**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Руководитель департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С.М. Авдошин  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 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 «События» («Приложение активности и уведомления ВШЭ»).

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](#_Toc8069927)

[1.1. Наименование программы 5](#_Toc8069928)

[1.2. Документы, на основании которых ведется разработка 5](#_Toc8069929)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 6](#_Toc8069930)

[2.1. Назначение программы 6](#_Toc8069931)

[2.1.1. Функциональное назначение 6](#_Toc8069932)

[2.1.2. Эксплуатационное назначение 6](#_Toc8069933)

[2.2. Краткая характеристика области применения 6](#_Toc8069934)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 7](#_Toc8069935)

[3.1. Постановка задачи на разработку программы 7](#_Toc8069936)

[3.2. Описание алгоритма и функционирования программы 7](#_Toc8069937)

[3.2.1. Описания алгоритма работы взаимодействия сервера и клиента 7](#_Toc8069938)

[3.2.2. Описания алгоритма взаимодействия экранов программы 7](#_Toc8069939)

[3.2.3. Описание работы сервера 8](#_Toc8069940)

[3.2.4. Описание алгоритма загрузки и обновления мероприятия 9](#_Toc8069941)

[3.2.5. Описания алгоритма сохранения данных 9](#_Toc8069942)

[3.2.6. Описание поиска мероприятий 11](#_Toc8069943)

[3.3. Описание и обоснование выбора метода организации входных и выходных данных 11](#_Toc8069944)

[3.3.1. Описание метода организации входных и выходных данных 11](#_Toc8069945)

[3.3.2. Обоснования выбора метода организации входных и выходных данных 12](#_Toc8069946)

[3.4. Описание и обоснование выбора состава технических и программных средств 12](#_Toc8069947)

[3.4.1. Состав технических и программных средств 12](#_Toc8069948)

[3.4.2. Обоснование выбора технических и программных средств 12](#_Toc8069949)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 13](#_Toc8069950)

[4.1. Предполагаемая потребность 13](#_Toc8069951)

[4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 13](#_Toc8069952)

[ПРИЛОЖЕНИЕ 1 14](#_Toc8069953)

[ТЕРМИНОЛОГИЯ 14](#_Toc8069954)

[ПРИЛОЖЕНИЕ 2 15](#_Toc8069955)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 15](#_Toc8069956)

[ПРИЛОЖЕНИЕ 3 16](#_Toc8069957)

[ПРАВИЛА ИГРЫ 16](#_Toc8069958)

[ПРИЛОЖЕНИЕ 4 17](#_Toc8069959)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ 17](#_Toc8069960)

[ПРИЛОЖЕНИЕ 5 18](#_Toc8069961)

[ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ ПОЛЕЙ, МЕТОДОВ И СВОЙСТВ 18](#_Toc8069962)

# ВВЕДЕНИЕ

## Наименование программы

Наименование программы – «Приложение активности и уведомления ВШЭ» или «HSE Activities and Notifications App».

Наименование программы для пользователя – «События».

## 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

Функциональным назначением программы является мобильное приложение, помогающее в организации мероприятий НИУ ВШЭ. Программа может как создавать и редактировать мероприятия, так и регистрировать на них участников.

### Эксплуатационное назначение

Функциональным назначением программы является мобильное приложение, помогающее в мониторинге мероприятий НИУ ВШЭ. Программа может искать и добавлять мероприятия в избранное, так же предлагает выбрать язык и тему приложения.

## Краткая характеристика области применения

Приложение активности и уведомления ВШЭ может использоваться как вспомогательный инструмент для мониторинга мероприятий НИУ ВШЭ с использованием мобильного устройства.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## Постановка задачи на разработку программы

Разрабатываемая программа должна соответствовать:

- описанным в техническом задании функциональным характеристикам («Приложение активности и уведомления ВШЭ». Техническое задание)

- описанным в техническом задании требованиям к интерфейсу («Приложение активности и уведомления ВШЭ». Техническое задание)

- описанным в техническом задании требованиям к надежности («Приложение активности и уведомления ВШЭ». Техническое задание)

## Описание алгоритма и функционирования программы

### Описания алгоритма работы взаимодействия сервера и клиента

Программа реализована в виде клиент-серверного приложения.

