досуга пользователями любого возраста. Программа будет полезна в сфере образования, как доступный и интересный способ объяснить человеку основные понятия движения в космическом пространстве.

## Краткая характеристика области применения

«Кроссплатформенная обучающая игра с моделированием реактивного движения и космических тел» – программа, позволяющая проводить досуг и развивать логическое мышление, а также способная объяснить обычному пользователю устройство гравитации и реактивное движение, рассказать о планетах солнечной системы и других космических телах. Область применения программы – развлекательная и образовательная сфера.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## Постановка задачи на разработку программы

Разрабатываемая программа должна соответствовать:

- описанным в техническом задании функциональным характеристикам («Кроссплатформенная обучающая игра с моделированием реактивного движения и космических тел». Техническое задание)

- описанным в техническом задании требованиям к интерфейсу («Кроссплатформенная обучающая игра с моделированием реактивного движения и космических тел». Техническое задание)

- описанным в техническом задании требованиям к надежности («Кроссплатформенная обучающая игра с моделированием реактивного движения и космических тел». Техническое задание)

## Описание алгоритма и функционирования программы

### Описания алгоритма работы программы при перенесении данных из Google Таблицы

Для удобства изменения данных в игре (добавления, изменения или удаления уровней, планет, табличек с текстом и ракет) в программе используются Google Таблицы. Алгоритм работы программы с которыми приведен на рис. 1:



*Рисунок 1.* Перенесение данных из Google Таблицы

### Описания алгоритма взаимодействия экранов программы

Алгоритм взаимодействия экранов программы приведен на рис. 2:



*Рисунок 2.* Взаимодействие экранов программы

### Описание процесса заполнения меню уровней

Меню, в котором отображаются уровни, заполняется по количеству уровней, пройденных игроком, каждый уровень имеет вид планеты.

for (int i = 0; i < Mathf.Min(staticData.levels.list.Count, PlayerPrefs.GetInt("CanDo") + 1); ++i)

{

GameObject go = GameObject.Instantiate(level, this.transform.position, this.transform.rotation);

go.transform.SetParent(levelsContent.transform);

levelsButtons.Add(go);

go.GetComponent<LevelController>().Init(i);

}

Количество баллов, набранных игроком на каждом уровне отображается на картинке планеты.

staticData = GameObject.Find("[StaticData]").GetComponent<StaticData>();

int buffer = levelId;

this.transform.Find("Text").GetComponent<Text>().text = staticData.levels.list[levelId].KeyName;

if (PlayerPrefs.HasKey("star" + levelId))

{

this.transform.Find("Star").GetComponent<Text>().text = PlayerPrefs.GetInt("star" + levelId).ToString();

}

else

{

this.transform.Find("Star").GetComponent<Text>().text = "0";

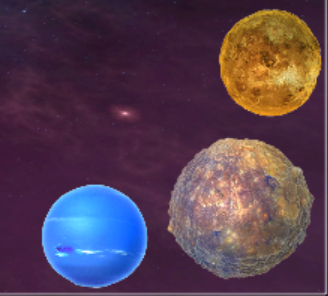
}

this.gameObject.GetComponent<Button>().onClick.AddListener(delegate { LoadLevelClick(buffer); });

### Описание алгоритма обновления и вытаскивания из меню планет

На каждом уровне пользователь может переключать текущую планету (отображенную в правом нижнем углу) на следующую (отображенную над текущей планетой) или на предыдущую (отображенную слева от текущей планеты) нажатием на соответствующую планету (см. рис. 3).

По нажатию на текущую планету открывается таблица с её характеристиками, повторное нажатие закрывает ее.



*Рисунок 3.* Выбор планет

Для добавления планеты на игровое поле пользователь должен переместить текущую планету на нужное место. При этом на игровом поле появляется дубликат текущей планеты с соответствующими массой и радиусом.

### Описания алгоритма сохранения данных

С помощью класса PlayerPrefs в UnityEngine в игре выполняется сохранение настроек музыки, внешнего вида ракеты, баллов, набранных за каждый пройденный уровень, номер последнего выбранного уровня и номер крайнего уровня, который пользователь может пройти, расположение и параметры планет на уровнях.

