МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ

КАЛИНИНГРАДСКОЙ ОБЛАСТИ

ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ

«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И СТРОИТЕЛЬСТВА»

(ГБУ КО ПОО «КИТиС»)

|  |  |  |
| --- | --- | --- |
| Курсовой проект  допущен к защите  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (дата)  Зам.директора по УМР  \_\_\_\_\_\_\_\_\_\_\_\_\_Павленко Г.Я.  (подпись) |  | Курсовой проект  защищен с оценкой  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (оценка)  Руководитель работы  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) |

Пояснительная записка к курсовому проекту

по дисциплине: МДК 01.01 Разработка программных модулей

Тема Разработка приложения по логической игре «Эрудит»

Специальность 09.02.07 «Информационные системы и программирование»

|  |  |
| --- | --- |
| Выполнил: | студент 3 курса,  группы ИСп 19-1  Попов Ярослав Дмитриевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |
| Руководитель: | Большакова-Стрекалова Анна Викторовна, преподаватель ГБУ КО ПОО КИТиС  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) |

Калининград

2022

Содержание:

|  |  |
| --- | --- |
| Рецензия | 3 |
| Введение | 4 |
| 1 Описание предметной области | 5 |
| 1.1 Аналоги разрабатываемого приложения | 5 |
| 1.2 Техническое задание | 7 |
| 1.3 Описание структуры приложения | 7 |
| 2 Описание разработки приложения | 8 |
| 2.1 Обоснование средств разработки | 8 |
| 2.2 Разработка интерфейса | 10 |
| 2.3 Разработка логики работы приложения. Схема взаимодействия компонентов проекта | 16 |
| 2.4 Описание переменных, компонентов, классов и подпрограмм | 16 |
| 3 Тестирование и установка приложения | 23 |

Рецензия

В ходе тестирования данного приложения было выявлено, что приложение сделано просто и без анимаций. Смена окон приложения осуществляется путем нажатия кнопок.

Из минусов: нет сохранения прогресса, нет кнопок выхода в меню с самих уровней, а также малое количество предоставленных в данный момент уровень.

Из плюсов: приятный дизайн, мало багов.

Под итожив выше перечисленные доводы можно сказать, что приложение уже можно использовать, но стоит исправить перечисленные выше недостатки, тогда приложение станет значительно лучше.

Введение:

Из-за современных технологий существует возможность создавать огромное количество разного рода приложений, способствующих интересному или развивающему проведению досуга. Данный проект создан для проведения досуга с пользой, он позволит развить мышление и словарный запас, что в современном мире будет очень полезно.

Описание предметной области.

Актуальность данного проекта не особо велика, но все же некоторый спрос в этой теме присутствует, так как каждый человек хочет провести досуг так, как ему понравится.

По этой причине возникла идея создать приложения для проведения досуга с пользой и интересом. Данное приложение будет доступно людям почти с любой возрастной категорией.

Для использования данного приложения не будет особых требований. Единственным требованием будет лишь соответствие возраста для возможности использовать приложение и возможность установить приложение на устройство.

Аналоги.

Онлайн эрудит от Yandex.

Пример данного приложения представлен ниже на рисунке 1. Интерфейс данного приложения выглядит очень просто, но в тоже время абсолютно понятно какой элемент где находится. Версия 1.0. Общий рейтинг данного приложения составляет 3.2\5.

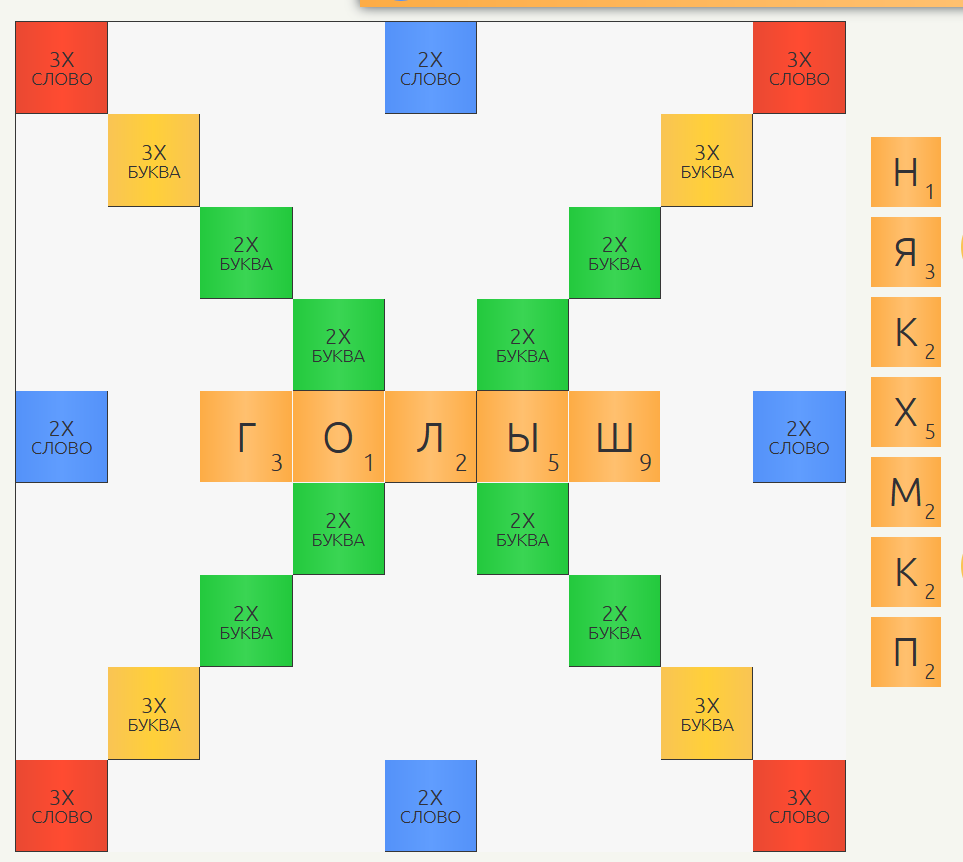


Рисунок 1.

Онлайн игра «Буквоплет»

Пример данного приложения представлен ниже на рисунке 2. Интерфейс приложения достаточно простой, но для нового игрока будет не понятно, что значат клетки разных цветов, что конечно является минусом. Так же хоть здесь и есть короткое обучение, но даже оно не проливает свет на ячейки разных цветов.

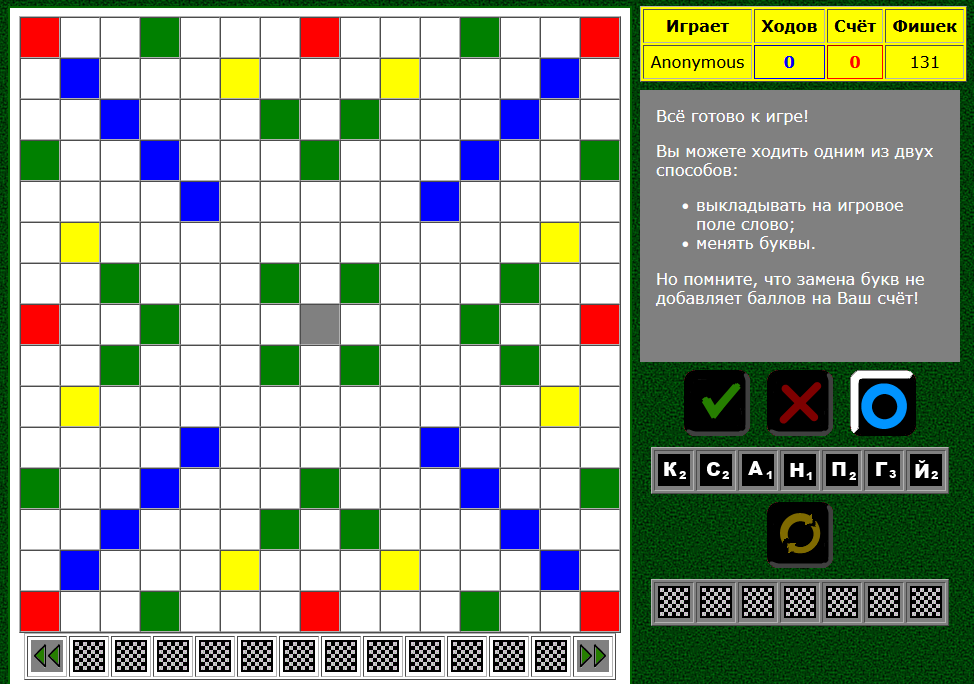


Рисунок 2.

Аналог «Scramble words» приложение, найденное на просторах интернета, пример которого представлен ниже на рисунке 3. Версия приложения 1. Интерфейс оформлен хоть и просто, но нельзя сказать, что он некрасивый. Он очень приятен глазу. Сама же игра полностью понятна даже без обучений.



Рисунок 3.

Техническое задание.

В данном приложение должно присутствовать следующее: Поле для подстановки букв. Поле для букв из которых будет составлено слово. При открытии приложения, должна отображаться главная форма. Несколько уровней в каждом из которых будет слово, которое раннее не было представлено. Всплывающее окно «победа» при правильно составленном слове. Дизайн должен быть унифицирован. Должен быть инстинктивно понятный интерфейс. После реализации приложения и составления всей документации, проект требуется выложить на репозиторий GitHub для её проверки руководителем.

Структура приложения.

Приложение состоит из кода, написанного на c#, главного меню, меню выбора уровня и самого экрана выбранного уровня, в котором будут поле для подстановки букв и поле для получения букв, которые требуется подставить в ячейки для получения слова.

Описание разработки приложения.

Обоснование средств разработки.

При реализации спроектированной системы необходимо выбрать средства разработки. Из целей разрабатываемой системы и требований технического задания можно выделить требования к среде разработки. Были выявлены следующие требования к среде разработки:

1. Поддержка объектно-ориентированного программирования.
2. Возможность построения графического интерфейса.

В настоящее время на рынке систем разработки существует большое количество разного рода средств разработки, у каждой из которых имеется индивидуальный интерфейс и методы программирования на разных языка. В ходе размышлений были выявлены следующие системы:

1. Microsoft Visual C++.
2. Unity C#.
3. Intelij Idea java.
4. Android studio.

Все они поддерживают объектно-ориентированное программирование. Объектно-ориентированное программирование (ООП) - это методика, позволяющая концентрировать основное внимание программиста на связях между объектами, а не на деталях их реализации. ООП - это совершенно новый подход к построению сложных программ и систем.

В объектно-ориентированном программировании и проектировании главной, отправной точкой является объект. Данный подход представляется как достаточно естественный, так как в реальном мире мы взаимодействуем именно с объектами (предметы, люди, разного рода технические устройства и так далее). Так же можно сказать и про взаимодействие программы и человека. Прикладная программа (приложение), построенная по принципам объектной ориентации – это не последовательность, каких-то операторов, а некий жесткий алгоритм. Объектно–ориентированное программирование - это совокупность объектов и способов взаимодействия. Обмен между объектами происходит посредством сообщений.

ООП стало одним из их основных способов при разработке разного рода программного обеспечения. В мире объектно-ориентированное программирование применяется применяется в огромном количестве областей, таких как банковские транзакции, управление коммунальным хозяйством и исследование генов человека. Во многих случаях новые поколения операционных систем, систем управления базами данных, телефонных служб, систем Авионики и мультимедиа-программ пишутся в объектно-ориентированном стиле. В большинстве таких проектов предпочли использовать объектно-ориентированную технологию просто потому, что не было другой возможности создать достаточно надежную и жизнеспособную систему.

Каждое из средств разработки имеет свои плюсы и минусы.

1. Visual C++.

Microsoft visual C++ - среда разработки приложений на языке C++. Разработана фирмой Microsoft поставляется как в комплекте, так и отдельно. Позволяет работать с внутренними функциями операционных систем серии Windows. Также C++ включает в себя встроенный отладчик низкого уровня, позволяющий проводить анализ работы написанной программы на уровне машинного кода.

1. Unity C#.

Unity среда разработки, созданная американской компанией Unity Technologies в 2005 году. Позволяет создавать приложения, работающие сразу на нескольких платформах, таких как мобильные телефоны, персональные компьютеры, игровые консоли, разного рода интернет приложения и так далее. Главными преимуществами unity являются: модульная система компонентов, межплатформенная поддержка и наличие визуальной среды разработки. Минусами же являются сложности при подключении внешних библиотек и появление сложностей при работе с многокомпонентными схемами.

1. Intelij Idea java.

Intelij idea – среда разработки поддерживающая многие языки программирования, в их число входят: java, javaScript, YAML, Groovy, Kotlin, C++, PHP, SQL и так далее, была выпущена в январе 2001 года компанией JetBrains. Среда хорошо совместима с многими свободными инструментами разработчиков. Имеет встроенный инструментарий, позволяющий разрабатывать графический пользовательский интерфейс.

1. Android studio

Android studio – среда разработки для приложений на платформе Android, была выпущена в мае 2013 компанией Google. Доступна для установки на Windows, MacOS, Linux. Имеет поддержку двух языков программирования, ими являются Java и Kotlin. Имеет различные виды сборок и генерации .apk файлов. Имеет встроенные шаблоны для макетов и компонентов Android. Позволяет разрабатывать приложения для любого устройства на Android включая Android Wear и Android TV.

После анализа всех вышеперечисленных доводов было принято решение начать разработку выбранного приложения в среде разработки, именуемой Unity используя такой язык программирования как C#.

Разработка интерфейса.

В данном случае для разработки интерфейса приложения не пришлось использовать что-то кроме выбранной среды разработки, т.к. выбранная среда уже имеет необходимый инструментарий для разработки интерфейса приложения.

Чтоб разработать интерфейс сначала нужно установить модули, которые будут присутствовать в данном приложении. В данном случае будут использованы следующие модули:

1. Модуль главного меню
2. Модуль выбора уровня
3. Модуль победного экрана
4. Модуль самого уровня

Далее разработка интерфейса и его готовый вариант будут описываться в соответствии с конкретно выбранным модулем и присутствующими в нем компонентами.

1. Модуль главного меню.

В данном модуле будет присутствовать несколько компонентов, а именно кнопка выхода из приложения, кнопка выбора уровня и текст с приветствием пользователя.

В верхней части экрана по центру будет располагаться текстовое поле, содержащее приветствие пользователя, а по центру, под полем приветсвия будут кнопка выбора уровня, и кнопка выхода из приложения соответственно. Далее на рисунке 4 будет представлен макет интерфейса главного меню.

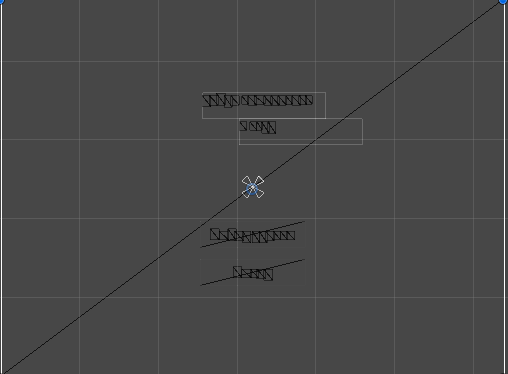


Рисунок 4.

Далее на рисунке 5 будет представлен итоговый интерфейс окна главного меню.

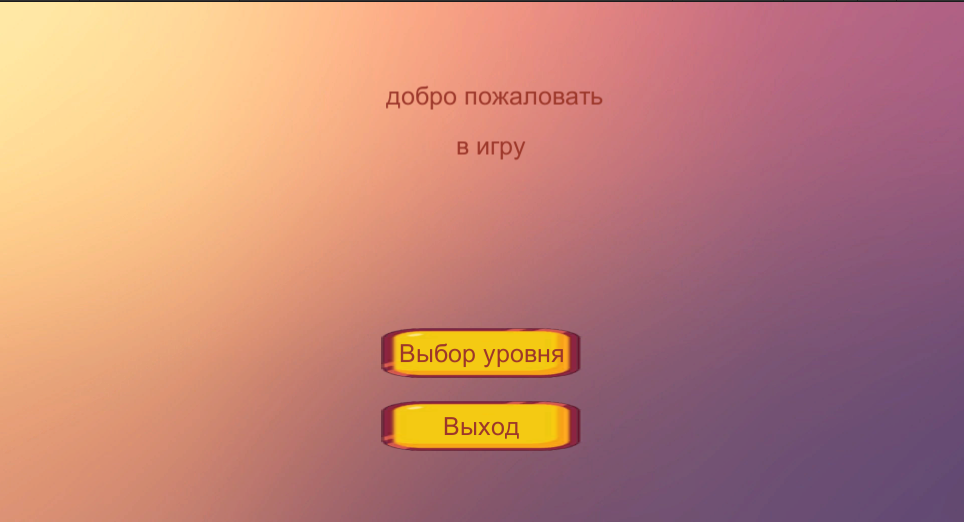


Рисунок 5.

1. Модуль выбора уровня.

В данном модуле будут представлены следующие компоненты: текстовое поле с предложением выбрать уровень из представленных ниже, кнопки, количество которых будет соответствовать количеству присутствующих в приложении уровней, а также кнопка, позволяющая вернуться обратно в главное меню. Далее на рисунке 6 будет представлен макет интерфейса данного модуля.

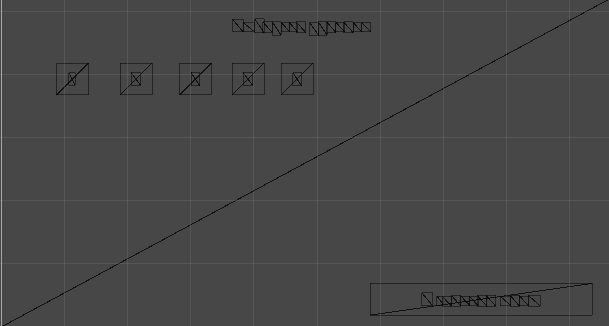


Рисунок 6.

Далее на рисунке 7 будет представлен итоговый внешний вид интерфейса выбора уровня.

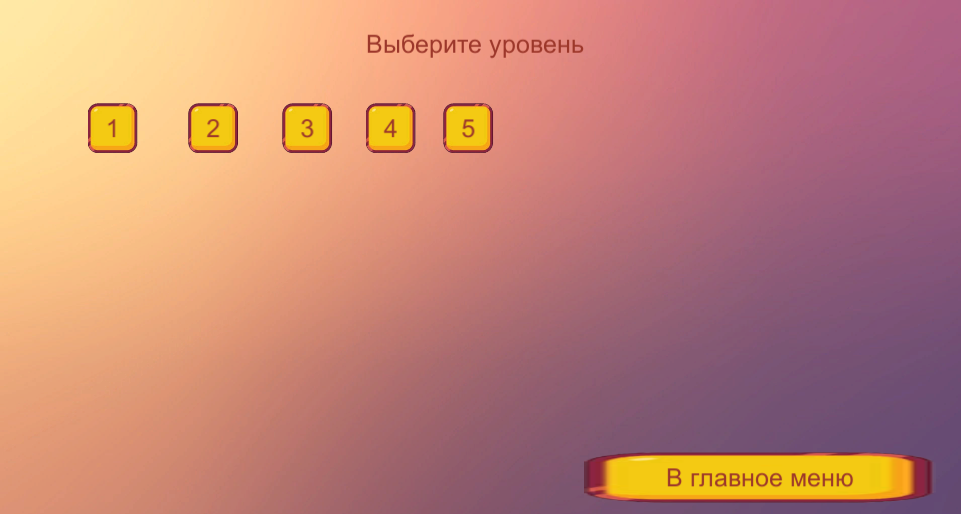


Рисунок 7.

1. Модуль победного экрана

На победном экране будут располагаться: текстовое поле, в котором будет сказано, что пользователь успешно прошел уровень, далее под данным текстовым полем будут располагаться 3 кнопки. Кнопка, возвращающая пользователя обратно в главное меню. Кнопка, возвращающая пользователя к модулю выбора уровня и кнопка, позволяющая пользователю выйти из приложения. Макет интерфейса данного модуля представлен ниже на рисунке 8.

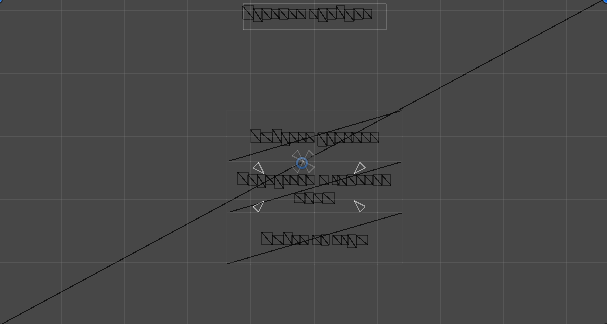


Рисунок 8.

Далее на рисунке 9 будет представлен итоговый внешний вид победного экрана.

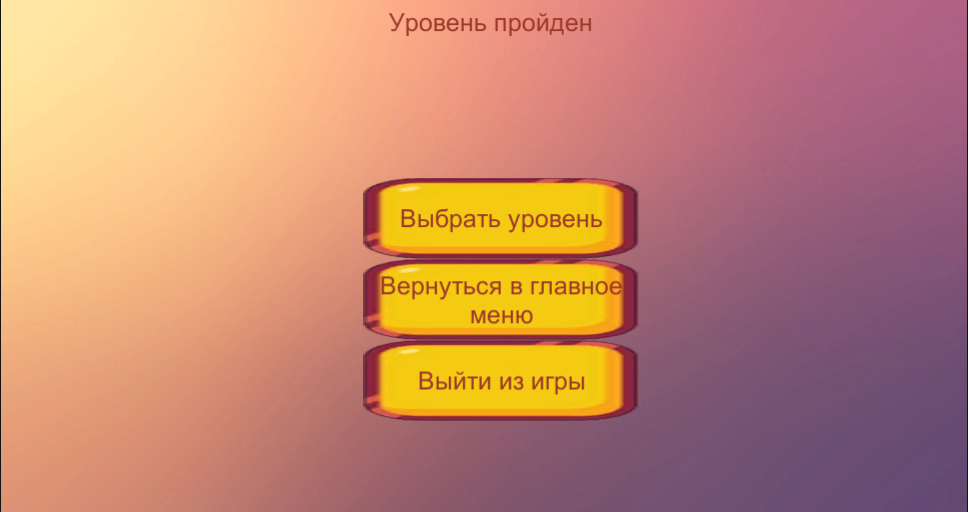


Рисунок 9.

1. Модуль самого уровня.

В данном случае не имеет смысла прикладывать макет и итоговый интерфейс для каждого конкретного уровня так как все различия на них — это лишь количество букв и слотов для них на экране. На данном экране будет представлено 2 горизонтальных поля. Первое будет для расположения слотов для вставки букв. А второе горизонтальное поле будет для расположения перемешанных букв, из которых пользователь будет составлять слово, перемещая полученные буквы в поле слотов. Также будут представлены кнопки для выхода в главное меню и выхода на экран выбора уровня соответственно. Макет интерфейса данного модуля представлен ниже на рисунке 10.

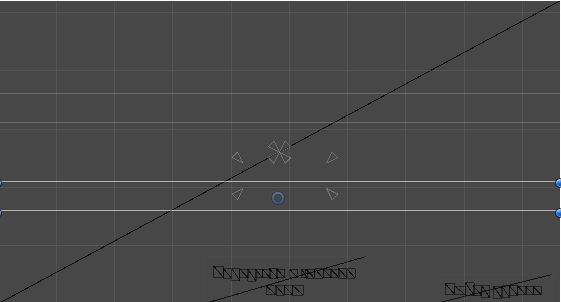


Рисунок 10.

Далее на рисунке 11 будет представлен итоговый внешний вид интерфейса данного окна.

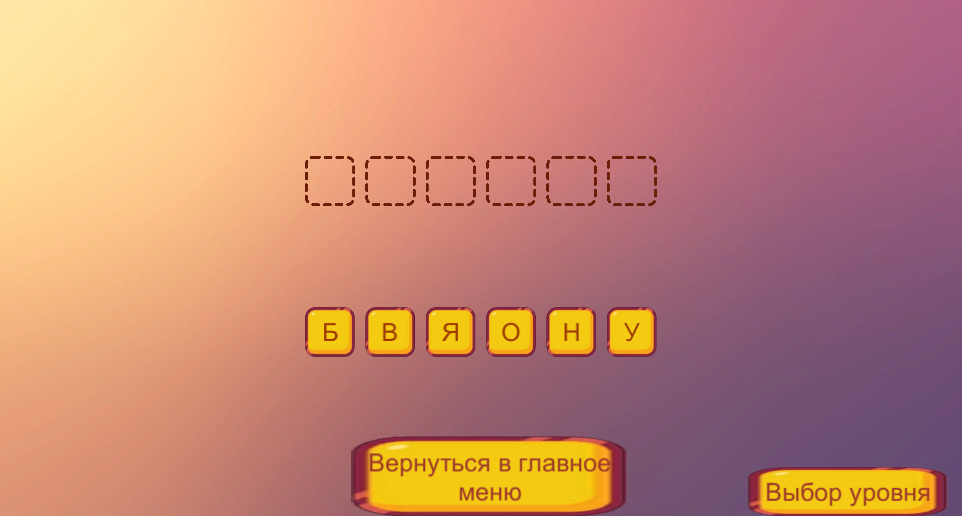


Рисунок 11.

Разработка логики работы приложения. Схема взаимодействия компонентов проекта.

В данной программе взаимодействуют лишь несколько компонентов, а именно слоты для букв и перетаскиваемы буквы. Если при окончании перетаскивания буква находиться в месте, где расположен слот происходит проверка совпадения заданных текстовых значений у объектов, при их совпадении в слот попадает перетащенная буква и счетчик увеличивается на 1, если после проверки выходит, что заданные значения не совпадают или буква была перетащена не на место слота то перетащенная буква возвращается на свою изначальную позицию.

Описание переменных, компонентов, классов и подпрограмм

Класс mainmenu

Поля

|  |
| --- |
| Таблица 1. |
| Название | Назначение | Значение |
| text | Приветствие игрока при входе | «Добро пожаловать в игру» |

Методы

|  |
| --- |
| Таблица 2. |
| Название | Назначение | Принимающиеся поля |
| onExitClick | Позволяет выйти из приложения | Приложение |
| onLevelChoiceClick | Открывает окно выбора уровня | Окно приложения |

Класс LevelChoice

Поля

|  |
| --- |
| Таблица 3 |
| Название | Назначение | Значение |
| text | Предложить пользователю выбрать уровень | «Выберите уровень» |

Методы

|  |
| --- |
| Таблица 4 |
| Название | Назначение | Принимающиеся поля |
| onLevelOneClick | Открывает первый уровень | Окно приложения |
| onLevelTwoClick | Открывает второй уровень | Окно приложения |
| onLevelThree | Открывает третий уровень | Окно приложения |
| onLevelFour | Открывает четвертый уровень | Окно приложения |
| onLevelFive | Открывает пятый уровень | Окно приложения |
| onMenuBackClick | Возвращает пользователя в главное меню | Окно приложения |

Класс WinRound

Поля

|  |
| --- |
| Таблица 5. |
| Название | Назначение | Значение |
| text | Донесение до пользователя, что уровень успешно пройден | «Уровень пройден» |

Методы

|  |  |  |
| --- | --- | --- |
| Таблица 6. | | |
| Название | Назначение | Принимающиеся поля |
| onExitClick | Позволяет выйти из приложения | Приложение |
| onMenuBackClick | Возвращает пользователя в главное меню | Окно приложения |
| onLevelChoiceClick | Открывает окно выбора уровня | Окно приложения |

Класс LevelOne

Переменные

|  |
| --- |
| Таблица 7. |
| Название | Назначение | Тип |
| Word | Изначальное слово | String |
| WordSplit | Слово, разделенное по буквам | Array |
| WordSplitShufle | Слово, разделенное по буквам и после перемешанное | Array |
| Letter | Буквенное значение для объекта | String |

Методы

|  |
| --- |
| Таблица 8 |
| Название | Назначение | К какому объекту относится |
| setLetter | Присваивает буквенное значение объекту | Слоту для букв |
| setShufleLetter | Присваивает буквенное значение объекту | К «букве» которую пользователь будет перетаскивать в выделенный для нее слот |
| OnDrag | Позволяет перетаскивать «буквы» | Перетаскиваемая буква |

|  |  |  |
| --- | --- | --- |
| Продолжение таблицы 8. | | |
| OnDrop | При окончании перетаскивания запускает другой метод getLetter | Перетаскиваемая буква и слот. |
| onWin | Проверка победил ли игрок, и вывод победного экрана | Экран приложения |
| getLetter | Проверка на совпадение значения присвоенного слоту для буквы и перетаскиваемой букве | Слот и перетаскиваемая буква |

Класс LevelTwo

Переменные

|  |
| --- |
| Таблица 9. |
| Название | Назначение | Тип |
| Word | Изначальное слово | String |
| WordSplit | Слово, разделенное по буквам | Array |
| WordSplitShufle | Слово, разделенное по буквам и после перемешанное | Array |
| Letter | Буквенное значение для объекта | String |

Методы

|  |
| --- |
| Таблица 10. |
| Название | Назначение | К какому объекту относится |
| setLetter | Присваивает буквенное значение объекту | Слоту для букв |

|  |  |  |
| --- | --- | --- |
| Продолжение таблицы 10. | | |
| setShufleLetter | Присваивает буквенное значение объекту | К «букве» которую пользователь будет перетаскивать в выделенный для нее слот |
| onWin | Проверка победил ли игрок, и вывод победного экрана | Экран приложения |
| getLetter | Проверка на совпадение значения присвоенного слоту для буквы и перетаскиваемой букве | Слот и перетаскиваемая буква |
| OnDrag | Позволяет перетаскивать «буквы» | Перетаскиваемая буква |
| OnDrop | При окончании перетаскивания запускает другой метод getLetter | Перетаскиваемая буква и слот. |

Класс LevelThree

Переменные

|  |
| --- |
| Таблица 11. |
| Название | Назначение | Тип |
| Word | Изначальное слово | String |
| WordSplit | Слово, разделенное по буквам | Array |
| WordSplitShufle | Слово, разделенное по буквам и после перемешанное | Array |
| Letter | Буквенное значение для объекта | String |

Методы

|  |
| --- |
| Таблица 12 |
| Название | Назначение | К какому объекту относится |
| setLetter | Присваивает буквенное значение объекту | Слоту для букв |
| setShufleLetter | Присваивает буквенное значение объекту | К «букве» которую пользователь будет перетаскивать в выделенный для нее слот |
| onWin | Проверка победил ли игрок, и вывод победного экрана | Экран приложения |
| getLetter | Проверка на совпадение значения присвоенного слоту для буквы и перетаскиваемой букве | Слот и перетаскиваемая буква |
| OnDrag | Позволяет перетаскивать «буквы» | Перетаскиваемая буква |
| OnDrop | При окончании перетаскивания запускает другой метод getLetter | Перетаскиваемая буква и слот. |

Класс LevelFour

Переменные

|  |
| --- |
| Таблица 13 |
| Название | Назначение | Тип |
| Word | Изначальное слово | String |
| WordSplit | Слово, разделенное по буквам | Array |
| WordSplitShufle | Слово, разделенное по буквам и после перемешанное | Array |
| Letter | Буквенное значение для объекта | String |

Методы

|  |
| --- |
| Таблица 14. |
| Название | Назначение | К какому объекту относится |
| setLetter | Присваивает буквенное значение объекту | Слоту для букв |
| setShufleLetter | Присваивает буквенное значение объекту | К «букве» которую пользователь будет перетаскивать в выделенный для нее слот |
| onWin | Проверка победил ли игрок, и вывод победного экрана | Экран приложения |
| getLetter | Проверка на совпадение значения присвоенного слоту для буквы и перетаскиваемой букве | Слот и перетаскиваемая буква |
| OnDrag | Позволяет перетаскивать «буквы» | Перетаскиваемая буква |
| OnDrop | При окончании перетаскивания запускает другой метод getLetter | Перетаскиваемая буква и слот. |

Класс LevelFive

Переменные

|  |
| --- |
| Таблица 15. |
| Название | Назначение | Тип |
| Word | Изначальное слово | String |
| WordSplit | Слово, разделенное по буквам | Array |
| WordSplitShufle | Слово, разделенное по буквам и после перемешанное | Array |
| Letter | Буквенное значение для объекта | String |

Методы

|  |
| --- |
| Таблица 16. |
| Название | Назначение | К какому объекту относится |
| setLetter | Присваивает буквенное значение объекту | Слоту для букв |
| setShufleLetter | Присваивает буквенное значение объекту | К «букве» которую пользователь будет перетаскивать в выделенный для нее слот |
| onWin | Проверка победил ли игрок, и вывод победного экрана | Экран приложения |
| getLetter | Проверка на совпадение значения присвоенного слоту для буквы и перетаскиваемой букве | Слот и перетаскиваемая буква |
| OnDrag | Позволяет перетаскивать «буквы» | Перетаскиваемая буква |
| OnDrop | При окончании перетаскивания запускает другой метод getLetter | Перетаскиваемая буква и слот. |

Тестирование и установка приложения

Тест 1. Проверка на неожиданные действия пользователя

В главном меню в таких случаях просто ничего не происходит. В меню выбора уровня и экрана победы ситуация аналогична. В самом уровне при хаотичных действиях ничего не происходит, но если пользователь начнет беспорядочно перемещать буквы, то они вернуться на свои изначальные позиции.

Тест 2. Неопытный пользователь открыл программу

В ходе эксплуатации данной программы не опытным пользователем не произошло никаких проблем.

Тест 3. Функциональное тестирование

В ходе тестирования было выявлено, что программа работает, как и было заявлено, но также была выявлена ошибка, в ходе которой при совпадении букв в слове их можно было поставить в одну ячейку.

Тест 4. Модульное тестирование

В ходе тестирования были проверены все модули программы, при работе с ними не было выявлено никаких ошибок.

Тест 5. Юзабилити тестирование

В ходе тестирования было выявлено, что приложение является удобным во всех аспектах, кроме попыток выйти из запущенного уровня, такой функционал отсутствует.

Для установки приложения требуется распаковать архив в папку.

Заключение

Приложение по логической игре «Эрудит» успешно было создано, были получены практические навыки создания подобных приложений. В течении работы были получены навыки создания, проектирования интерфейса приложения, а также тестирования итогового продукта.

Список использованной литературы

Хокинг, Джозеф. Unity — в действии. Мультиплатформенная разработка на C# : [рус.]. — 2. — СПб : Питер, 2016. — 336 с. — ISBN 978-1617292323.

Торн, Алан. Искусство создания сценариев в Unity : [рус.]. — СПб : ДМК, 2016. — 362 с.

С. Давыдов, А. Ефимов. IntelliJ IDEA. Профессиональное программирование на Java. — СПб.: БХВ, 2005. — 800 с. — (В подлиннике). — ISBN 5941576072.