Клиентское приложение реализовывает следующие функции:

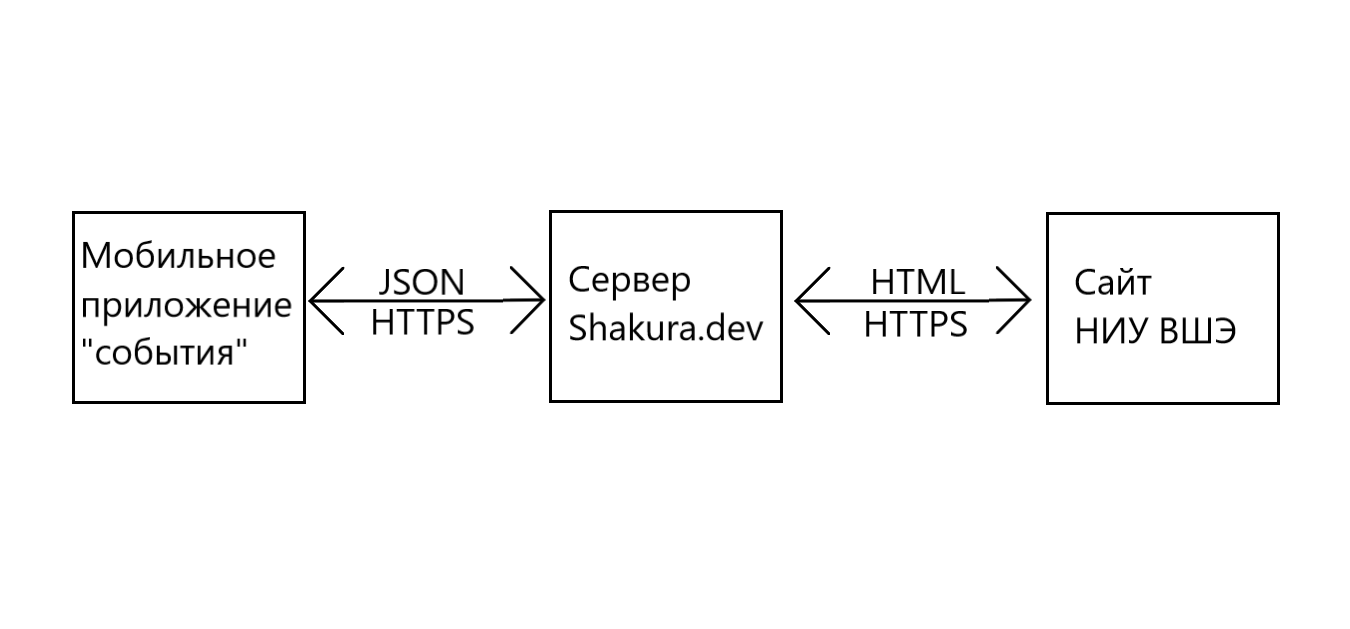
- отправлять запрос серверу на получение информации о мероприятиях;

- предоставляет функционал для работы с мероприятиями;

Серверное приложение должно реализовывать следующие функции:

- раз в час обновлять информацию о мероприятиях в соответствии с сайтом НИУ ВШЭ.

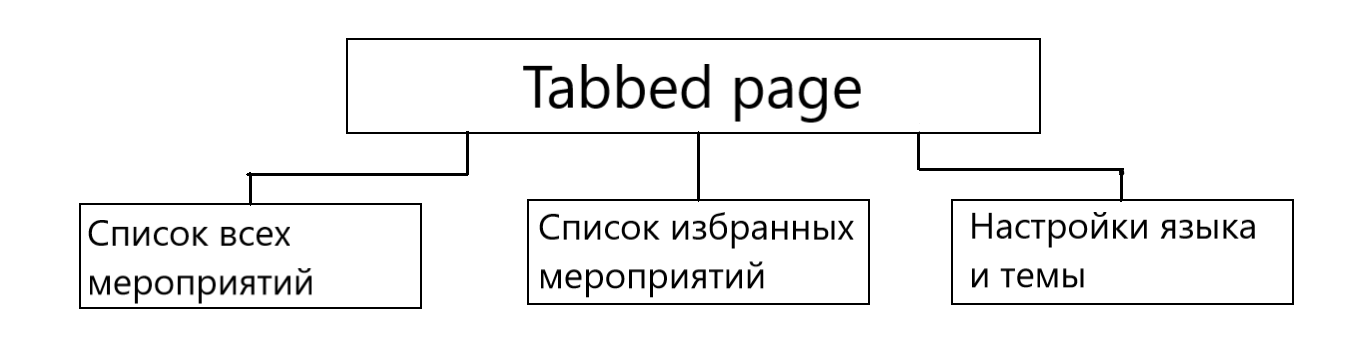
- иметь программный интерфейс приложения для взаимодействия с клиентскими приложениям



*Рисунок 1.* Обмен данными между клиентом, сервером и

### Описания алгоритма взаимодействия экранов программы

Алгоритм взаимодействия экранов программы приведен на рис. 2:



*Рисунок 2.* Взаимодействие экранов программы

### Описание работы сервера

Раз в час сервер делает запрос к сайтам <https://www.hse.ru/news/announcements/> и <https://www.hse.ru/en/news/announcements/>, затем парсит данные о мероприятиях в JSON.