Схема сохранения уровня представлена на рис. 5:



*Рисунок 5.* Сохранение уровня

### Описание процесса удаления игровых объектов

Планета удаляется, если пользователь перемещает её за пределы игрового поля или на иконку «корзины».

Рассчитываются координаты области удаления – расположение иконки «корзины» и области за пределами игрового поля, программа проверяет, находится ли курсор в области удаления, если да, объект удаляется:

var v3 = Input.mousePosition;

v3 = Camera.main.ScreenToWorldPoint(v3);

Vector3[] corners = new Vector3[4];

this.GetComponent<RectTransform>().GetWorldCorners(corners);

Rect newRect = new Rect(corners[0], corners[2] - corners[0]);

isMouseOver = newRect.Contains(v3);

Удаление ракеты не предусмотрено, если ракета вылетает за пределы игрового поля, уровень начинается заново.

### Реактивное движение

Реактивное движение в игре подчиняется уравнению для материальной точки переменной массы русского ученого Ивана Всеволодовича Мещерского: , где – текущая масса ракеты, – ежесекундный расход массы, – скорость газовой струи, – внешние силы, а – ускорение. В игре внешние силы отсутствуют.

За реактивное движение ракеты отвечает класс GameController, при запуске ракеты раз в фиксированный интервал времени (FixedUpdate) высчитывается текущее ускорение () по формуле.

### Гравитация

Гравитация в игре подчиняется закону всемирного тяготения английского ученого Исаака Ньютона и выражается формулой , где – некоторая константа, – масса тела, – расстояние между центрами масс тел.

В игре возможность притягиваться у планет отключается. За гравитацию отвечает класс GravityController, который собирает для объекта список всех объектов, которым он может притягиваться и высчитывает силу притяжения.

### Астероид и черная дыра

Черная дыра – это два игровых объекта, один из которых служит входной точкой для ракеты, а другой – выходной. Перемещает ракету между этими точками.

Астероид – игровой объект, имеющий гравитацию и летающий по определенной траектории. При столкновении с астероидом игрок начинает уровень заново.

### Оптимизация сравнения строк

Для оптимизации процесса сравнения строк в некоторых местах программы используется встроенное хэширование.

## Описание и обоснование выбора метода организации входных и выходных данных

### Описание метода организации входных и выходных данных

Входные данные в программе – действия пользователя в игре (нажатие на кнопку, передвижения игрового объекта). Приложение предоставляет возможность открыть ранее сохранённой и установленной версии в виде apk файла (для смартфона на платформе Android) или exe файла (для компьютера с операционной системой Windows).

Промежуточные входные данные при сборке проекта – данные в формате JSON.

Выходные данные в программе – отклики приложения на действия пользователя в игре.

В программе предусмотрена функция сохранения изменений в игре в виде данных типа PlayerPrefs.

### Обоснования выбора метода организации входных и выходных данных

Механизм десериализации был выбран потому, что он позволяет сократить объем исходного кода и восстановить объекты в их первоначальном виде.

## Описание и обоснование выбора состава технических и программных средств

### Состав технических и программных средств

Для работы программы необходим следующий состав технических средств:

Для компьютера:

- процессор не ниже Intel Core i3 или аналогичный с тактовой частотой не ниже 1 Ггц;

- 512 мб ОЗУ или более;

- жесткий диск с объемом свободной памяти не менее 512 мб;

- VGA-совместимые видеоадаптер и монитор;

- тачпад или компьютерная мышь.

Для смартфона:

- процессор с тактовой частотой не ниже 1 Ггц;

- 230 мб оперативной памяти или более;

- объем свободной памяти не менее 112 мб.

Для работы программы необходим следующий состав программных средств:

Для компьютера:

- операционная система Windows XP и выше;

- библиотека Microsoft .NET Framework 3.5 и выше.

Для смартфона:

- платформа Android 4.1 и выше.

