**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Научный руководитель,  доцент департамента  программной инженерии факультета компьютерных наук,  канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Р.З.Ахметсафина  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. |  | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия»  профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **ANDROID-ГОЛОВОЛОМКА «ПЕРВЫЕ ВСТРЕЧНЫЕ»**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.507600-01 81 01-1-ЛУ** | | |
|  |  | |
| Исполнитель  студент группы БПИ 164  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / М.Р. Зиганшин /  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2017 г. | |
|  | | |
|  | |  |

**Москва 2017**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| УТВЕРЖДЕН  RU.17701729.507600-01 81 01-1-ЛУ | |  |  | |
| |  |  | | --- | --- | | ***Подп. и дата*** |  | | ***Инв. № дубл.*** |  | | ***Взам. инв. №*** |  | | ***Подп. и дата*** |  | | ***Инв. № подл*** |  | | **ANDROID-ГОЛОВОЛОМКА «ПЕРВЫЕ ВСТРЕЧНЫЕ»**  **Пояснительная записка**  **RU.17701729.507600-01 81 01-1**  **Листов 26** | | | |
|  | |  | |
|  | |
|  | | | |
| **Москва 2017** | | |  |

# **АННОТАЦИЯ**

В данном программном документе приведена пояснительная записка к программе  
«Программа управления курсором на основе распознавания жестов». Данный документ содержит разделы «Введение», «Назначение и область применения программы», «Технические характеристики», «Ожидаемые технико-экономические показатели», «Источники, использованные при разработке»

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 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 [3];  
2) ГОСТ 19.102-77 Стадии разработки [4];  
3) ГОСТ 19.103-77 Обозначения программ и программных документов [5];  
4) ГОСТ 19.104-78 Основные надписи [6];  
5) ГОСТ 19.105-78 Общие требования к программным документам [7];  
6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом  
[8];  
7) ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению [9].  
Изменения к Пояснительной записке оформляются согласно ГОСТ 19.603-78 [10], ГОСТ  
19.604-78 [11].

**СОДЕРЖАНИЕ**

[АННОТАЦИЯ 2](#_Toc483242788)

[1. ВВЕДЕНИЕ 4](#_Toc483242789)

[1.1. Наименование программы 4](#_Toc483242790)

[1.2. Документы, на основании которых ведется разработка 4](#_Toc483242791)

[2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 5](#_Toc483242792)

[2.1 Функциональное назначение 5](#_Toc483242793)

[2.2 Эксплуатационное назначение 5](#_Toc483242794)

[3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_Toc483242795)

[3.1. Постановка задачи 6](#_Toc483242796)

[3.2. Описание алгоритма и функционирования программы 6](#_Toc483242797)

[3.2.1 Алгоритм решения головоломки по входным данным 6](#_Toc483242798)

[3.2.2 Алгоритм создания случайного игрового поля 10](#_Toc483242799)

[3.3. Описание и обоснование выбора состава технических и программных средств 10](#_Toc483242800)

[3.3.1 Состав технических и программных средств 10](#_Toc483242801)

[3.3.2 Обоснование выбора состава технических и программных средств 10](#_Toc483242802)

[4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 12](#_Toc483242803)

[4.1. Предполагаемая потребность 12](#_Toc483242804)

[4.2. Экономические преимущества разработки по сравнению с отечественными и зарубежными образцами или аналогами 12](#_Toc483242805)

[5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 13](#_Toc483242806)

[Приложение 1 14](#_Toc483242807)

[Описание и функциональное назначение классов 14](#_Toc483242808)

[Приложение 2 16](#_Toc483242809)

[Описание и функциональное назначение полей, методов и свойств 16](#_Toc483242810)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 26](#_Toc483242811)

# 1. ВВЕДЕНИЕ

**1.1.** **Наименование** **программы**

Наименование программы – «Программа управления курсором на основе распознавания жестов» («Cursor Control Using Hand Gesture Recognizing»).

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

Приказ НИУ ВШЭ Приказ декана ФКН И.В. Аржанцева № ХХХХХХХ от ХХ.ХХ.2017 "ХХХХХХХХХ". Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук.

# 2. НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

**2.1** **Функциональное** **назначение**

Функциональным назначением является считывание и обработка изображения с веб-камеры, распознавание местоположения и жеста руки. Курсор мыши перемещается по экрану следуя за указательным пальцем пользователя. В случае отгибания также и большого пальца производится нажатие левой кнопки мыши, в случае еще и среднего - правой.