from flask import json

def hsepars(a):

Events1 = a.findAll("div", {"class": "b-events"})

Announcements = []

for e1 in Events1:

Day = e1.findAll("div", {"class": "b-events\_\_title"})[0].getText()

# Announcements.append({"Type": "Day", "Day": Day})

Events = e1.findAll("div", {"class": "b-events\_\_item js-events-item"})

for e in Events:

Header = e.findAll("div", {"class": "b-events\_\_body\_title large"})[0].getText()

Link = e.findAll("div", {"class": "b-events\_\_body\_title large"})[0].a['href']

PreTags = e.findAll("div", {"class": "tag-set smaller"})[0]

PreExtra = e.findAll("span", {"class": "grey small"})

Time = e.findAll("div", {"class": "b-events\_\_extra date"})[0].getText()

Tags = []

Extra = []

for tag in PreTags:

Tags.append(tag.getText())

for extra in PreExtra:

Extra.append(extra.getText())

Place = ""

try:

Place = e.findAll("p", {"class": "b-text-grey small"})[0].getText()

except:

pass

Announcements.append(

{"Type": "Event", "Header": Header, "Tags": Tags, "Link": Link, "Place": Place, "Extra": Extra,

"Time": Time, "Day": Day})

return json.dumps(Announcements)

### Описание алгоритма загрузки и обновления мероприятий

Обращение к серверу и последующая загрузка мероприятий происходит при запуске приложения и при выполнения жеста обновления списка:

namespace App5.Services

{

public class MockDataStore : IDataStore<Item>

{

List<Item> items;

static public string Get(string url)

{

HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);

request.AutomaticDecompression = DecompressionMethods.GZip | DecompressionMethods.Deflate;

try

{

using (HttpWebResponse response = (HttpWebResponse)request.GetResponse())

using (Stream stream = response.GetResponseStream())

using (StreamReader reader = new StreamReader(stream))

return reader.ReadToEnd();

}

catch

{

return null;

}

}

static public void RealoadData()

{

AppData.ru = Get("https://shakura.dev/hseapi");

AppData.en = Get("https://shakura.dev/hseapien");

AboutPage.f();

}

public MockDataStore()

{

Item[] data = JsonConvert.DeserializeObject<Item[]>(AppData.isrus ? AppData.ru : AppData.en);

items = data.OfType<Item>().ToList();

}

public async Task<IEnumerable<Item>> GetItemsAsync(bool forceRefresh = false)

{

return await Task.FromResult(items);

}

}

}

### Описание алгоритма сохранения данных

С помощью класса AppData в App5.Models в приложение выполняется сохранение настроек языка и темы, которое происходит после каждого их изменения.

namespace App5.Models

{

struct FileStruct

{

public bool isrus { set; get; }

public bool IsThemeWhite { set; get; }

public List<string> Links;

}

static class AppData

{

static public Action SettingChanged;

static public List<string> Links = new List<string>();

/// <summary>

/// Langs

/// </summary>

static public string ru;

static public string en;

static public bool isrus { set; get; } = true;

/// <summary>

/// Colors

/// </summary>

static public bool IsThemeWhite { set; get; } = true;

static public Color BackgroundColor { get => IsThemeWhite ? Color.White : Color.FromRgb(16, 16, 16); }

static public Color BarBackgroundColor { get => IsThemeWhite ? Color.White : Color.FromRgb(10, 10, 10); }

static public Color FrontColor { get => IsThemeWhite ? Color.Black : Color.White; }

static AppData()

{

SettingChanged += Load;

try

{

FileStruct a;

string json;

string path = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

string filePath = Path.Combine(path, "AppData.txt");

using (var file = File.Open(filePath, FileMode.Open, FileAccess.Read))

using (var strm = new StreamReader(file))

{

json = strm.ReadToEnd();

}

a = JsonConvert.DeserializeObject<FileStruct>(json);

isrus = a.isrus;

IsThemeWhite = a.IsThemeWhite;

Links = a.Links;

}

catch (Exception e){}

}

static void Load()

{

try

{

FileStruct a = new FileStruct() {isrus=isrus,

IsThemeWhite= IsThemeWhite,

Links = Links};

string json = JsonConvert.SerializeObject(a);

string path = Environment.GetFolderPath(Environment.SpecialFolder.Personal);

string filePath = Path.Combine(path, "AppData.txt");

using (var file = File.Open(filePath, FileMode.Create, FileAccess.Write))

using (var strm = new StreamWriter(file))

{

strm.Write(json);

}

}

catch (Exception e){}

}

}

}

### Описание поиска мероприятий

В списке всех мероприятий и избранных мероприятий есть поиск по названию, тегам, месту и времени проведения мероприятий. Список мероприятий изменяется при каждом изменение поля поиска:

async Task ExecuteSearchItems()

{

if (IsBusy)

return;

IsBusy = true;

string SearchQuery = ItemsPage.SearchQuery;

Items.Clear();

string Day = "";

foreach (var item in items)

{

if (item.Header.ToLower().