### Обоснование выбора технических и программных средств

Данный состав технических и программных средств позволит программе в хорошем качестве отображать игровые объекты, осуществлять быстрый отклик на действия пользователя и работать без перебоев.

# ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Предполагаемая потребность

«Кроссплатформенная обучающая игра с моделированием реактивного движения и космических тел» будет востребована в сфере развлечений, как интересный и полезный способ проведения досуга. Программа может быть использована в образовательных учреждениях для демонстрации и объяснения основных понятий, связанных с космосом.

## Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами

Поиск в сети Интернет на момент создания приложения не выявил аналогов данной программы.

Данное приложение:

1. распространяется бесплатно;
2. не требует вложения денежных средств во время использования;
3. имеет неограниченный срок службы.

# ПРИЛОЖЕНИЕ 1

## ТЕРМИНОЛОГИЯ

Ниже приведен список необходимых терминов для ознакомления.

**Главное меню** – экран, который видит пользователь после запуска программы.

**Игровое поле** – участок экрана, на котором пользователь может манипулировать игровыми объектами.

**Ракета** – основной игровой объект, имеющий вид летательного аппарата с реактивным двигателем.

**Масса** – мера инертности тела.

**Скорость** – степень быстроты передвижения.

**Выбрасываемые газы** – то, что отделяется от ракеты в следствие реактивного движения.

**Скорость газовой струи** – относительная скорость отделяющихся частиц.

**Полёт** – моделирование движения тела.

**Запуск ракеты** – начало полёта.

**Точка старта** – точка, из которой осуществляется запуск ракеты.

**Планета** – игровой объект, имеющий вид небесного тела, вращающегося по орбите вокруг звезды или её остатков.

**Космические тела** – игровой объект, имеющий вид объекта в космическом пространстве.

**Характеристики** – набор некоторых свойств из пунктов 5 – 9 или каких-то других.

**Уровень** – задание, которое надо пройти пользователю.

**Успешное прохождение уровня** – завершение уровня пользователем в следствие того, что он справился с уровнем.

**Точка успешного финиша** – место, через которое необходимо пролететь ракете для успешного прохождения уровня пользователем.

**Баллы** – игровые единицы, являющиеся поощрением пользователя за успешное прохождение уровня.

# ПРИЛОЖЕНИЕ 2

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

2. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

3. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

4. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

5. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

7. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

8. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.

9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способ

10. Реактивное движение [Электронный ресурс]// URL: https://college.ru/astronomy/course/content/chapter3/section2/paragraph1/theory.html (Дата обращения: 11.05.2018, режим доступа: свободный).

11. Классическая теория тяготения Ньютона [Электронный ресурс]// URL: https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F\_%D1%82%D0%B5%D0%BE%D1%80%D0%B8%D1%8F\_%D1%82%D1%8F%D0%B3%D0%BE%D1%82%D0%B5%D0%BD%D0%B8%D1%8F\_%D0%9D%D1%8C%D1%8E%D1%82%D0%BE%D0%BD%D0%B0 (Дата обращения: 11.05.2018, режим доступа: свободный).

12. Англоязычная документация Unity [Электронный ресурс]// URL: https://docs.unity3d.com/Manual/index.html (Дата обращения: 11.05.2018, режим доступа: свободный).

# ПРИЛОЖЕНИЕ 3

## ПРАВИЛА ИГРЫ

На каждом из уровней игрок узнает параметры ракеты (её массу, размер, направление движения, скорость газовой струи, ежесекундный расход газов и массу выбрасываемых газов), а также точку старта и успешного финиша.

Пользователю предлагается выбор из различных планет. Игрок выбирает необходимое ему количество этих объектов, расставляет их на игровом поле и запускает ракету. Ракета совершает полёт, смоделированный аналогично тому, как он проходил бы в реальных условиях.

Цель игры – пройти все уровни, в каждом уровне надо долететь до точки успешного финиша (галактики).

Пользователь может прочитать сводку о предлагаемых ему на выбор космических телах и узнать их характеристики, необходимые для удачного прохождения уровня.

# ПРИЛОЖЕНИЕ 4

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 4.1

Описание и функциональное назначение классов

|  |  |
| --- | --- |
| Класс | Назначение |
| AsteroidController | Контролирует поведение астероида. |
| Blackhole | Контролирует поведение чёрной дыры. |
| DrawPlanet | Отрисовывает процесс вытягивания из меню и выставления планет на экране. |
| ExitController | Контролирует меню "Exit". |
| GameController | Отвечает за работу и отрисовку игрового меню. |
| GravityController | Контроль всех объектов, имеющих гравитацию. |
| Help | Сериализует тексты. |
| HelpController | Контролирует текстовые таблички в меню "Help". |
| InFinish | Отслеживает нахождение элементов в зоне финиша. |
| Level | Описание текущего уровня. |
| LevelController | Контролирует кнопку меню уровней. |
| LevelSave | Поддерживает сохранение уровня. |
| LevelsController | Контролирует меню уровней. |
| MenuController | Контролирует главное меню. |
| Model | Поддерживает текущее состояние игры. |
| PlayObject | Игровые объекты. |
| RocketController | Инициализирует ракету при запуске сцены. |
| Save | Поддерживает глобальное сохранение. |
| SettingsController | Контролирует меню "Setting". |
| StaticData | Отвечает за хранение не изменяемых во время игры данных, а также за их загрузку и обработку. |
| TablePlanetController | Контролирует таблицу с характеристиками планеты. |
| TableRocketController | Контролирует таблицу с характеристиками ракеты. |
| Trash | Корзина для удаления игровых объектов. |
| Planets | Обертка планет (wrapper) для json. |
| Levels | Обертка уровней (wrapper) для json. |
| HelpText | Обертка табличек помощи (wrapper) для json. |

# ПРИЛОЖЕНИЕ 5

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ

Таблица 5.1

Описание полей, методов и свойств класса AsteroidController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| startPosition | | private | | Vector2 | | Начальная позиция астероида. | |
| radius | | private | | float | | Радиус описываемой окружности, при движении астероида. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Start | private | | void | | - | | Вызывается в начале работы скрипта, инициализирует начальную позицию астероида. |
| Update | private | | void | | - | | Вызывается раз в кадр, меняет позицию астероида согласно выбранной траектории (окружность). |

Таблица 5.2

Описание полей, методов и свойств класса Blackhole.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| otherBlackhole | | private | | GameObject | | Вторая чёрная дыра. Объект, из которого должна вылетать ракета, после того, как она влетает в чёрную дыру. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Start | private | | void | | - | | Инициализация, находит вторую чёрную дыру. |
| OnTriggerEnter2D | private | | void | | Collider2D other | | Обрабатывает столкновение с чёрной дырой, меняет позицию ракеты на позицию второй чёрной дыры. |

Таблица 5.3

Описание полей, методов и свойств класса DrawPlanet.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| prefab | | public | | GameObject | | Модель планеты. | |
| curPlanet | | private | | PlayObject | | Планета, которую тянет пользователь. | |
| curDrag | | private | | GameObject | | Текущая планета. | |
| nowRocket | | private | | GameObject | | Текущая ракета. | |
| trash | | private | | Trash | | Корзина для удаления планет. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Init | public | | void | | PlayObject planet | | Инициализация планеты, которую необходимо отрисовывать. |
| OnBeginDrag | public | | void | | PointerEventData eventData | | Вызывается, когда пользователь начинает тянуть планету. |
| OnDrag | public | | void | | PointerEventData data | | Вызывается, когда пользователь тянет планету. |
| OnEndDrag | public | | void | | PointerEventData eventData | | Вызывается, когда пользователь заканчивает тянуть планету. |
| Awake | private | | void | | - | | Выполняется при загрузке скрипта, находит текущую планету и корзины. |

Таблица 5.4

Описание полей, методов и свойств класса ExitController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| yes | | private | | GameObject | | Кнопка выхода из игры. | |
| no | | private | | GameObject | | Кнопка выхода в главное меню. | |
| back | | private | | GameObject | | Кнопка возвращения в главное меню. | |
| staticData | | private | | GameObject | | Объект StaticData. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Awake | private | | void | | - | | Вызывается при загрузке скрипта. |
| BackClick | private | | void | | - | | Открывает главное меню. |
| YesClick | private | | void | | - | | Выходит из игры. |

Таблица 5.5

Описание полей, методов и свойств класса GameController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| all\_object | | private | | GravityController[] | | Все объекты на экране с гравитацией. | |
| planets | | private | | List<GameObject> | | Все планеты на экране. | |
| prefab | | public | | GameObject | | Модель планеты. | |
| fly | | private | | bool | | Показывает, летит ли ракета. | |
| buttonBack | | private | | GameObject | | Предыдущая планета в списке. | |
| buttonForward | | private | | GameObject | | Следующая планета в списке. | |
| nowPlanet | | private | | GameObject | | Текущая планета. | |
| buttonStart | | private | | GameObject | | Кнопка начала игры. | |
| nowRocket | | private | | GameObject | | Текущая ракета. | |
| winTable | | private | | GameObject | | Таблица, появляющаяся после выигрыша. | |
| buttonReturn | | private | | GameObject | | Кнопка повторения уровня. | |
| buttonMenu | | private | | GameObject | | Кнопка выхода в главное меню. | |
| buttonNextLevel | | private | | GameObject | | Кнопка загрузки следующего уровня. | |
| buttonBackLevel | | private | | GameObject | | Кнопка выхода в главное меню. | |
| rocketTable | | private | | GameObject | | Таблица с характеристиками ракеты. | |
| planetTable | | private | | TablePlanetController | | Таблица с характеристиками планеты. | |
| finish | | private | | InFinish | | Объект, в который надо прилететь для успешного завершения уровня. | |
| staticData | | private | | StaticData | | Объект StaticData. | |
| curPlanet | | private | | int | | Планета на курсоре. | |
| model | | private | | Model | | Объект Model. | |
| startFlag | | private | | bool | | Показывает, была ли начата игра. | |
| nowPlanetClicked | | private | | bool | | Показывает, была ли нажата текущая планета. | |
| rocketClicked | | private | | bool | | Показывает, была ли нажата ракета. | |
| canvas | | private | | GameObject | | Объект Canvas. | |
| nowMass | | private | | float | | Масса ракеты. | |
| nowLevel | | private | | Level | | Текущий уровень. | |
| F | | private | | float | | Сила для полета ракеты в текущий момент. | |
| mass | | private | | float | | Масса оставшихся газов в текущий момент. | |
| start | | private | | float | | Время запуска ракеты. | |
| planetCount | | private | | int | | Количество планет на экране. | |
| nowRecord | | private | | int | | Рекорд по количеству планет на экране для текущего уровня. | |
| nowStars | | private | | int | | Количесво набранных за уровень звёзд. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Start | private | | void | | - | | Вызывается в начале работы скрипта. |
| RocketTableClick | private | | void | | - | | Открывает и закрывает информацию о ракете. |
| MenuClick | private | | void | | - | | Открывает главное меню. |
| NextLevelClick | private | | void | | - | | Запускает следующий уровень. |
| NowPlanetClick | private | | void | | - | | Открывает и закрывает информацию о текущей планете. |
| UpdateCurPlanet | private | | void | | int id | | Обновляет внешний вид меню выбора планет |
| PrevClick | private | | void | | - | | Обрабатывает нажатие на предыдущую планету. |
| NextClick | private | | void | | - | | Обрабатывает нажатие на следующую планету. |
| StartClick | private | | void | | - | | Обрабатывает начало полета ракеты. |
| FixedUpdate | private | | void | | - | | Вызывается каждую фиксированную рамку кадра. |
| StopClick | private | | void | | - | | Запускает текущий уровень заново. |
| Save | private | | void | | - | | Сохраняет текущее расположение планет. |
| Win | private | | void | | - | | Вызывается при выигрыше игрока. |
| Update | private | | void | | - | | Вызывается раз в кадр, проверяет, финишировала ли ракета. |
| **Свойства** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| Fly | public | | bool | | get | | Свойство доступа к fly. |

Таблица 5.6

Описание полей, методов и свойств класса GravityController.cs

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | | |  | | | |  | |
| Имя | | Модификатор доступа | | | Тип | | | | Назначение | |
| can\_gravity | | private | | | List<GravityController> | | | | Список объектов, которые имеют гравитацию. | |
| all\_object | | private | | | GravityController[] | | | | Список всех игровых объектов. | |
| plan | | public | | | PlayObject | | | | Текущая планета. | |
| physicBody | | private | | | Rigidbody2D | | | | Текущий объект. | |
| GC | | private | | | GameController | | | | Объект GameController. | |
| trash | | private | | | Trash | | | | Корзина для удаления планет. | |
| canMoves | | private | | | bool | | | | Определяет, может ли двигаться объект. | |
| **Методы** | | |  | | |  | |  | | |
| Имя | Модификатор доступа | | | Тип | | | Аргументы | | | Назначение |
| OnDrag | private | | | void | | | PointerEventData data | | | Вызывается, когда пользователь тянет планету. |
| OnEndDrag | private | | | void | | | PointerEventData eventData | | | Вызывается, когда пользователь заканчивает тянуть планету. |
| Init | private | | | void | | | PlayObject planet | | | Инициализирует объект. |
| Start | private | | | void | | | - | | | Вызывается в начале работы скрипта. |
| Update | private | | | void | | | - | | | Вызывается раз в кадр. |

Таблица 5.7

Описание полей, методов и свойств класса Help.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Модификатор доступа | Тип | Назначение |
| text | public | string | Текст. |

Таблица 5.8

Описание полей, методов и свойств класса HelpController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| staticData | | private | | StaticData | | Объект StaticData. | |
| helpTextTable | | private | | GameObject | | Таблица для вывода текста. | |
| text | | private | | GameObject | | Поле для текста. | |
| stop | | private | | GameObject | | Кнопка перехода к следующей карточке. | |
| id | | private | | int | | Номер таблички с текстом. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Start | private | | void | | - | | Вызывается в начале работы скрипта. |
| PutTextAndWait | private | | void | | int id | | Вставляет нужный текст в таблицу. |
| StopClick | private | | void | | - | | Обновляет таблцу с текстом. |

Таблица 5.9

Описание полей, методов и свойств класса InFinish.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| finishFlag | | private | | bool | | Определяет, пересекла ли ракета финиш. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| OnTriggerEnter2D | private | | void | | Collider2D other | | Вызывается, когда какой-то предмет оказывается в зоне финиша. |
| **Свойства** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| FinishFlag | public | | bool | | get | | Свойство доступа к finishFlag. |

Таблица 5.10

Описание полей, методов и свойств класса Level.cs

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Модификатор доступа | Тип | Назначение |
| KeyName | public | string | Название уровня. |
| rocketSpeed | public | float | Скорость газовой струи. |
| gasMass | public | float | Масса газов. |
| gasSpeed | public | float | Ежесекудный расход массы газов. |
| record | public | int | Рекорд по количеству планет. |

Таблица 5.11

Описание полей, методов и свойств класса LevelController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| staticData | | private | | StaticData | | Объект StaticData. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Init | public | | void | | int levelId | | Инициализация. |
| LoadLevelClick | private | | void | | int lvl | | Загрузка уровня. |

Таблица 5.12

Описание полей, методов и свойств класса LevelSave.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| planets | | public | | List<PlayObject> | | Список планет. | |
| name | | public | | string | | Название уровня. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| SaveObj | public | | void | | string key | | Сохранение объекта. |
| LevelSave | public | | конструктор | | - | | Пустой конструктор. |
| LevelSave | public | | конструктор | | string key | | Конструктор с парметром. |

Таблица 5.13

Описание полей, методов и свойств класса LevelsController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| levelsContent | | private | | GameObject | | Контент уровня. | |
| level | | private | | GameObject | | Объект уровня. | |
| staticData | | private | | StaticData | | Объект StaticData. | |
| levelsButtons | | private | | List<GameObject> | | Список кнопок уровней. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Init | public | | void | | - | | Инициализация меню уровней. |
| Redraw | public | | void | | - | | Перерисовывает экран игрока. |
| BackClick | public | | void | | - | | Возвращает игрока в главное меню. |
| GoClick | public | | void | | - | | Продолжает игру с момента последнего захода в неё. |

Таблица 5.14

Описание полей, методов и свойств класса MenuController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| start | | private | | GameObject | | Кнопка перехода в меню уровней. | |
| help | | private | | GameObject | | Кнопка перехода в меню "Help". | |
| settings | | private | | GameObject | | Кнопка перехода в меню "Setting". | |
| exit | | private | | GameObject | | Кнопка перехода в меню "Exit". | |
| chooseLevel | | private | | LevelsController | | Контроллер уровней. | |
| model | | private | | Model | | Объект Model. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Awake | private | | void | | - | | Вызывается при загрузке скрипта. |
| StartClick | private | | void | | - | | Открывает меню уровней. |
| HelpClick | private | | void | | - | | Открывает меню "Help". |
| SettingsClick | private | | void | | - | | Открывает меню "Setting". |
| ExitClick | private | | void | | - | | Открывает меню "Exit". |

Таблица 5.15

Описание полей, методов и свойств класса Model.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| nowLevel | | private | | int | | Номер текущего уровня. | |
| save | | private | | Save | | Объект Save. | |
| levelMenu | | private | | bool | | Определяет, нужно ли открыть меню уровней. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Save | public | | void | | - | | Сохранение уровня. |
| Awake | public | | void | | - | | Вызывается, при загрузке скрипта. |
| Load | public | | void | | - | | Загрузка уровня. |
| **Свойства** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| NowLevel | public | | int | | get, set | | Свойство доступа к nowLevel. |
| \_Save | public | | Save | | get, set | | Свойство доступа к save. |
| LevelMenu | public | | bool | | get, set | | Свойство доступа к levelMenu. |

Таблица 5.16

Описание полей, методов и свойств класса PlayObject.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| G | | private | | float | | Гравитационная константа. | |
| mass | | public | | float | | Масса объекта. | |
| radius | | public | | float | | Радиус объекта. | |
| keyName | | public | | string | | Название объекта. | |
| keyNameHash | | public | | int | | Хэш имени объекта. | |
| sprite | | public | | string | | Название картинки для объекта. | |
| type | | public | | string | | Тип объекта. | |
| realMass | | public | | string | | Реальнаяя масса объекта. | |
| realRadius | | public | | string | | Реальный радиус объекта. | |
| curPosition | | public | | Vector3 | | Позиция объекта в пространстве. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| PlayObject | public | | конструктор | | PlayObject obj | | Конструктор. |
| PlayObject | public | | конструктор | | float mass, float radius, string keyName | | Конструктор. |
| PlayObject | public | | конструктор | | string key | | Конструктор. |
| PlayObject | public | | конструктор | | - | | Конструктор. |
| Gravity | public | | float | | PlayObject first, PlayObject second, float R | | Расчёт гравитационной силы для двух игровых объектов. |
| SaveObj | public | | void | | string key | | Сохранение объекта. |
| Init | public | | void | | - | | Инициализация объекта. |

Таблица 5.17

Описание полей, методов и свойств класса RocketController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| staticData | | private | | StaticData | | Объект StaticData. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Start | private | | void | | - | | Вызывается в начале работы скрипта. |

Таблица 5.18

Описание полей, методов и свойств класса Save.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| levels | | public | | List<LevelSave> | | Список уровней. | |
| staticData | | private | | StaticData | | Объект StaticData. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Save | public | | конструктор | | - | | Конструктор. |
| Save | public | | конструктор | | string key | | Конструктор с параметром. |
| SaveObj | public | | void | | string key | | Сохранение объекта. |

Таблица 5.19

Описание полей, методов и свойств класса SettingsController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| on | | private | | GameObject | | Кнопка включения музыки. | |
| off | | private | | GameObject | | Кнопка выключения музыки. | |
| first | | private | | GameObject | | Кнопка выбора первого вида ракеты. | |
| second | | private | | GameObject | | Кнопка выбора второго вида ракеты. | |
| back | | private | | GameObject | | Кнопка возвращения в главное меню. | |
| staticData | | private | | GameObject | | Объект StaticData. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Awake | public | | void | | - | | Вызывается при загрузке скрипта. |
| BackClick | public | | void | | - | | Открывает главное меню. |
| OnClick | public | | void | | - | | Включает музыку. |
| OffClick | public | | void | | - | | Выключает музыку. |
| FirstClick | public | | void | | - | | Меняет внешний вид ракеты на первый вариант. |
| SecondClick | public | | void | | - | | Меняет внешний вид ракеты на второй вариант. |

Таблица 5.20

Описание полей, методов и свойств класса SettingsController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| planets | | public | | Planets | | Объект Planets. | |
| levels | | public | | Levels | | Объект Levels. | |
| texts | | public | | HelpText | | Объект HelpText. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| LoadResourceTextfile | public | | string | | string name | | Загружает текстовый файл. |
| GetSpriteByKeyName | public | | Sprite | | string keyName | | Получает Sprite по названию. |
| GetLevelByKeyName | public | | Level | | string keyName | | Получает уровень по названию. |
| GetNextPlanet | public | | int | | int curPlanet | | Получает следующую планету. |
| GetBackPlanet | public | | int | | int curPlanet | | Получает предыдущую планету. |
| GetPlanetByKeyName | public | | PlayObject | | string key\_name | | Получает планету по названию. |
| Awake | public | | void | | - | | Инициализация объектов из таблицы. |

Таблица 5.21

Описание полей, методов и свойств класса Levels

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Модификатор доступа | Тип | Назначение |
| list | public | List<Level> | Список уровней. |

Таблица 5.22

Описание полей, методов и свойств класса Planets

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Модификатор доступа | Тип | Назначение |
| list | public | List<PlayObject> | Список планет. |

Таблица 5.23

Описание полей, методов и свойств класса HelpText

|  |  |  |  |
| --- | --- | --- | --- |
| **Поля** |  |  |  |
| Имя | Модификатор доступа | Тип | Назначение |
| list | public | List<Help> | Список текстов. |

Таблица 5.24

Описание полей, методов и свойств класса TablePlanetController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| radius | | private | | Text | | Радиус планеты. | |
| mass | | private | | Text | | Масса планеты. | |
| name | | private | | Text | | Название планеты. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Redraw | public | | void | | PlayObject planet | | Перерисовка таблицы. |
| Awake | private | | void | | - | | Вызывается при загрузке скрипта. |

Таблица 5.25

Описание полей, методов и свойств класса TableRocketController.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| staticData | | private | | StaticData | | Объект StaticData. | |
| model | | private | | Model | | Объект Model. | |
| nowLevel | | private | | Level | | Текущий уровень. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Start | private | | void | | - | | Вызывается в начале работы скрипта. |

Таблица 5.26

Описание полей, методов и свойств класса Trash.cs

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Поля** | |  | |  | |  | |
| Имя | | Модификатор доступа | | Тип | | Назначение | |
| isMouseOver | | private | | bool | | Определяет, находится ли мышка в корзине. | |
| canvas | | private | | GameObject | | Объект Canvas. | |
| buttonTrash | | private | | GameObject | | Корзина для удаления игровых объектов. | |
| **Методы** | |  | |  | |  | |
| Имя | Модификатор доступа | | Тип | | Аргументы | | Назначение |
| Awake | private | | void | | - | | Вызывается при загрузе скрипта. |
| ClearAll | private | | void | | - | | Очистка игрового поля. |
| Update | private | | void | | - | | Вызывается раз в кадр. |
| **Свойства** |  | |  | |  | |  |
| Имя | Модификатор доступа | | Тип | | Доступ | | Назначение |
| IsMouseOver | public | | bool | | get | | Свойство доступа к isMouseOver. |

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |