

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

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Профессор департамента
программной инженерии
Факультета компьютерных наук
кандидат технических наук

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

_____ Е.М.Гринкруг
«___» _____ 2019 г.

_____ В.В. Шилов
«___» _____ 2019 г.

Java-приложение «Ханойские башни»

Пояснительная записка

ЛИСТ УТВЕРЖДЕНИЯ
RU.17701729.04.01-01 81 01-1-ЛУ

Исполнитель
Студент БПИ172
_____/Червинский А. Л./
«___» _____ 2019 г.

Инв. № подл.	Подп. и дата	Взам. Инв. №	Инв. № дубл.	одп. и дата
RU.17701729.04.01-01 8101-1				

Москва 2019

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

**Java-приложение «Ханойские башни»
Пояснительная записка**

**RU.17701729.04.01-01 81 01-1
Листов 31**

<i>Инв. № подл.</i>	<i>Подп. и дата</i>	<i>Взам. Инв. №</i>	<i>Инв. № дубл.</i>	<i>Подп. и дата</i>
RU.17701729.04.01-01 81 01-1				

Москва 2019

Содержание

1. Введение.....	3
1.1. Наименование программы.....	3
1.2. Основания для разработки.....	3
2. Назначение и область применения.....	4
2.1. Назначение разработки.....	4
2.1.1. Функциональное назначение.....	4
2.1.2. Эксплуатационное назначение.....	4
3. Технические характеристики.....	5
3.1. Постановка задачи на разработку программы.....	5
3.2. Описание и обоснование алгоритма и функционирования программы.....	5
3.2.1. Описание алгоритма и функционирования программы.....	5
3.2.2. Обоснование выбора алгоритма.....	6
3.3. Описание и обоснование метода организации входных и выходных данных.....	6
3.3.1. Описание метода организации входных и выходных данных.....	6
3.3.2. Обоснование метода организации входных и выходных данных.....	6
3.4. Описание и обоснование выбора состава технических и программных средств.....	7
3.4.1. Состав технических и программных средств.....	7
3.4.2. Обоснование выбора состава технических и программных средств.....	7
4. Техничко-экономические показатели.....	8
4.1. Предполагаемая потребность.....	8
4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами.....	8
Приложение 1.....	9
Список используемой литературы.....	9
Приложение 2. Описание и функциональное назначение методов и свойств и полей классов	10

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

1. Введение

1.1. Наименование программы

Наименование программы – «Java-приложение «Ханойские башни»».

Наименование программы на английском языке –

«*"Tower of Hanoi" - Java-application*».

1.2. Основания для разработки

Разработка «Java-приложение «Ханойские башни»» ведется на основании приказа декана факультета компьютерных наук Национального исследовательского университета «Высшая школа экономики» № 2.3-02/1212-01 от 12.12.2018 "Об утверждении тем, руководителей курсовых работ студентов образовательной программы Программная инженерия факультета компьютерных наук".

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

2. Назначение и область применения

2.1. Назначение разработки

2.1.1. Функциональное назначение

Функциональным назначением программы является — 3D-визуализация манипуляций с игровыми объектами в соответствии с правилами игры.

Правила игры: Даны три стержня, на левый из которых нанизаны кольца (их количество будет отличаться в зависимости от выбранной сложности игры), причем кольца отличаются размером и лежат меньшее на большем. Задача состоит в том, чтобы перенести пирамиду из колец за наименьшее число ходов на правый стержень. За один ход разрешается перенести только одно кольцо, причем нельзя класть большее на меньшее.

2.1.2. Эксплуатационное назначение

Программа дает возможность пользователю играть при помощи персонального компьютера.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

3. Технические характеристики

3.1. Постановка задачи на разработку программы

Программа должна поддерживать следующие функции:

1. Готовить поле для игры "Ханойские башни" при запуске
 2. Правильно отрисовывать фигуры для игры "Ханойские башни"
 3. Передвигать кольца с одного шеста на другой в соответствии с правилами игры "Ханойские башни"
 4. Сообщать пользователю о победе и затраченном времени и затраченных ходах на решение головоломки в случае, если выполнены все условия победы в игре "Ханойские башни"
 5. Возможность изменять угол поворота камеры вокруг поля игры
 6. После победы перерисовывать поле для игры "Ханойские башни" при запросе пользователя
 7. Автоматически решать головоломку
 8. Отображать время и количество ходов уже затраченных на решение головоломки во время игры
 9. Записывать данные в таблицу рекордов.
 10. Корректно отображать меню игры
 11. Предоставлять возможность пользователю выбирать сложность
 12. Генерировать таблицу и позволять пользователю изменять сложность для просмотра отдельных разделов таблицы рекордов
- Сохранять игру и загружать при запросе пользователя

3.2. Описание и обоснование алгоритма и функционирования программы

3.2.1. Описание алгоритма и функционирования программы

- Алгоритм передвижения колец и проверки правил игры:
При помощи событий колец мы отслеживаем на какое кольцо нажал пользователь, после того как пользователь нажал на кольцо/стержень мы записываем номер стержня на котором он находился или если был нажат стержень просто записываем его номер (номера: 0,1,2) в поле tempStackNumber, таким образом программа узнает откуда будет перенос, когда пользователь нажмет кольцо/стержень еще раз программа будет знать куда переносить и выполнит метод Move(tempStackNumber, <новый номер стержня>). В методе Move будет выполнена проверка при помощи вспомогательного массива стеков, где хранятся целые числа, которые

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

соответствуют массиву стеков колец, но вместо радиусов целые числа от 1 до числа которое соответствует количеству колец, таким образом: поскольку массив стеков целых чисел полностью дублирует список стеков колец идет проверка, если верхнее число стека больше верхнего числа стека с которого идет перенос мы осуществляем перенос для массива чисел, если перенос возможен с массива стеков целых чисел, перенос возможен и для списка стеков колец поэтому далее мы осуществляем перенос при помощи алгоритма позиционирования колец при переносе в методе pMove.

- Алгоритм позиционирования колец после переноса:
Данный алгоритм вызывается в методе pMove, после проверки правил игры в методе Move. В pMove передается откуда и куда осуществляется перенос, в константах записаны координаты по X и они соответствуют номерам стержней (0 = xLeft, 1=xCenter, 2=xRight). А для определения координаты по y мы рассчитываем сначала начальное место для колец(в самом низу) по формуле $500 + (\text{difficulty} + 1.5) * \text{blocksize} / 2 + \text{blocksize} / 2 * (\text{yStart})$. Здесь $500 + (\text{difficulty} + 1.5) * \text{blocksize} / 2$ координата нижней части стержня. Далее исходя из количества колец в стеке с кольцами идет расчет по формуле $\text{yStart} - \text{pField}[\text{to}].\text{Size}() * \text{blockSize}$. Таким образом получаются координаты x и y, которые отправляются в метод Animate() для анимации.
- Алгоритм создания анимации:
Для создания анимации используется отдельный поток в котором по очереди запускаются таймлайны (goingUp, goingDown, goingToX), каждый из которых определяет отдельную анимацию:
goingUp — перемещение вверх,
goingToX — движение по горизонтали,
goingDown — движение вниз.
При этом после запуска каждого таймлайна поток засыпает до момента когда отдельный таймлайн закончится, после чего запускается следующий
- Алгоритм создания колец при помощи MeshView выполняется следующим образом:
 1. Рассчитать координаты вершин и сохранить их в сетке
 2. Рассчитать грани треугольников и сохранить их в сетке
 3. Рассчитать координаты текстур и сохранить их в сетке

Расчет координат вершин:

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Сначала вычисляется вершина поперечного сечения в плоскости $Z = 0$, вычисление всех остальных точек получается за счет вращения точек, вычисленные на шаге 1, относительно центра кольца $[0,0,0]$.

Расчет граней треугольников:

Данные о вершинах были сохранены таким образом, чтобы индекс массива для сегмента r кольца и сегмента трубки t был задан формулой $r + t * \text{nbrRingSteps}$, поэтому мы можем вычислить 4 индекса, которые составляют пересечение сегмента кольца и трубки. Для каждого треугольника должны предоставить индексы элементов массива для данных вершин и координат текстуры для каждого из трех углов треугольника. Вершины должны быть упорядочены против часовой стрелки. Так как между элементами вершинного и текстурного координатного массива есть соответствие 1: 1, что упрощает используемый алгоритм. Далее полученный квадрат делится пополам диагональю.

Расчет координат для текстур:

Так как мы покрываем поверхность кольца, нам нужна координата текстуры для каждой вершины. Координаты будут в диапазоне $[0,0] - [1,1]$. При вызове этого метода параметры определяют частоту мозаичного изображения.

3.2.2. Обоснование выбора алгоритмов

Я выбрал эти алгоритмы, так как они наилучшим образом удовлетворяют поставленной задаче.

3.3. Описание и обоснование метода организации входных и выходных данных

3.3.1. Описание метода организации входных и выходных данных

3.3.1.1. Описание метода организации входных данных

Программа при помощи событий, мыши и клавиатуры, получает данные автоматически

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

3.3.1.2. Описание метода организации выходных данных

Программа правильно отображает данные исходя из действий пользователя

3.3.2. Обоснование метода организации входных и выходных данных

Данные методы были выбраны мною, так как они лучше всего подходят для выполнения данной задачи

3.4. Описание и обоснование выбора состава технических и программных средств

3.4.1. Состав технических и программных средств

3.4.1.1. Состав технических средств

Для нормального функционирования программы требуется компьютер, оснащенный следующими техническими компонентами:

1. Персональный компьютер, оснащенный 32-разрядным или 64-разрядным процессором Pentium с тактовой частотой 400 МГц и выше или аналогичный процессор. (рекомендуется Pentium с тактовой частотой 1 ГГц и выше или аналогичный процессор)
2. Оперативная память не ниже 1 Гб.
3. Не менее 1.5 Гб свободного места
4. видеокарта и монитор, поддерживающие режим Super VGA с разрешением не менее чем 850x600 точек
5. мышь Microsoft Mouse или совместимое указывающее устройство

3.4.1.2. Состав программных средств

1. Операционная система windows 7 версии 6.1.7601.23403 и более поздние версии или linux ubuntu версии 18.04 LTS и более поздние.
2. Java jre 8

3.4.2. Обоснование выбора состава технических и программных средств

3.4.2.1. Обоснование выбора состава технических средств

Это минимальные характеристики поддерживаемых Устройств, при их отсутствии или не соответствии программа может работать некорректно.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

3.4.2.2. Обоснование выбора состава программных средств

Программа не будет корректно работать без программных средств приведенных в составе программных средств

.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

4. Техничко-экономические показатели

4.1. Предполагаемая потребность

Данная программа — это компьютерная игра. Она будет представлять интерес для пользователей, которые любя головоломки и хотят провести свой досуг за игрой.

4.2. Ориентировочная экономическая эффективность

В рамках данной работы расчет экономической эффективности не предусмотрен, так как программа будет распространяться бесплатно.

4.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Программа обладает более удобным управлением нежели аналоги. Никто из аналогов не использует трехмерную графику, также не у кого из аналогов нет анимированного авто-сбора.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Приложение 1

Список используемой литературы

- 1) ГОСТ 19.101-77 Виды программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 2) ГОСТ 19.102-77 Стадии разработки. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 3) ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 4) ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 5) ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 6) ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 7) ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 8) ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 9) ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
- 10) ГОСТ 15150-69 Машины, приборы и другие технические изделия. Исполнения для различных климатических районов. Категории, условия эксплуатации, хранения и транспортирования в части воздействия климатических факторов внешней среды. – М.: Изд-во стандартов, 1997.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Приложение 2. Описание и функциональное назначение методов и свойств и полей классов

Табл. 1 Методы класса *DifficultyScene*

Имя	Модификатор доступа	Назначение
addButtonEvents()	private	Добавляет события кнопок.
closeStage()	private	Закрывает сцену.
Init()	Package private	Инициализирует класс и добавляет события кнопок.

Табл.2 Поля класса *DifficultyScene*

Имя	Назначение
backButton	Кнопка в меню, переводит обратно в меню при нажатии
easyBtn	Выбор сложности - легкая
hardButton	Выбор сложности - сложная
insaneBtn	Выбор сложности - очень сложная
midBtn	Выбор сложности - средняя
playButton	Кнопка играть

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

difficulty	Сложность игры, где целое число означает количество колец
gamePane	Содержит AnchorPane с игрой
primaryStage	Главня сцена

Табл 3. Методы и события класса Game

Имя	Модификатор доступа	Назначение
animate()	private	Анимировать переход колец
auto()	public	Начинает автоматический сбор
autoSolve()	public	Автоматический сбор
buildTorus()	public	Создает кольца
clickHandler()	private	Метод вызывается при нажатии на игровые фигуры, обрабатывает нажатия
getDifficulty()	Package private	Возвращает сохраненную сложность, необходимо для передачи в winnerController, если объект сериализовался
getGroup()	private	Возвращает группу построенную на основе всех элементов
getMoves()	public	Возвращает сохраненное количество шагов, необходимо для

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

		передачи в winnerController
getTime()	public	Возвращает сохраненное время, необходимо для передачи в winnerController
move()	private	проверяет можно ли передвинуть кольцо с одного шеста на другой, если можно передвигает и передает в главный контроллер результат передвижения.
pMove()	private	Передвигает кольцо с одного шеста на другой
returnGameField()	public	Возвращает группу со всеми элементами для добавления на сцену
returnStructNumber()	private	Вспомогательный метод для clickHandler, возвращает номер стержня, на который нажал пользователь
save()	public	Сохраняет время и количество шагов для дальнейшей сериализации.
setControllerMain()	public	Передает ссылку на контроллер игры.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл 4. Поля класса Game

Имя	Назначение
animationInProgress	Определяет идет ли анимация
auto	Определяет включен ли автоматический сбор
blockSize	Высота кольца(диаметр)
controllerMain	Содержит контроллер игры
difficulty	Сложность игры, целое число означает количество колец
field	Содержит целочисленные переменные, где чем больше номер тем больше размер кольца(дублирует стек с кольцами, но вместо радиуса используется целые числа), необходимо для сравнения колец
first	Определяет первый раз или не первый

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

	заходит в метод clickHandler
maxSize	Максимальный размер в ширину(размер низа шеста)
moves	Количество ходов (для сохранения)
pField	Содержит стеки с кольцами, где номер стека соответствует левому, правому, и центральному шесту в такой последовательно сти в массиве
stractField	Содержит шесты
tempStackNumber	Хранит номер шеста с которого идет перенос, tempStackNumber = -1, когда никакой шест не выбран
time	Затраченное время (для сохранения)
topR	Радиус шеста
xCenter	Координата x центра
xLeft	Координата x левого штыря
xRight	Координата x правого штыря

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.5 Методы класса MaterialsGenerator

Имя	Модификатор доступа	Назначение
getFieldTexture()	Package private	Возвращает текстуру для шестов
GetMAterialByld()	Package private	Возвращает материал по id

Табл 6. Поля MaterialsGenerator

Имя	Назначение
materials	Лист для материалов

Табл.7 Методы Main

Имя	Модификатор доступа	Назначение
main()	public	Запускает поток программы
start()	public	Запускает начальный раздел меню

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.8 Методы Data

Имя	Модификатор доступа	Назначение
getMoves()	public	Возвращает количество шагов
getName()	public	Возвращает имя
getTime()	public	Возвращает время затраченное на прохождение

Табл. 9 Поля Data

Имя	Назначение
moves	Затранные шаги
name	Имя поставившего рекорд
time	Затраченное время

Табл.10 Методы RecordTableData

Имя	Модификатор доступа	Назначение
addNewRecord()	public	Добавляет новую ячейку
compare()	private	Сравнивает 2 времени, возвращает true, если первое больше второго
getList()	public	Возвращает лист для

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

		таблицы
getPlace()	public	Возвращает какое место займет переданная Data
Reset ()	public	Сбрасывает таблицу рекордов private void
Save()	private	Сохраняет таблицу рекордов
toString()	public	Приводит таблицу к строковому формату

Табл.10 поля RecordTableData

Имя	Назначение
difficulty	Сложность, равна количеству колец
table	Лист для таблицы рекордов

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.10 методы MenuController

Имя	Модификатор доступа	Назначение
closeStage()	private	Закрывает окно
onClickContinue()	public	Обработка нажатия на продолжить
onClickExit()	public	Событие нажатия на выход.
onClickNew()	public	Событие нажатия на "новая игра".
onClickTable()	public	Событие нажатия на таблицу рекордов.

Табл. 11 поля MenuController

Имя	Назначение
newButton	Кнопка новая игра
gamePane	Содержит AnchorPane с игрой
primaryStage	Главная сцена

Табл.12 методы Time

Имя	Модификатор доступа	Назначение
addTime()	public	Добавляет переданное время к общему затраченному времени
getMinutes()	public	Возвращает количество затраченных минут
getSeconds()	public	Возвращает количество затраченных секунд

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

toString()	public	Возвращает текстовое представление затраченного времени
getMseconds()	public	Возвращает количество затраченных миллисекунд

Табл. 13 поля Time

Имя	Назначение
minutes	Хранит минуты
mseconds	Хранит миллисекунды
seconds	Хранит секунды

Табл.14 методы Toroid

Имя	Модификатор доступа	Назначение
calcFacesPT()	private	Рассчитывает грани кольца
calcPoints()	private	Рассчитывает точки вершин на основе размеров тороида.
calcTexturePoints()	private	Рассчитывает места для текстур

Табл.15 поля Toroid

Имя	Назначение
NBR_RING_SEGMENTS	Количество ячеек в кольцах по умолчанию
NBR_TUBE_SEGMENTS	Количество ячеек в трубе по

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

	умолчанию
nbrRingSegments	Количество ячеек в кольцах
nbrRingSteps	Количество ячеек в кольцах + 1, необходимо для подсчета квадратов
nbrTubeSegments	Количество ячеек в трубах
nbrTubeSteps	Количество ячеек в трубах + 1, необходимо для подсчета квадратов
ringRad	Радиус кольца
tubeRadX	Радиус по x
tubeRadY	Радиус по y
tubeZeroPos	Начальная позиция кольца

Табл.16 методы WinnerController

Имя	Модификатор доступа	Назначение
closeStage()	<i>private</i>	Закрывает окно
Init()	<i>private</i>	инициализирует контроллер по сложности, количеству шагов и времени

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.17 поля WinnerController

Имя	Назначение
data	Ячейка таблицы
moveLabel	Отображает количество шагов
moves	Количество ходов
nameTextField	Нужен для написания имени пользователем
recordTableData	Таблица рекордов
saveButton	Кнопка при нажатии на которую сохраняет ячейку в таблицу
time	Затраченное время
timeLabel	Отображает затраченное время
toMenuButton	Переводит пользователя в меню
winnerPlace	Отображает какое место занял пользователь
gamePane	Содержит AnchorPane с игрой
primaryStage	Главня сцена

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.18 методы RecordTableController

Имя	Модификатор доступа	Назначение
choiceToInt()	private	Переводит слово в число соответствующее сложности
CloseStage()	public	Закрывает окно
Init()	public	Загружает нужную таблицу по сложности.
intToChoice()	private	Переводит сложность в слово
onClickChoice()	public	Событие выбора сложности
onClickReset()	public	Событие нажатия на сброс, сбрасывает таблицу.
onClickToMenu()	public	Событие нажатия на меню
Refresh()	public	Обновление таблицы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.19 поля RecordTableController

Имя	Назначение
choiceBox	Для выбора сложности
difficulty	Сложность
forChoiceBox	Лист со словесным выбором сложности для ChoiceBox
movesColumn	Колонка с шагами
nameColumn	Колонка с именами
recordTable	Отображаемая таблица рекордов
recordTableData	Таблица рекордов
timeColumn	Колонка со временем
toMenuButton	Переход в меню
gamePane	Содержит AnchorPane с игрой
primaryStage	Главня сцена

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.20 методы ControllerMain

Имя	Модификатор доступа	Назначение
addMouseEvent ()	private	Добавляет события для мыши
buildPane()	private	Создает AnchorPane и добавляет на него элементы
closeStage()	private	Закрывает окно
Init()	public	Инициализирует и создает view с полем по переданному количеству колец.
MoveHandler()	public	Получает из класса game результат передвижения элемента.
setCamera()	private	Возвращает камеру с выставленными параметрами.
StartTimer()	public	Запускает таймер

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.21 поля ControllerMain

Имя	Назначение
autoButton	Включает авто-сбор
camera	Камера для сцены
difficulty	Сложность, означает количество колец
game	Класс игры
MOUSE_SPEED	Скорость мыши
mouseDeltaX	Смещение мыши
mouseOldX	Позиция мыши до начало передвижения
mousePosX	Позиция мыши
moves	Затраченные шаги
movesLabel	Отображает затраченные шаги
pivot	Точка вокруг которой вращается камера
root3d	Группа со всеми элементами
rotateY	Класс вращения вокруг Y
scrollCounter	Счетчик скроллинга
stage	Главная сцена
time	Лист для таблицы

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

	рекордов
timeLabel	Отображает затраченное время
timer	Считает затраченное время
gamePane	Содержит AnchorPane с игрой
primaryStage	Главня сцена

Табл.22 методы GamePane

Имя	Модификатор доступа	Назначение
setCamera	private	Создает и настраивает камеру
returnGame()	public	Возвращает поле с игрой
returnPane()	public	Возвращает AnchorPane с добавленным root и игрой
setRoot()	public	Добавляет Root на AnchorPane
clearPain	public	Убирает root с AnchorPane
Blur()	public	Включает анимацию для создания блюр эффекта
unBlur	public	Включает анимацию для убирания блюр эффекта

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.23 поля GamePane

Имя	Назначение
game	Класс в котором находится игра
pane	AnchorPane на который собираются все игровые элементы
root	Элементы отображаемые на экране, но не относящиеся к игре
gb	Класс блюра
rotateY	Класс при помощи которого происходит вращение.
camera	Камера для сцены

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Табл.24 методы `PauseMenuController`

Имя	Модификатор доступа	Назначение
<code>Init()</code>	<code>public</code>	Инициализирует все поля класса
<code>onClickContinue</code>	<code>public</code>	Событие, срабатывает когда пользователь нажимает продолжить
<code>onClickToMenu</code>	<code>public</code>	Событие, срабатывает когда пользователь нажимает в меню

Табл.25 поля `PauseMenuController`

Имя	Назначение
<code>gamePane</code>	Содержит <code>AnchorPane</code> с игрой
<code>primaryStage</code>	Главня сцена
<code>controllerMain</code>	Ссылка на контроллер игры

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.01-01 81 01-1				
Инв. № подл.	Подп. и дата	Взам. инв №	Инв. № дубл.	Подп. и дата

Лист регистрации изменений

[illegible]