**2.2** **Эксплуатационное** **назначение**

Программа может быть использована обычным пользователем для управления компьютером в случае отсутствия мыши

# 3. ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

**3.1.** **Постановка задачи**

* Программа должна считывать изображения с веб-камеры компьютера
* Программа должна определять местоположение руки пользователя
* Программа должна распознавать жест руки пользователя

1. Указательный палец
2. Большой и указательный палец
3. Большой, указательный и средний палец

* В зависимости от жестов, указанных выше соответственно, программа должна выполнять определённые действия:

1. Перемещение курсора по экрану (Курсор должен следовать за движением руки)
2. Нажатие левой кнопки мыши
3. Нажатие правой кнопки мыши

**3.2.** **Описание алгоритма и функционирования программы**

**3.2.1** **Алгоритм обработки сырого изображения**

Перед началом работы программы сохраняется изображение заднего фона.

Изображение с камеры будет обрабатываться покадрово и каждый кадр будет проходить несколько этапов обработки изображения:

1. Bilateral filter (двусторонняя фильтрация) – алгоритм для устранения теней и шума с изображения. В отличии от большого количества подобных алгоритмов, двусторонняя фильтрация оставляет углы изображения острыми. Этот алгоритм учитывает окружение пикселя и высчитывает среднее значение цвета.

Поскольку изображение берется с веб-камеры, которые обычно не обладают высоким качеством съемки, данный фильтр применяется для того чтобы убрать дефекты с изображения.

1. После того как с изображения были убраны мелкие шумы и дефекты выполняется вычитание заднего фона, который был сохранен заранее. Таким образом получается изображение переднего плана. Однако это изображение все равно довольно плохого качества и требуется применить несколько других алгоритмов обработки изображения.
2. Первый из них - Erosion (размытие) – одна из фундаментальных операций в обработке изображений. Основная идея размытия – исследование изображения с помощью заранее определенного ядра. Ядро само является изображением. Ядро накладывается на каждый пиксель изображения и затем вычисляется локальный минимум из всех пикселей под ядром.
3. Gaussian blur (размытие Гаусса) – фильтр для размытия изображения, который использует нормальное распределение для вычисления новых значений для цвета пикселей.
4. После того как изображение полностью обработано выделяется белые и черные области на изображении. Для этого используются алгоритм threshold. Threshold – способ обработки изображений, который применяется к черно-белым изображениям. Его суть состоит в том, что каждый пиксель сравнивается с порогом. Если пиксель больше порога то пиксель становится белым, иначе черным.

**3.2.2** **Алгоритм распознавание жеста**

Поле того как было получено изображение руки на черном фоне необходимо распознать жест и выполнить необходимое действие.

1. Сначала необходимо построить выпуклую оболочку вокруг изображения руки. Для этого применяется алгоритм построения контуров (Drawing Contours). Алгоритм строит линии на переходе из белую в черную область.

**3.3.** **Описание и обоснование выбора состава технических и программных средств**

**3.3.1** **Состав технических и программных средств**

Для работы программы необходим следующий набор программных средств:

1. Операционная система Android 2.3.1 или более поздняя

Для работы программы необходим следующий набор технических средств:

1. Сенсорный дисплей
2. Не менее 65 МБ свободной памяти

**3.3.2**  **Обоснование выбора состава технических и программных средств**

При создании графического интерфейса программы использовался Unity 5.5.1f1, поэтому программа может работать только на устройствах с уровнем API выше 9, а значит минимальная версия Android, которая должна быть на устройстве – 2.3.1.

# 4. ОЖИДАЕМЫЕ ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

**4.1.** **Предполагаемая потребность**

В наше время мобильные игры являются популярным видом времяпрепровождения. Игра-головоломка “Первые встречные” может быть использована любыми пользователями для развлечения.

**4.2.** **Экономические преимущества разработки по сравнению с отечественными и  
зарубежными образцами или аналогами**

Аналогов программы на Android найдено не было. Существует несколько веб-версий, но во всех используются заранее заготовленные поля, отсутствует генерация случайного поля.

# 5. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Knuth D. E. Dancing Links / Millenial Perspectives in Computer Science, 2000, (c. 187-214) – 416 с
2. Kunigami G. The Algorithm X and the Dancing Links [Электронный ресурс] / Режим доступа: https://kunigami.wordpress.com/2013/04/28/the-algorithm-x-and-the-dancing-links/, свободный. (Дата обращения 19.03.16)
3. ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
4. ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
5. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система  
   программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
6. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
7. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
8. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
9. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
10. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной  
    документации. – М.: ИПК Издательство стандартов, 2001. – 126 с
11. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные  
    печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001. – 126 с

# Приложение 1

## **Описание и функциональное назначение классов**

Таблица 1 – Описание и функциональное назначение классов

|  |  |
| --- | --- |
| Класс | Назначение |
| Matrix | Класс, отвечающий за составление матрицы входных данных. |
| Dancing Links | Класс, отвечающий за составление структуры данных и содержащий алгоритм Dancing Links для решения головоломки. |
| Node | Класс, описывающий элементы в структуре данных |
| Head | Класс, описывающий элементы в верхней строке структуры данных. |
| Gen | Класс, отвечающий за случайную генерацию поля. |
| Back | Класс (скрипт в Unity), отвечающий за переход на начальный экран |
| back2 | Класс (скрипт в Unity), отвечающий за переход на экран выбора поля |
| buttons | Класс (скрипт в Unity), отвечающий за кнопки в приложении и переходы между сценами |
| check4x4 | Класс (скрипт в Unity), отвечающий за проверку введенного пользователем решения |
| chooseLanguge | Класс (скрипт в Unity), отвечающий за смену языка правил |
| ChooseText | Класс (скрипт в Unity), отвечающий за выбор надписи в зависимости от языка и выигрыша или проигрыша |
| GenarateField | Класс (скрипт в Unity), отвечающий за генерацию поле и начала отсчета времени, затраченного на решения |
| Home | Класс (скрипт в Unity), отвечающий за переход на начальный экран |
| Letter2Tap | Класс (скрипт в Unity), отвечающий за обработку нажатия на клетки по краю поля |
| LetterTap | Класс (скрипт в Unity), отвечающий за обработку нажатия на клетки поля |
| nextRules | Класс (скрипт в Unity), отвечающий за переход к следующей странице правил |
| NextStats | Класс (скрипт в Unity), отвечающий за переход к следующей странице статистики |
| script4x4 | Класс (скрипт в Unity), отвечающий за заполнение букв по краю игрового поля |
| ShowStats | Класс (скрипт в Unity), отвечающий за вывод статистики |
| ShowTime | Класс (скрипт в Unity), отвечающий за вывод времени и лучшего времени |
| SolveScript | Класс (скрипт в Unity), отвечающий за решение головоломки по заданным пользователем входным данным |
| DLScript | Класс (скрипт в Unity), хранящий информацию о всех массивах букв по краю и размер поля |

# Приложение 2

## **Описание и функциональное назначение полей, методов и свойств**

Таблица 2.1 – описание полей, методов и свойств класса Matrix

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Matrix | | | | | |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| N | public | int | Размер поля | | |
| Data | int[,] | Матрица входных данных | | |
| additionalColumn | int[,] | Дополнительные столбцы для составления матрицы | | |
| additionalRow | int[,] | Дополнительные строки для составления матрицы | | |
| candidatesCount | int | Количество столбцов | | |
| requestsCount | int | Количество строк | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| LetterCount | private | int | | char[] a | Подсчет количества всех букв по бокам |
| Matrix | public | конструктор | | int N, int numberOfLetters,  char[] upperRow,  char[] leftColumn,  char[] lowerRow,  char[] rightColumn | Конструктор матрицы входных данных |

Таблица 2.2 – описание полей, методов и свойств класса DancingLinks

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DancingLinks | | | | | |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| basic | public | Matrix | Матрица входных данных | | |
| countSolutions | public | int | Количество решений | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| random | private | Random | Random, необходимый для генерации случайного числа | | |
| root | private | Head | Корень структуры данных | | |
| data | private | Node[,] | Структура данных | | |
| head | private | Head[] | Строка сверху структуры данных | | |
| solution | private | Node[] | Одно решение | | |
| solutions | private | Solutions[] | Все решения | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| DancingLinks | public | конструктор | | Matrix | Конструктор структуры данных |
| Remove | public | void | | Head h | Удаление столбца |
| Restore | public | void | | Head h | Восстановление столбца |
| Solver1 | public | bool | | - | Поиск одного верного решения |
| Solver2 | public | bool | | - | Поиск всех верных решений |
| Solver3 | public | bool | | - | Поиск случайного верного решения |
| ShortestHead | public | Head | | - | Поиск столбца с минимальным количеством элементов |
| RandomShortestHead | public | Head | | - | Поиск случайного столбца с минимальным количеством решений |
| OutSolution | public | char[,] | | - | Перевод решения в матрицу символов |
| OutSolutions | public | char[][,] | | - | Перевод всех решений в массив матриц символов |

Таблица 2.3 – описание полей, методов и свойств класса Node

|  |  |  |  |
| --- | --- | --- | --- |
| Node | | | |
| **Свойства** | | | |
| Имя | Модификатор доступа | Тип | Описание |
| right | public | Node | Ссылка на элемент справа |
| left | Node | Ссылка на элемент слева |
| up | Node | Ссылка на элемент сверху |
| down | Node | Ссылка на элемент снизу |
| header | Head | Количество столбцов |
| rowNumber | int | Номер строки |

Таблица 2.4 – описание полей, методов и свойств класса Head

|  |  |  |  |
| --- | --- | --- | --- |
| Head | | | |
| **Свойства** | | | |
| Имя | Модификатор доступа | Тип | Описание |
| rightHead | public | Head | Ссылка на элемент справа |
| leftHead | Head | Ссылка на элемент слева |
| rowCount | int | Количество элементов в столбце |
| columnNumber | int | Номер столбца |

Таблица 2.5 – описание полей, методов и свойств класса Gen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Gen | | | | | |
| **Свойства** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| N | public | int | Размер игрового поля | | |
| numberOfLetters | public | int | Количество всех возможных букв | | |
| Field | public | char[,] | Сгенерированное игровое поле | | |
| upperRow | public | char[] | Массив букв сверху поля | | |
| leftColumn | public | char[] | Массив букв слева поля | | |
| lowerRow | public | char[] | Массив букв снизу поля | | |
| rightColumn | public | char[] | Массив букв справа поля | | |
| OutSolution | public | char[][,] | Массив всех решений | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| random | private | Random | Random, необходимый для генерации случайного числа | | |
| letters | private | Letter[] | Массив всех букв по бокам | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| Gen | public | конструктор | | int N, int numberOfLetters | Конструктор игрового поля |
| Mix | public | void | | Letter[] letter | Перемешивание букв в случайном порядке |
| DeleteLetters | public | void | | - | Удаление букв по краю |

Таблица 2.6 – описание полей, методов и свойств класса Back

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Back | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| OnMouseUpAsButton | private | void | - | Обработка нажатия на обьект и переход на начальный экран |

Таблица 2.7 – описание полей, методов и свойств класса back

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| back2 | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| OnMouseUpAsButton | private | void | - | Обработка нажатия на обьект и переход к выбору размера поля |

Таблица 2.8 – описание полей, методов и свойств класса buttons

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| buttons | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| layerBlue | public | Sprite | Картинка на кнопке в ненажатом состоянии | | |
| layerDark | Sprite | Картинка на кнопке в нажатом состоянии | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| OnMouseDown | private | void | | - | Обработка нажатия кнопки, смена картинки |
| OnMouseUp | private | void | | - | Обработка отпускания кнопки, смена картинки |
| OnMouseUpAsButton | private | void | | - | Обработка нажатой кнопки, переход на другую сцену |

Таблица 2.9 – описание полей, методов и свойств класса check4x4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| check4x4 | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| ready | public | Sprite | Картинка на кнопке в ненажатом состоянии | | |
| readyActive | Sprite | Картинка на кнопке в нажатом состоянии | | |
| CanvasField | GenerateField | Поле, которое заполнил игрок | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| Start | private | void | | - | Заполнение текста на кнопке в зависимости от языка |
| OnMouseDown | private | void | | - | Обработка нажатия кнопки, смена картинки |
| OnMouseUpAsButton | private | void | | - | Обработка нажатой кнопки, переход на другую сцену |

Таблица 2.10 – описание полей, методов и свойств класса chooseLanguge

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| chooseLanguge | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| rules0 | public | Text | Английский текст | | |
| rules1 | Text | Русский текст | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| Start | private | void | | - | Вывод правил в зависимости от языка |

Таблица 2.11 – описание полей, методов и свойств класса ChooseText

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ChooseText | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Start | private | void | - | Вывод надписи в зависимости от выигрыша или проигрыша и языка |

Таблица 2.12 – описание полей, методов и свойств класса GenerateField

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| GenerateField | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| N | public | int | Размер поля | | |
| numberOfLetters | int | Количество всех | | |
| Field | Gen | Сгенерированное игровое поле | | |
| userField | char[,] | Поле в виде матрицы символов | | |
| currentTime | float | Время начала игры | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| Awake | private | void | | - | Генерация поля, начало отсчета времени |

Таблица 2.13 – описание полей, методов и свойств класса Home

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Home | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| OnMouseUpAsButton | private | void | - | Обработка нажатия на обьект и переход на начальный экран |

Таблица 2.14 – описание полей, методов и свойств класса Letter2Tap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Letter2Tap | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| OnMouseUpAsButton | private | void | - | Обработка нажатия на объект, смена буквы в клетки |

Таблица 2.15 – описание полей, методов и свойств класса LetterTap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LetterTap | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Описание |
| OnMouseUpAsButton | private | void | - | Обработка нажатия на объект, смена буквы в клетки поля |

Таблица 2.16 – описание полей, методов и свойств класса nextRules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| nextRules | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| previos | public | Text | Предыдущие правила | | |
| next | Text | Следующие правила | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| OnMouseUpAsButton | private | void | | - | Обработка нажатия, переход к следующим правилам |

Таблица 2.17 – описание полей, методов и свойств класса nextRules

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| nextRules | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| previos | public | GameObject | Предыдущая статистика | | |
| next | GameObject | Следующая статистика | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| OnMouseUpAsButton | private | void | | - | Обработка нажатия, переход к следующей статистике |

Таблица 2.18 – описание полей, методов и свойств класса check4x4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| check4x4 | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| Letter2 | public | Sprite | Картинка на кнопке в ненажатом состоянии | | |
| activeLetter | Sprite | Картинка на кнопке в нажатом состоянии | | |
| CanvasField | GenerateField | Поле, которое было сгенерировно | | |
| Letter | Text | Буква в клетке | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| Start | private | void | | - | Заполнение букв по краю |

Таблица 2.19 – описание полей, методов и свойств класса ShowStats

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ShowStats | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Start | private | void | - | Вывод статистики в зависимости от размера поля и языка |

Таблица 2.20 – описание полей, методов и свойств класса ShowTime

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ShowTime | | | | |
| **Методы** | | | | |
| Имя | Модификатор доступа | Тип | Аргументы | Назначение |
| Start | private | void | - | Вывод времени, которое затрачено на решение и лучшего решения |

Таблица 2.21 – описание полей, методов и свойств класса SolveScript

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SolveScript | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| ready | public | Sprite | Картинка на кнопке в ненажатом состоянии | | |
| readyActive | Sprite | Картинка на кнопке в нажатом состоянии | | |
| Canvas | DLScript | Все массивы букв по краю, размер поля | | |
| Field | GameObject | Объект, в котором содержатся все клетки поля | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| Start | private | void | | - | Заполнение текста на кнопке в зависимости от языка |
| OnMouseDown | private | void | | - | Обработка отпускания кнопки, смена картинки |
| OnMouseDown | private | void | | - | Обработка нажатия кнопки, смена картинки |
| OnMouseUpAsButton | private | void | | - | Обработка нажатой кнопки, заполнение  игрового поля |

Таблица 2.22 – описание полей, методов и свойств класса DLScript

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| DLScript | | | | | |
| **Поля** | | | | | |
| Имя | Модификатор доступа | Тип | Описание | | |
| N | public | int | Размер поля | | |
| numberOfLetters | int | Количество всех возможных букв | | |
| upperRow | char[] | Массив букв сверху | | |
| leftColumn | char[] | Массив букв слева | | |
| lowerRow | char[] | Массив букв снизу | | |
| rightColumn | char[] | Массив букв справа | | |
| **Методы** | | | | | |
| Имя | Модификатор доступа | Тип | | Аргументы | Назначение |
| Awake | private | void | | - | Заполнение начальных значений |

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

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