Contains(SearchQuery.ToLower()) ||

item.Description.ToLower().

Contains(SearchQuery.ToLower()) ||

item.Place.ToLower().

Contains(SearchQuery.ToLower())||

item.Date.ToLower().

Contains(SearchQuery.ToLower()))

{

if (item.Day != Day)

{

Items.Add(new Item() { Type = "Day", Day = Day = item.Day });

}

Items.Add(item);

}

}

IsBusy = false;

}

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

### Описание метода организации входных и выходных данных

Входные данные в программе – действия пользователя в приложении (нажатие на кнопку, свайпы). Приложение предоставляет возможность открыть ранее сохранённой и установленной версии в виде apk файла (для смартфона на платформе Android).

Промежуточные входные данные при сборке проекта – данные в формате JSON получаемые от сервера.

Выходные данные в программе – отклики приложения на действия пользователя.

В программе предусмотрена функция сохранения изменений настроек в виде данных типа AppData.

### 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

В состав технических средств должен входить мобильное устройство, включающее в себя:

1) не менее 1 Гб оперативной памяти;

2) свободная память на диске не менее 1 Гб;

3) монитор, разрешение не ниже 800х600 пикселей.

4) Устройства ввода\вывода (сенсорный экран).

Для сервера:

1) Операционная система Linux.

2) не менее 1 Гб оперативной памяти;

3) свободная память на диске не менее 1 Гб;

4) Устройства ввода\вывода.

5) Доступ к сети интернет.

### Обоснование выбора технических и программных средств

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

# ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

## Предполагаемая потребность

Программа для получения информации о мероприятиях. Будет полезна для получения новостей с сайта НИУ ВШЭ;

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

В рамках данной работы расчет экономической эффективности не предусмотрен.

# ПРИЛОЖЕНИЕ 1

## ТЕРМИНОЛОГИЯ

Ниже приведен список необходимых терминов для ознакомления.

**Android** - операционная система для смартфонов.

**JSON** - текстовый формат обмена данными, основанный на JavaScript.

**Десериализация** – процесс восстановления данных из формата JSON.

**Сервер –** компьютер с операционной системой, подключенной к сети интеренет.

**Ubuntu –** операционная система для вычислительных машин (в том числе серверов).

**Python –** язык программирования.

**C# –** язык программирования.

**Xamarin –** платформа для разработки мобильных приложений на языке C#.

# ПРИЛОЖЕНИЕ 2

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Xamarin documentation - <https://docs.microsoft.com/ru-ru/xamarin/>

# ПРИЛОЖЕНИЕ 3

## АЛГОРИТМ ПОЛЬЗОВАНИЯ

В нижнем меню пользователь может выбрать один из трех экранов:

1. Список всех мероприятий
2. Список избранных мероприятий
3. Настройки приложения

В первом окне отображается список всех мероприятий. В верхней части экрана доступна кнопка поиска по мероприятиям: по нажатию на нее становится доступа строка ввода, по содержания которой отфильтровываются лишние мероприятия. По клику на мероприятия открывается экран с подробной информацией о мероприятии. Так же появляется возможность открыть страницу мероприятия и добавить его в избранное.

Второе окно аналогично первому, только в списке мероприятий находятся только избранные.

В третьем окне находится список настроек: цветовой и языковой тем. Для удобства пользователя, каждое изменение настроек фиксируется в память телефона и автоматически применяется при следующем запуске.

# ПРИЛОЖЕНИЕ 4

## ОПИСАНИЕ И ФУНКЦИОНАЛЬНОЕ НАЗНАЧЕНИЕ КЛАССОВ

Таблица 4.1

Описание и функциональное назначение классов

|  |  |
| --- | --- |
| Класс | Назначение |
| AppData | Фиксирует все данные в программе. |
| Item | Является классом хранения данных мероприятия. |
| MockDataStore | Обновляет данные присылаемые сервером. |
| AboutViewModel | Модель для меню настроек. |
| BaseViewModel | Класс от которого наследуется все модели экранов. |
| ItemDetailViewModel | Модель экрана подробностей мероприятия. |
| ItemsViewModel | Модель списков мероприятий. |
| AboutPage | Контролирует экран настроек. |
| Favorite | Контролирует экран избранных мероприятий. |
| ItemDetailPage | Контролирует экран подробностей мероприятий. |
| ItemsPage | Контролирует экран списка всех мероприятий. |
| MainPage | Контролирует общий экран приложения. |
| App | Контролирует меню уровней. |

# 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 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |