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

Факультет компьютерных наук

Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  научный сотрудник Международной лаборатории интеллектуальных систем и структурного анализа, канд. Техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.В. Максименкова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2020 г. | УТВЕРЖДАЮ  Руководитель  образовательной программы  «Программная инженерия» проф, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Шилов  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2020 г. |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. | RU.17701729.04.01-01 81 01-1 | | **Компьютерная игра в жанре головоломка.**  **Пояснительная записка**  **ЛИСТ УТВЕРЖДЕНИЯ**  **RU.17701729.04.01-01 81 01-1-ЛУ**  Исполнитель:  Студентка группы БПИ 199  \_\_\_\_\_\_\_\_\_\_\_/Точилина П.В./  «\_\_\_»\_\_\_\_\_\_\_\_\_\_2020 г.    **Москва 2020** | |  |

УТВЕРЖДЕН  
RU.17701729.04.01-01 81 01-1-ЛУ

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Подп. и дата |  | | Инв. № дубл. |  | | Взам. Инв. № |  | | Подп. и дата |  | | Инв. № подл. | RU.17701729.04.01-01 81 01-1 | | **Компьютерная игра в жанре головоломка.**  **Пояснительная записка**  **RU.17701729.04.01-01 81 01-1**  **Листов 24** |  |

**Содержание**

[1. Введение 4](#_Toc40740414)

[1.1 Наименование программы: 4](#_Toc40740415)

[1.2 Документы, на основании которых ведется разработка: 4](#_Toc40740416)

[2. Назначение и область применения 5](#_Toc40740417)

[2.1 Назначение программы 5](#_Toc40740418)

[2.2 Область применения 5](#_Toc40740419)

[3. Технические характеристики 6](#_Toc40740420)

[3.1 Постановка задачи на разработку программы 6](#_Toc40740421)

[3.2. Описание и обоснование выбора инструментов разработки 6](#_Toc40740422)

[3.2 Описание функционирования программы 6](#_Toc40740423)

[3.2.1 Список частей приложения и описание их функций 6](#_Toc40740424)

[3.2.2 Описание взаимодействия с пользователем 6](#_Toc40740425)

[3.2.3 Описание алгоритма движения 6](#_Toc40740426)

[3.2.3 Возможные взаимодействия программы с другими программами 7](#_Toc40740427)

[3.3 Описание и обоснование выбора метода организации входных и выходных данных 7](#_Toc40740428)

[3.3.1 Входные данные 7](#_Toc40740429)

[3.3.2 Выходные данные. 7](#_Toc40740430)

[3.4 Описание и обоснование выбора организации хранения данных 7](#_Toc40740431)

[3.5 Описание и обоснование выбора состава технических и программных средств 7](#_Toc40740432)

[4. Ожидаемые технико-экономические показатели 9](#_Toc40740433)

[4.1 Предполагаемая потребность 9](#_Toc40740434)

[4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 9](#_Toc40740435)

[5. Список источников. 10](#_Toc40740436)

[*Приложение 1* 11](#_Toc40740437)

[Терминология. 11](#_Toc40740438)

[*Приложение 2* 12](#_Toc40740439)

[Описание и функциональное назначение классов 12](#_Toc40740440)

[*Приложение 3* 13](#_Toc40740441)

[Описание и функциональное назначение методов, полей и свойств 13](#_Toc40740442)

[*Приложение 4* 22](#_Toc40740443)

[Диаграмма классов 22](#_Toc40740444)

[ЛИСТ РЕГИСТРАЦИЙ ИЗМЕНЕНИЙ 24](#_Toc40740445)

1. Введение

1.1 Наименование программы:

Наименование программы – «Компьютерная игра в жанре головоломка» или «The PC Puzzle Game».

1.2 Документы, на основании которых ведется разработка:

Разработка ведется на основании приказа Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1112-04 от 11.12.2019.

2. Назначение и область применения

2.1 Назначение программы

Данная программа – игра в жанре головоломка, в которой игроку необходимо найти оптимальное решение поставленной задачи.

2.2 Область применения

Программа может быть использована для проведения досуга и развития логического мышления.

3. Технические характеристики

3.1 Постановка задачи на разработку программы

Курсовая работа. Тема: «Компьютерная игра в жанре головоломка». Цель разработки программы: предоставить работающую игру, в которую пользователь может играть без дополнительной подготовки.

Программа должна:

1. Запоминать прогресс игрока. Данные хранятся локально, используя json сериализацию.
2. Показывать анимацию прохождения уровня.
3. Позволять пользователю изменять настройки (громкость, разрешение).
4. Переключать уровни.
5. Показывать правила игры, если пользователь ещё их не видел.
6. Позволять игроку добавлять, двигать, вращать и удалять игровые объекты.
7. Уведомлять пользователя об успешном/неуспешном прохождении уровня.

3.2. Описание и обоснование выбора инструментов разработки

Для разработки использовалась межплатформенная среда разработки компьютерных игр Unity. Разработка велась на языке C#. Код написан в среде Microsoft Visual Studio 2019. Выбор Unity обоснован ограничением на выбор языка.

3.2 Описание функционирования программы

3.2.1 Список частей приложения и описание их функций

Программа состоит из двух основных частей: системы пользовательского интерфейса и системы уровней в определённой последовательности. Через главное меню осуществляется доступ к игровым уровням и настройкам, а также выход из программы.

3.2.2 Описание взаимодействия с пользователем

После запуска программы открывается главное меню, в котором есть 3 кнопки: Start – осуществляет переход к игровым уровням, Settings – переход к настройкам, Exit – завершение работы программы. В настройках пользователь может изменять громкость игры и разрешение.

Если пользователь ещё не проходил правила, они будут показаны ему при запуске уровня.

На каждом уровне пользователь может размещать неограниченное количество объектов-стрелок. При нажатии на кнопку Play запускается анимация уровня. В случае успешного прохождения включается следующий уровень, либо меню конца, в котором пользователь может начать новую игру.

3.2.3 Описание алгоритма движения

Объект персонажа имеет компонент Rigidbody2D и физический материал с параметрами: Friction = 0, Bounciness = 1. Это позволяет не обрабатывать специально отталкивание персонажа от стен и использовать встроенную в движок физику. В настройках проекта значение Velocity Threshold установлено на 0.0001, таким образом наш герой не будет застревать в стенах на меньшей скорости.

При запуске уровня меняется значение переменной running, от которой зависит наличие анимации. Далее поведение меняется только в случае коллизии с объектами на карте, либо при нажатии на кнопку остановки уровня.

Варианты коллизии:

1. Со стеной. Стены не являются триггерами, поэтому движок сам обрабатывает поведение в данном случае.
2. С объектом-стрелкой.

- Записывается информация об угле, на который повёрнута стрелка.

- По значению velocity вычисляется текущий угол поворота игрового персонажа.

- Высчитывается разница между двумя углами, не превышающая 180 градусов.

- К значению счётчика прибавляется разница углов.

- При превышении максимально допустимого значения счётчика появляется уведомление, игровой персонаж возвращается в начальную позицию, анимация останавливается.

- Если ограничение не превышено, Velocity персонажа приравнивается к вектору (x – cos поворота стрелки, y – sin поворота стрелки), помноженному на скорость.

1. С точкой назначения. При коллизии с финишем будет показано уведомление и включён следующий уровень. Если это был последний уровень, тогда пользователь увидит меню окончания игры.
2. С объектом опасности и краем карты. Показывается уведомление, игровой персонаж возвращается в начальное положение, выключается анимация.

3.2.3 Возможные взаимодействия программы с другими программами

Взаимодействие с другими программами не предусмотрено.

3.3 Описание и обоснование выбора метода организации входных и выходных данных

3.3.1 Входные данные

Во всех меню ввод данных производится через мышь. Во время игры ввод осуществляется через:

1. Мышь при добавлении новых объектов на уровень, перемещении объектов по карте, передвижении камеры, запуске и остановке анимации, закрытии диалоговых окон, выходе в главное меню.
2. Стрелки ←, → при вращении объектов.
3. Клавишу delete при удалении объекта с карты.

3.3.2 Выходные данные.

Выходные данные передаются на дисплей в графический интерфейс. При запуске уровня должна включаться анимация персонажа.

3.4 Описание и обоснование выбора организации хранения данных

Данные о прогрессе пользователя хранятся локально в json для более гибких возможностей изменения хранимой информации.

3.5 Описание и обоснование выбора состава технических и программных средств

1. Манипулятор типа «Мышь».
2. Клавиатура.
3. Монитор с разрешением не менее 800x600 точек.
4. Не менее 1 ГБ свободного места для хранения программы.
5. Операционная система Windows 7 (SP1+) или Windows 10.
6. x86, x64 architecture with SSE2 instruction set support.

4. Ожидаемые технико-экономические показатели

В рамках данной работы расчёт экономической эффективности не предусмотрен.

4.1 Предполагаемая потребность

Целевая аудитория – подростки и взрослые, заинтересованные в подобном проведении досуга. Программа может быть использована как в развлекательных целях, так и в целях развития логического мышления.

4.2 Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

1. Уникальная механика игры;
2. Низкий входной порог;
3. Бесплатное распространение.

5. Список источников.

1. ГОСТ 19.404-79. ЕСПД. Пояснительная записка. Требования к содержанию и оформлению. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.103-77 ЕСПД. Обозначения программ и программных документов. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.105-78 ЕСПД. Общие требования к программным документам. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.106-78 ЕСПД. Требования к программным документам, выполненным печатным способом. – М.: ИПК Издательство стандартов, 2001.
5. Unity Documentation [Электронный ресурс]. Режим доступа: <https://docs.unity3d.com/Manual/index.html>, свободный. Дата обращения: многократно 1.03. - 10.05.2020.
6. Unity system requirements [Электронный ресурс]. Режим доступа: <https://docs.unity3d.com/Manual/system-requirements.html#desktop>, свободный. Дата обращения: 15.05.2020.

# *Приложение 1*

Терминология.

*Таблица 1 - Терминология*

|  |  |
| --- | --- |
| Термин | Определение |
| Игровой персонаж | Игровой объект, которым управляет человек-игрок. |
| Сцена | Объект, на котором расставляется уровень. |
| Камера | Игровой объект, который как бы является глазами игрока. Камерой ограничена область видимости сцены. |
| Сила | Задание направления и скорости движения некоторого физического объекта. |
| Сериализация | Перевод некоторой структуры (в данном случае класс Progress) в последовательность битов. |
| json | Текстовый формат хранения данных. Удобен для чтения людьми. |

# *Приложение 2*

Описание и функциональное назначение классов

Все классы, кроме Progress, являются наследниками MonoBehavior.

*Таблица 2 - Классы приложения*

|  |  |
| --- | --- |
| **Класс** | **Назначение** |
| AngleBar | Скрипт счётчика угла. |
| AnimatorHege | Скрипт анимации главного героя. |
| ApplicationCloser | Скрипт закрытия приложения. |
| CameraDrag2 | Скрипт передвижения камеры. |
| DoNotDestroy | Для сохранения аудио между сценами. |
| DragAndMove | Захват и передвижение объектов. |
| Globals | Настройки уровня. |
| MessageBoxScript | Скрипт для окна сообщения/ошибки. |
| PlayAgain | Скрипт для повторной игры после окончания. |
| PlayerController | Содержит алгоритм движения персонажа при запуске уровня. |
| Progress | Singleton. Хранит информацию о прогрессе пользователя. |
| RulesScript | Скрипт для диалоговых окон с правилами игры. |
| SceneLoader | Скрипт для загрузки сцен. |
| Settings | Скрипт для сцены настроек. |
| UiController | Контроллер для канваса поверх уровня. |

# *Приложение 3*

Описание и функциональное назначение методов, полей и свойств

*Таблица 3 – описание членов класса AngleBar*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Методы | | | | | |
| Имя | Мод. доступа | Тип | Аргументы | | Назначение |
| IncreaseAngle | Public | Void | float angle | | Увеличивает счётчик углов на переданное значение. |
| SetAngle | Public | Void | float angle | | Устанавливает счётчик угла на переданное значение. |
| Start | Private | Void | - | | Устанавливает начальные значения. |
| Поля | | | | | |
| Имя | Мод. доступа | Тип | | Назначение | |
| slider | Public | Slider | | Компонент, который позволяет проще реализовать счётчик угла, т. к. в данном случае по действию он схож со слайдером. | |
| gradient | Public | Gradient | | Градиент для наложения на изображение при изменении значения счётчика. | |
| fill | Public | Image | | Изображение, которым заполняется полоска счётчика. | |
| counter | Public | Text | | Текстовое поле для выведения точного значения счётчика. | |
| dialogCanvas | Public | GameObject | | Диалоговое окно, используемое для вывода информации о превышении лимита. | |
| globals | Private | Globals | | Настройки уровня. | |

*Таблица 4 – описание членов класса AnimatorHege*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Методы | | | | | |
| Имя | Мод. доступа | Тип | Аргументы | | Назначение |
| Start | Private | Void | - | | Устанавливает анимацию. |
| Поля | | | | | |
| Имя | Мод. доступа | Тип | | Назначение | |
| anim | Public static | Animator | | Аниматор главного персонажа. | |

*Таблица 5 – описание членов класса ApplicationCloser*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Тип | Аргументы | Назначение |
| CloseApplication | Public | Void | - | Закрывает программу. |

*Таблица 6 – описание членов класса CameraDrag2*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Методы | | | | | |
| Имя | Мод. доступа | Тип | Аргументы | | Назначение |
| Update | Private | Void | - | | Вызывается раз в кадр. Отслеживает зажатие и передвижение мыши. Высчитывает перемещение камеры, учитывая скорость, и задаёт новые значения для неё. Учитывает границы карты. |
| Поля | | | | | |
| Имя | Мод. доступа | Тип | | Назначение | |
| dragSpeed | Public | Float | | Скорость, с которой перемещается камера. | |
| dragOrigin | Private | Vector3 | | Координаты захвата камеры для расчёта перемещения. | |
| panLimit | Public | Vector2 | | Границы уровня (на сколько может передвинуться камера). | |

*Таблица 7 – описание членов класса DoNotDestroy*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Методы | | | | | |
| Имя | Мод. доступа | Тип | Аргументы | | Назначение |
| Awake | Private | Void | - | | Сохраняет единственный объект при переключении сцен. |
| Поля | | | | | |
| Имя | Мод. доступа | Тип | | Назначение | |
| instance | Private static | GameObject | | Единственный объект, который сохраняется всё время при смене сцен. | |

*Таблица 8 – описание членов класса DragAndMove*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Методы | | | | | |
| Имя | Мод. доступа | Тип | Аргументы | | Назначение |
| Start | Private | Void | - | | Получает глобальные настройки от камеры. |
| Update | Private | Void | - | | Отслеживает поворот, перемещение и удаление объекта. Выполняет все перечисленные функции. |
| OnMouseDown | Public | Void | - | | Устанавливает координаты, на которых был схвачен объект. |
| OnMouseUp | Public | Void | - | | Если не был выбран объект, то теперь он выбран. Если объект не был передвинут, то теперь он не выбран. |
| Поля | | | | | |
| Имя | Мод. доступа | Тип | | Назначение | |
| StartPosX | Private | Float | | Координата х, где был схвачен объект. | |
| StartPosY | Private | Float | | Координата y, где был схвачен объект. | |
| selected | Private | Bool | | Выбран ли объект. | |
| moved | Private | Bool | | Был ли объект передвинут. | |
| sprite | Public | GameObject | | Спрайт объекта для того, чтобы изменять цвет при выборе изменении значения Selected. | |
| self | Public | GameObject | | Для уничтожения конкретного экземпляра. | |
| rotateSpeed | Public | Float | | Скорость поворота объекта. | |
| isBeingHeld | Public | Bool | | Передвигают ли сейчас объект. | |
| globals | Private | Globals | | Настройки уровня. | |

*Таблица 9 – описание членов класса Globals*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | | |
| Имя | Мод. доступа | Тип | Аргументы | | | Назначение | |
| Start | Private | Void | - | | | Устанавливает значение endRules. | |
| GetDegreeAngle | Public | Float | Vector2 vec | | | Возвращает значение угла в градусах, найденного по переданному вектору. | |
| Поля | | | | | | | |
| Имя | Мод. доступа | Тип | | | Назначение | | |
| endRules | Private | Bool | | | Прошёл ли игрок правила. | | |
| speed | Public | Float | | | Скорость персонажа. | | |
| startDirection | Public | Vector2 | | | Направление, в котором персонаж начинает движение. | | |
| maxAngle | Public | Float | | | Максимальный суммарный угол поворота. | | |
| startPosition | Public | Vector3 | | | Начальное положение игрового персонажа. | | |
| selected | Public | Color | | | Цвет выбранного объекта. | | |
| nonSelected | Public | Color | | | Цвет невыбранного объекта. | | |
| playing | Private | Bool | | | Идёт ли анимация уровня. | | |
| Свойства | | | | | | | |
| Имя | Мод. доступа | Тип | | Доступ | | | Назначение |
| EndRules | Public | Bool | | Get, Set | | | К переменной endRules. При сеттере сериализует Progress. |
| Playing | Public | Bool | | Get, Set | | | К переменной playing. При сеттере делает объект невыбранным, если value=true. |

*Таблица 10 – описание членов класса MessageBoxScript*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | |
| Имя | | Мод. доступа | Тип | Аргументы | | Назначение |
| Start | | Private | Void | - | | Устанавливает canvas.active на false |
| Show | | Public | Void | string message, bool endLev = false | | Показывает диалоговое окно. |
| NotPlaying | | Public | Void | - | | Вызывается при нажатии кнопки внутри. Запускает следующий уровень, если текущий был пройден. |
| Поля | | | | | | | |
| Имя | Мод. доступа | | Тип | | Назначение | | |
| endLevel | Private | | Bool | | Пройден ли уровень. | | |
| canvas | Public | | GameObject | | Кэнвас, в котором показывается окно. | | |
| textField | Public | | Text | | Текстовое поле внутри окна. | | |
| playButton | Public | | Button | | Кнопка для запуска уровня. | | |
| globals | Private | | Globals | | Настройки уровня. | | |

*Таблица 11 – описание членов класса PlayAgain*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Тип | Аргументы | Назначение |
| OnButton | Private | Void | - | Обновляет json, устанавливая пустой Progress. |

*Таблица 12 – описание членов класса PlayerController*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | |
| Имя | | Мод. доступа | Тип | Аргументы | | Назначение |
| OnTriggerEnter2D | | Private | Void | Collider2D other | | Обработка коллизии с различными объектами на карте. |
| SetStartSettings | | Private | Void | - | | Устанавливает героя в начальное положение. |
| Start | | Private | Void | - | | Устанавливает игрового персонажа при запуске уровня. |
| FixedUpdate | | Public | Void | - | | Ставит значение анимации, если персонаж двигается. |
| Поля | | | | | | | |
| Имя | Мод. доступа | | Тип | | Назначение | | |
| dialogCanvas | Public | | GameObject | | Канвас, в котором открываются уведомления о столкновении. | | |
| angleBar | Public | | AngleBar | | Счётчик угла. | | |
| moveSpeed | Public | | Float | | Скорость движения игрового персонажа. | | |
| rotateSpeed | Public | | Float | | Скорость вращения персонажа. | | |
| globals | Private | | Globals | | Настройки уровня | | |

*Таблица 13 – описание членов класса Progress*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | |
| Имя | | Мод. доступа | Тип | Аргументы | | Назначение |
| GetInstance | | Public static | Progress | - | | Реализация Singleton. |
| - | | Private | Progress | - | | Конструктор для сериализации. |
| Serialize | | Public static | Void | - | | Сериализует самого себя. |
| Deserialize | | Public static | Void | - | | Десериализует файл в себя. |
| Поля | | | | | | | |
| Имя | Мод. доступа | | Тип | | Назначение | | |
| path | Public static | | String | | Путь, где хранится сериализованный экземпляр. | | |
| instance | Private static | | Progress | | Единственный экземпляр. | | |
| Свойства | | | | | | | |
| Имя | Мод. доступа | | Тип | Доступ | | Назначение | |
| CurrentLevel | Public | | Int | Get, Set | | Текущий уровень. | |
| LevelAmount | Public | | Int | Get, Set | | Общее количество уровней в игре. | |
| EndRule | Public | | Bool | Get, Set | | Пройдены ли правила. | |

*Таблица 14 – описание членов класса RulesScript*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | |
| Имя | | Мод. доступа | Тип | Аргументы | | Назначение |
| Start | | Private | Void | - | | Делает окно активным, если правила не пройдены. |
| Show | | Public | Void | - | | Делает окно активным. |
| OnButton | | Public | Void | - | | При нажатии на кнопку внутри окна, открывает следующее окно, либо записывает, что правила пройдены. |
| Поля | | | | | | | |
| Имя | Мод. доступа | | Тип | | Назначение | | |
| box | Public | | GameObject | | Окно. | | |
| text | Public | | Text | | Текстовое поле сообщения. | | |
| iter | Private | | Int | | Нужен для переключения текста в окне до переключения между окнами. | | |
| sentences | Public | | String[] | | Все предложения, которые будут показаны в окне. | | |
| next | Public | | GameObject | | Следующее окно. | | |
| isFirst | Public | | Bool | | Является ли окно первым. | | |
| globals | Private | | Globals | | Настройки уровня. | | |

*Таблица 15 – описание членов класса SceneLoader*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Тип | Аргументы | Назначение |
| Start | Private | Void | - | Десериализует прогресс из файла, если он существует. |
| LoadScene | Public | Void | string name | Загружае сцену с переданным названием. |
| OnStartButton | Public | Void | - | Вызывается при нажатии на кнопку старта. Открывает либо уровень, либо сцену конца. |

*Таблица 16 – описание членов класса Settings*

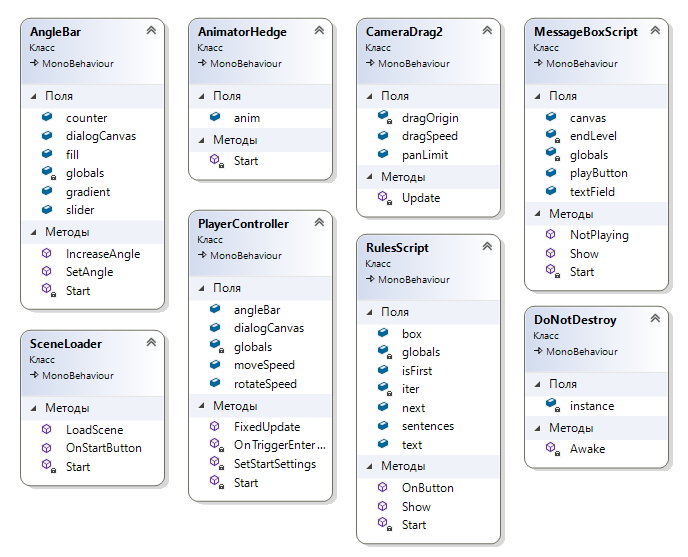
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | |
| Имя | | Мод. доступа | Тип | Аргументы | | Назначение |
| Start | | Private | Void | - | | Вызывается при запуске. Устанавливает в список все возможные разрешения пользователя. |
| SetVolume | | Public | Void | float volume | | Установка нового значения громкости. |
| SetFullscreen | | Public | Void | bool isFull | | Раскрытие программы на полное окно и сокрытие. |
| SetResolution | | Public | Void | int resIndex | | Изменение разрешения игры. |
| Поля | | | | | | | |
| Имя | Мод. доступа | | Тип | | Назначение | | |
| audioMixer | Public | | AudioMixer | | Аудиомиксер для изменения звука. | | |
| resolutions | Private | | Resolution[] | | Список всех возможных разрешений пользователя. | | |
| resDrop | Public | | Dropdown | | Дропдаун меню с разрешениями. | | |

*Таблица 17 – описание членов класса UiController*

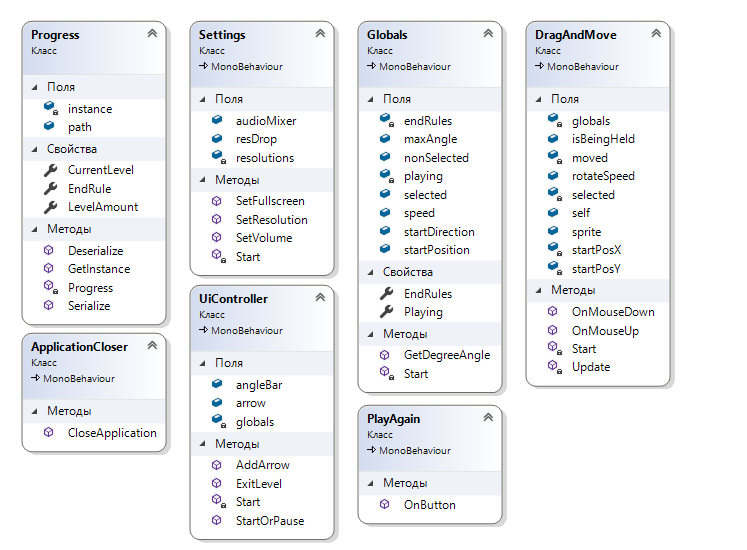
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Методы | | | | | | |
| Имя | | Мод. доступа | Тип | Аргументы | | Назначение |
| Start | | Private | Void | - | | Запускает один раз при запуске. Устанавливает настройки уровня. |
| StartOrPause | | Public | Void | - | | В зависимости от состояния запускает уровень, придавая силу персонажу, или возвращает всё в начальное положение. |
| AddArrow | | Public | Void | - | | Добавляет новую стрелку на карту. |
| ExitLevel | | Public | Void | - | | Загружает главное меню. |
| Поля | | | | | | | |
| Имя | Мод. доступа | | Тип | | Назначение | | |
| angleBar | Public | | AngleBar | | Счётчик угла поворотов. | | |
| arrow | Public | | Object | | Объект-стрелка для клонирования при добавлении на сцену новых стрелок. | | |
| globals | Private | | Globals | | Настройки уровня. | | |

# *Приложение 4*

Диаграмма классов



*Рисунок 1 – Диаграмма классов приложения*



*Рисунок 2 – Продолжение диаграммы классов приложения*

ЛИСТ РЕГИСТРАЦИЙ ИЗМЕНЕНИЙ

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