Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Программное обеспечение автоматизированных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту**

по дисциплине «Объектно-ориентированный анализ и программирование»

на тему: «Проектирование и реализация программы с использованием объектно-ориентированного подхода»

(индивидуальное задание – вариант №1, подвариант №1)

Студент: Ляшенко А.С.

Группа: ПрИн-366

Работа зачтена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «       »                            20\_\_ г.

Руководитель проекта, нормоконтроллер \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Литовкин Д.В.

Члены комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись и дата подписания) (инициалы и фамилия)

Волгоград 2015 г.

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Направление 09.03.04 «Программная инженерия»   
Кафедра «Программное обеспечение автоматизированных систем»

Дисциплина «Объектно-ориентированный анализ и программирование»

Утверждаю

Зав. кафедрой \_\_\_\_\_\_\_\_\_\_ Дворянкин А.М.

**Задание**

**на курсовой проект**

Студент: Ляшенко А.С.

Группа: ПрИн-366

1. Тема: «Проектирование и реализация программы с использованием объектно-ориентированного подхода» (индивидуальное задание – вариант №1, подвариант №1)

Утверждена приказом от «02» ноября 2015г. № 1489-ст

2. Срок представления работы к защите «      »                         20    г.

3. Содержание пояснительной записки:

формулировка задания, требования к программе, объектная модель предметной области, паспорта классов, код программы

4. Перечень графического материала:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

5. Дата выдачи задания «05» ноября 2015 г.

Руководитель проекта: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Литовкин Д.В.

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ляшенко А.С.

«      »                         2015 г.

**Содержание**

1 Формулировка задания 4

2 Требования к программе 4

3 Аналитическая модель 8

4 Паспорта классов 16

5 Список использованной литературы и других источников 16

Приложение A. Исходный код программы 17

# 1 Формулировка задания

Правила игры «Коза и капуста»:

* имеется загон NxM клеток, в котором находится коза, капуста, ящики и стены (располагаются внутри клеток);
* ящики и стены, возможно, образуют непроходимый лабиринт;
* цель козы — добраться до капусты, двигая ящики;
* коза может двигать только один ящик от себя или на себя;
* количество шагов козы ограничено;
* обстановка игры считывается из файла;

Вариативность: на поле могут появляться объекты, которые могут  или не могут быть передвинуты козой. Кроме того, объекты можно активировать, применив ранее собранные объекты. Результатом активации является изменение положения существующих объектов, появление новых объектов, уничтожение имеющихся.

Реализовать: телепортатор козы, который активируется ключом. Ключ можно найти в ящике, который нужно открыть, затем ящик нужно сдвинуть и взять ключ с поля.

# 2 Требования к программе

**2.1 Функциональные требования**

**Сценарий «Играть»**

**Главный успешный сценарий:**

1. Приложение загружает первый уровень.

2. Приложение показывает пользователю историю уровня игры.

3. Приложение информирует пользователя о способах управления в игре.

4. Приложение передает управление игроку.

5. До тех пор, пока не закончились уровни и уровень пройден успешно…

5.1. До тех пор, пока игрок не прошел уровень или отказался от прохождения

5.1.1. Игрок играет уровень.

5.1.2. Если уровень пройден, Приложение запускает следующий уровень.

5.1.3. Если уровень не пройден, Приложение предлагает игроку пройти уровень заново.

5.1.3.1. Если игрок соглашается, Приложение запускает уровень снова, иначе завершает игру.

5.2. Если игра завершена, Приложение информирует пользователя об этом.

**Расширения:**

1а. В системе имеется сохраненная информация о пройденных уровнях.

.1: Приложение загружает последний уровень, на котором игрок закончил играть в предыдущий раз и переходим к 4 шагу главного успешного сценария.

1б. Игрок выбирает конкретный уровень.

.1: Приложение загружает выбранный уровень.

.2: Приложение передает управление игроку.

.3: Игрок играет один уровень.

.4: После завершения уровня Приложение показывает игроку информацию об успешности прохождения.

**Сценарий «Выйти из игры»**

**Триггер**: Игрок инициировал завершение работы приложения

**Главный успешный сценарий:**

1. Приложение выясняет у игрока, действительно ли он хочет покинуть игру.

2. Приложение сохраняет текущий прогресс игрока.

3. Система закрывает окно игры.

**Расширения:**

1а. Игрок отменил выход.

.1: Приложение отменяет действие по выходу из игры.

**Сценарий «Играть уровень»**

**Главный успешный сценарий**

1. Пока игрок не достиг козлом клетки с эликсиром или у него не кончились ходы

1.1 Игрок делает ход козлом

2. Если козел достиг клетки с зельем, приложение оповещает игрока о успешном завершении уровня.

**Расширения:**

2. Игрок не смог пройти уровень – закончились ходы и эликсир не подобран.

.1: Приложение оповещает игрока о том, что уровень не пройден.

**Сценарий «Сделать ход козлом»**

**Предусловие:** загружен один из уровней игры.

**Главный успешный сценарий**

1. Игрок задает козлу направление движения.

2. Приложение поворачивает козла в заданную сторону, и, если ничего не мешает пройти, перемещает козла на одну клетку по этому направлению.

**Расширения:**

2а. Игрок подбирает предмет.

.1 Если на клетке, на которую должен перейти козел есть подбираемый предмет, приложение добавляет его в список предметов, убирает предмет с клетки и перемещает козла на него.

2b. Игрок активирует сундук в соседней клетке.

.2 Если на соседней по направлению клетке есть сундук, то выполняется сценарий «Активировать сундук», иначе ход игрока игнорируется.

2c. Игрок передвигает стол.

.1 Выполняется сценарий «Сдвинуть стол».

2e. Игрок активирует телепорт.

.1 Приложение проверяет наличие заклинания в инвентаре, и если оно есть, то перемещает козла на заданную для активированного телепорта клетку.

.2 Иначе, ход игнорируется.

**Сценарий «Активировать объект»**

**Предусловие**: на соседней клетке по выбранном направлению есть объект, с которым игрок может взаимодействовать.

**Главный успешный сценарий**

1 Если у игрока есть заклинание (аналог ключа) для взаимодействия с объектом, приложение удаляет его из списка предметов игрока.

2 Приложение удаляет сундук с клетки

3 Приложение перемещает козла на клетку, с которой был удален сундук

**Расширение.**

1а. У игрока нет нужного заклинания

.1 Если у игрока нет заклинания для открытия сундука, ход игрока игнорируется

2b. Сундук содержал предметы.

.1 При выполнении пункта 3 главного сценария, приложение добавляет содержимое сундука в список предметов.

**Сценарий «Сдвинуть стол»**

**Предусловие:** козел должен быть в соседней клетке с предметом и смотреть на него**.**

**Главный успешный сценарий:**

1. Игрок толкает стол вперед по направлению взгляда козла.

2. Приложение проверяет соседнюю от стола клетку по направлению взгляда козы.

3. Если клетка пуста, приложение перемещает козла с столом на соседнюю клетку в направлении взгляда козла, иначе приложение игнорирует ход игрока.

**Расширения:**

1а. Игрок тянет деревянный стол назад по направлению, противоположному взгляду козла.

.1: Приложение проверяет соседнюю клетку сзади козла по направлению, противоположному взгляду козла.

.2: Если клетка пуста, приложение перемещает козла с столом на соседнюю клетку по направлению, противоположному взгляду козла, иначе приложение игнорирует ход игрока.

**2.2 Нефункциональные требования**

1. Программа должна быть реализована на языке Java SE 7 с использованием стандартных библиотек.
2. В качестве среды разработки необходимо использовать NetBeans IDE.
3. Форматирование исходного кода программы должно соответствовать Java Code Conventions, September 12, 1997.

# 3 Аналитическая модель

**3.1 Словарь данных**

Игровое поле - прямоугольная область, которое поделено на ячейки. Размеры поля задаются в ячейках. В ячейках поля располагаются эликсир, столы, стены, сундуки, книги, телепорты.

Игровой объект – объект, который может находится в ячейке. Одновременно, в ячейке может находится только один объект. Некоторые объекты козел может сдвигать или активировать.

Игровой предмет – также, как и объект, предмет может находится в ячейке в единичном экземпляре. В отличии от объекта, сдвинуть или активировать предмет нельзя, козел может только подобрать его в инвентарь.

Эликсир – игровой предмет, дойдя до которого козел выигрывает. Эликсир занимает одну ячейку поля, и он единственный на поле.

Стол – игровой объект, который козел может двигать. Стол занимает одну ячейку поля, в одной ячейке может быть только один стол.

Неподвижный стол – игровой объект, который козел не может двигать. Стол занимает одну ячейку поля, в одной ячейке может быть только один стол.

Стена – игровой объект, являющийся препятствием и располагающийся в ячейке. Не может быть сдвинуто козлом. В одной ячейке может находиться только одна стена.

Ячейка поля – область поля, которую могут занимать козел, игровые объекты и игровые предметы. Объекты не могут занимать пространство между ячейками.

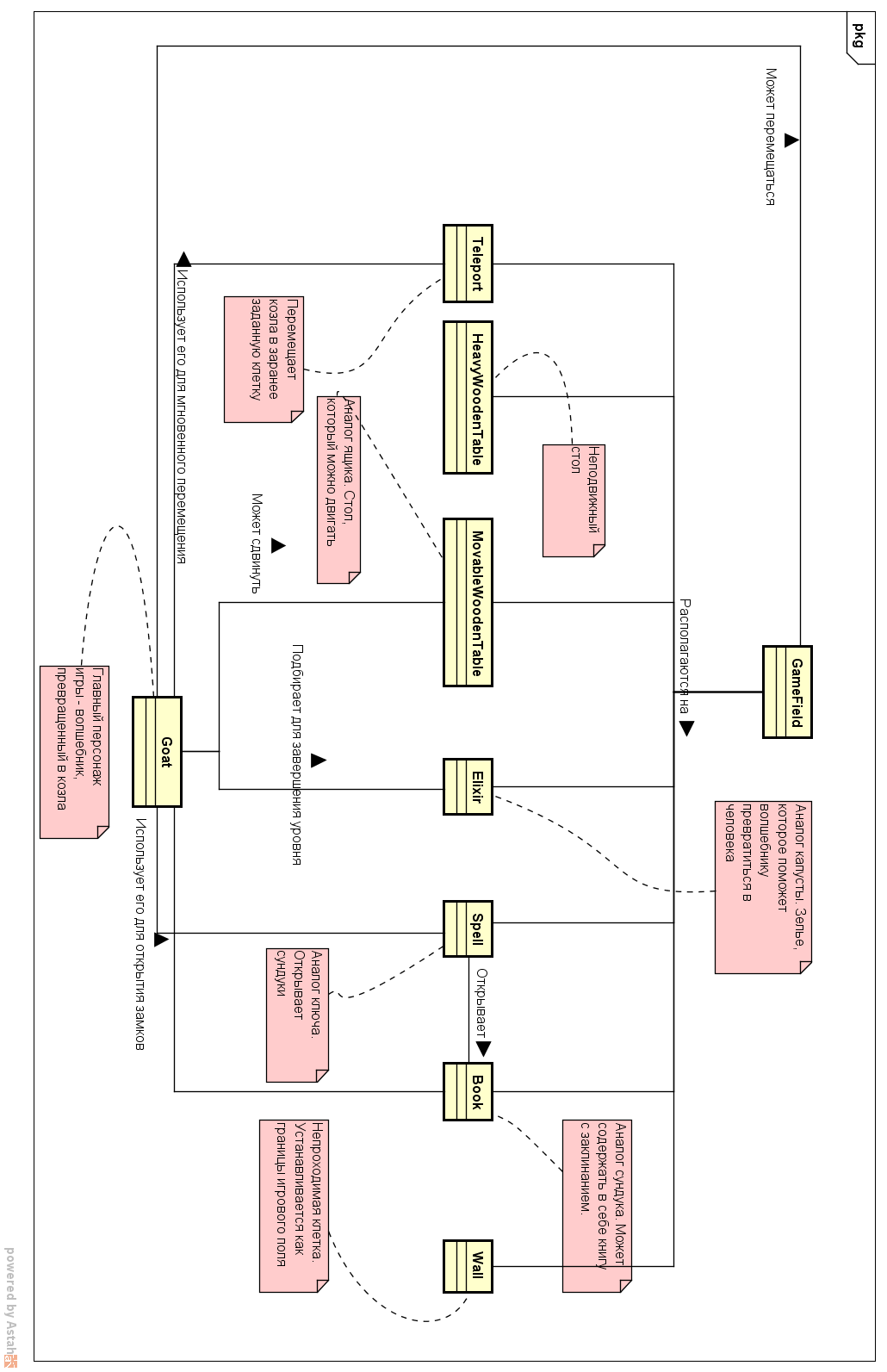
Козел – объект, который может перемещаться между соседними (четырьмя) ячейками. У козла есть определенное число ходов, по истечению которого, игра прекращается. Козел может подобрать эликсир с поля (игра завершится) или заклинание (положится в инвентарь), при этом поднятый предмет убирается с поля. Козел может активировать некоторые объекты, если у него есть соответствующее заклинание. Козел может сдвинуть стол, если нет объектов, препятствующих движению. Козел не может выйти за пределы поля.

Заклинание – игровой предмет, который располагается в книге. Если козел активирует книгу, заклинание автоматически добавляется в инвентарь. Если козел активирует объект, для которого требуется заклинание, подходящее заклинание изымается из инвентаря и удаляется.

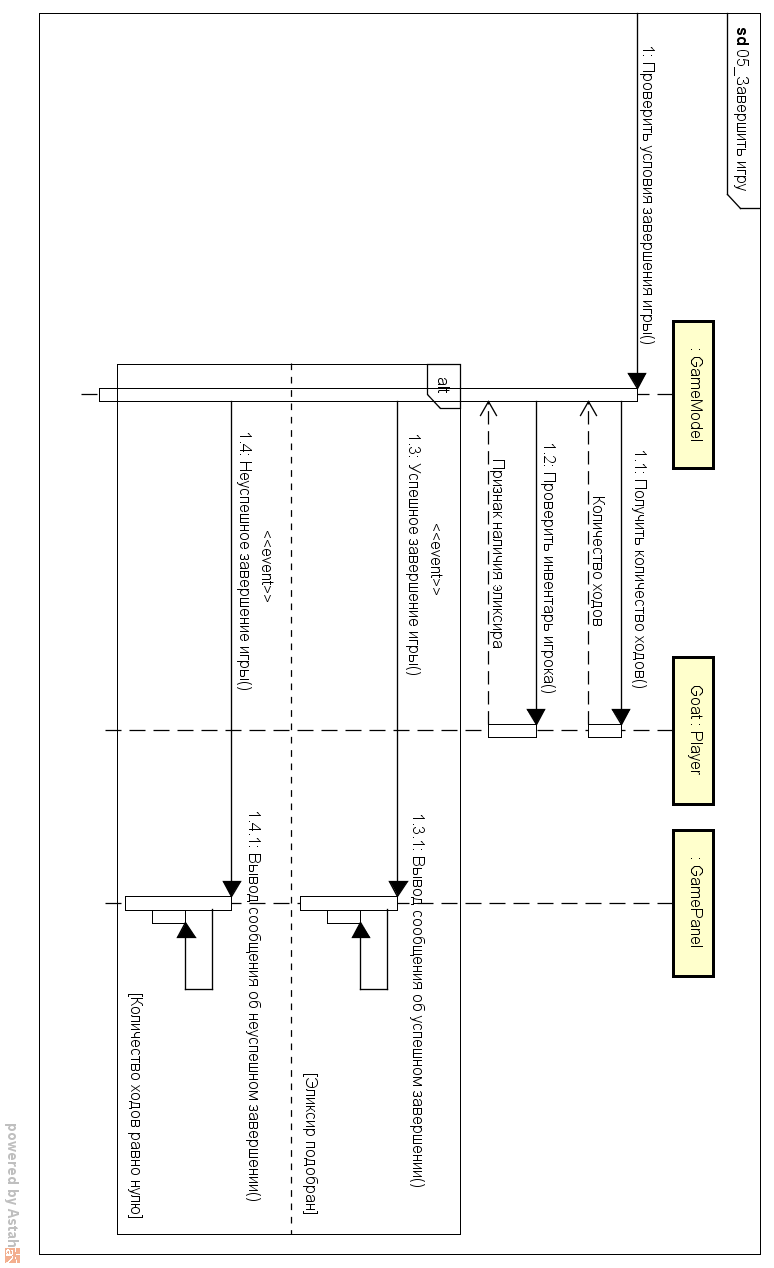
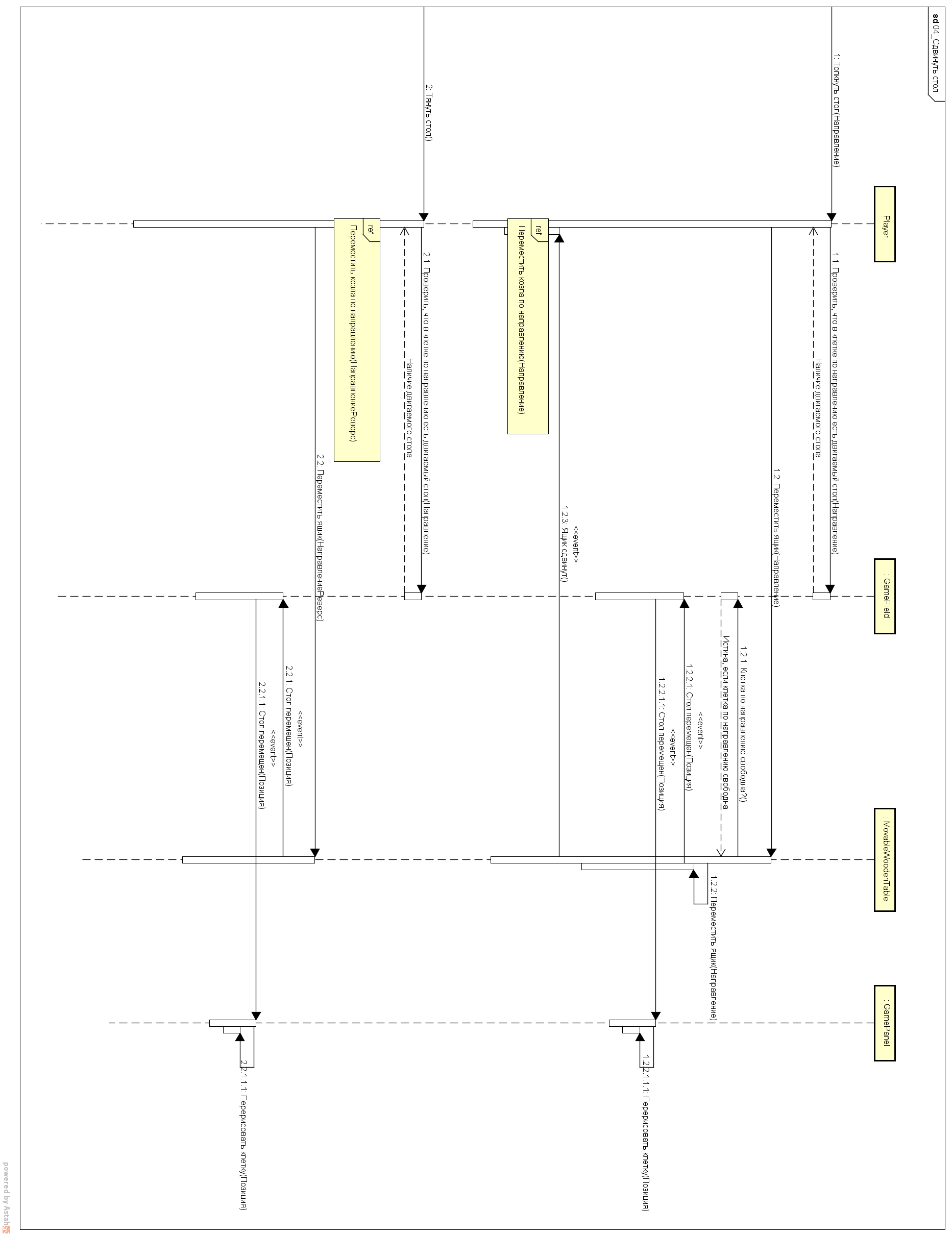
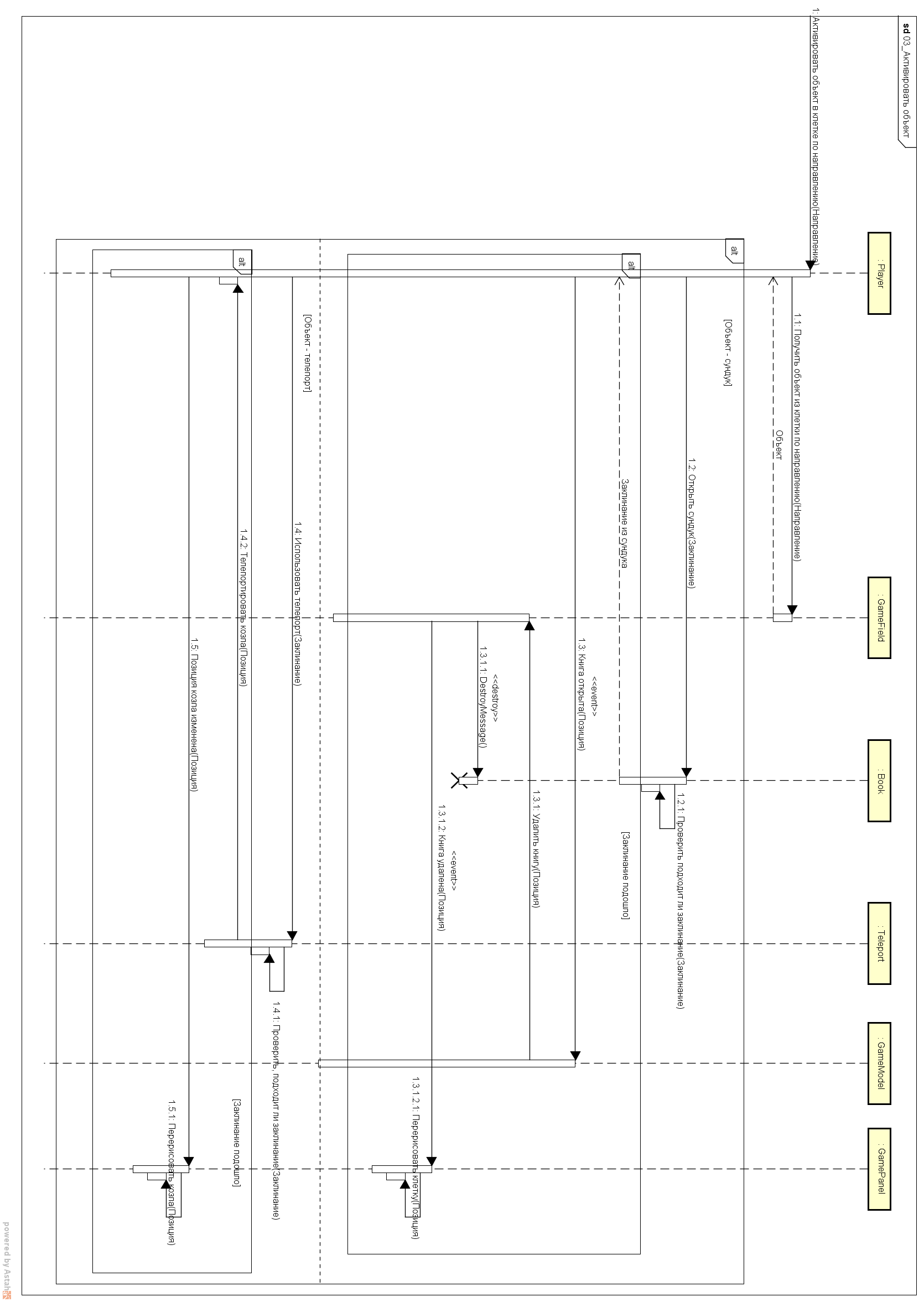
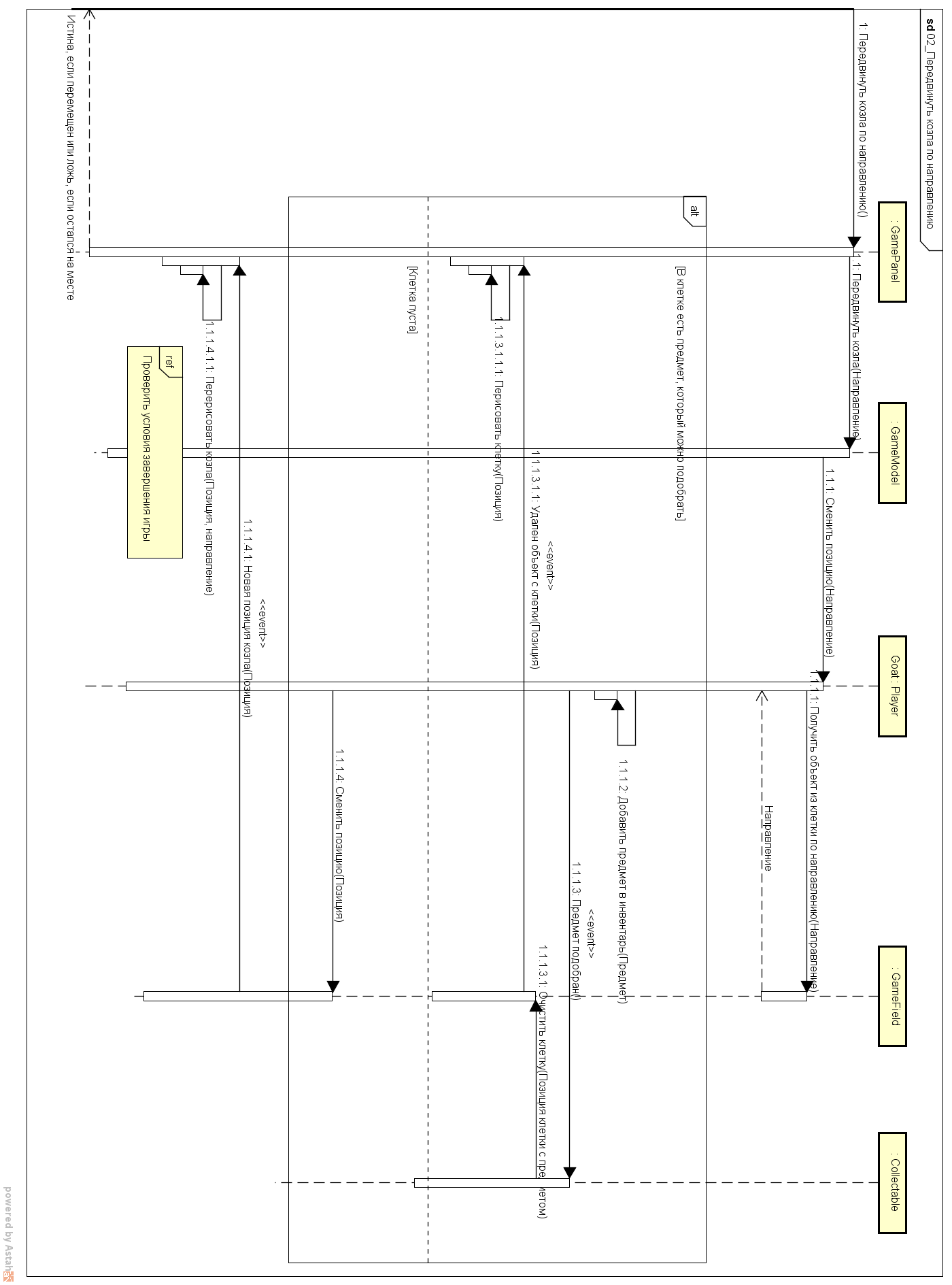
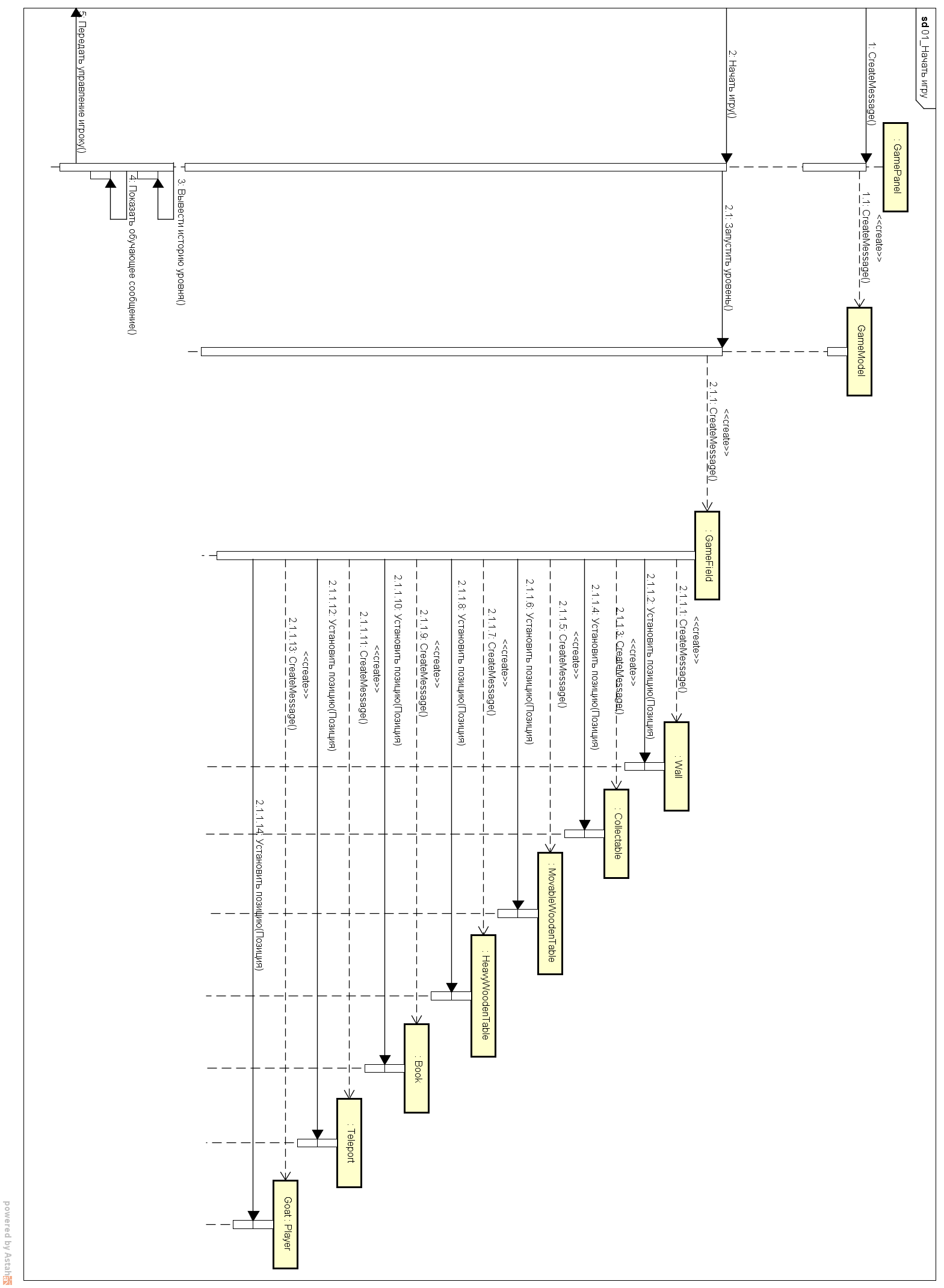
Книга – игровой объект, который может располагаться в ячейке. При активации книги, заклинание, которое она содержит, автоматически добавляется в инвентарь козла, а сама книга уничтожается. В ячейке может быть только одна книга, а сама книга может содержать только одно заклинание.

Телепорт – игровой объект, который может располагаться в ячейке. При активации проверяет инвентарь козла на наличие заклинания-ключа, и если находит его, то переносит козла на другое место, удалив заклинание из инвентаря. В одной ячейке может располагаться только один телепорт.

**3.2 Диаграмма классов**



**3.3 Диаграммы последовательности**



# 4 Паспорта классов

**Class GameField**

public class GameField

extends java.lang.Object

Класс представляет игровую поле. Построен на основе полиморфного контейнера игровых объектов с добавлением логики поля.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private java.util.Map <java.lang.Class,java.util.ArrayList<[**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)>> | [**\_gameObjects**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#Z:Z_gameObjects)  Игровые объекты поля |
| private int | [**\_height**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#Z:Z_height)  Высота поля |
| private java.util.ArrayList<java.lang.Integer> | [**\_spellIds**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#Z:Z_spellIds) |
| private int | [**\_width**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#Z:Z_width)  Ширина поля |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| boolean | [**addObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#addObject-java.awt.Point-badmagic.model.gameobjects.GameObject-)(java.awt.Point pos, [**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html) obj)  Метод добавления объекта на поле. |
| void | [**clear**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#clear--)()  Метод очистки игрового поля. |
| void | [**clearSpellIdsList**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#clearSpellIdsList--)() |
| int | [**getHeight**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#getHeight--)()  Метод получения высоты поля |
| java.awt.Point | [**getNextPos**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#getNextPos-java.awt.Point-badmagic.navigation.Direction-)(java.awt.Point currentPos, [**Direction**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\navigation\Direction.html) direction)  Метод получения следующей позиции в указанном направлнеии. |
| java.util.ArrayList<[**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)> | [**getObjects**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#getObjects--)()  Метод получения объектов поля. |
| java.util.ArrayList<[**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)> | [**getObjects**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#getObjects-java.lang.Class-)(java.lang.Class objType)  Метод получения объектов поля заданного типа. |
| java.util.ArrayList<[**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)> | [**getObjects**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#getObjects-java.lang.Class-java.awt.Point-)(java.lang.Class objType, java.awt.Point pos)  Метод получения объектов заданного типа и позиции. |
| java.util.ArrayList<[**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)> | [**getObjects**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#getObjects-java.awt.Point-)(java.awt.Point pos)  Метод получения объетов поля с заданной позицией. |
| int | [**getWidth**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#getWidth--)()  Метод получения ширины поля |
| boolean | [**isNextPosEmpty**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#isNextPosEmpty-java.awt.Point-badmagic.navigation.Direction-)(java.awt.Point currentPos,[**Direction**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\navigation\Direction.html) direction)  Метод проверки следующей позиции в направлении на незанятость. |
| boolean | [**isNextPosHasCollectable**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#isNextPosHasCollectable-java.awt.Point-badmagic.navigation.Direction-)(java.awt.Point currentPos,[**Direction**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\navigation\Direction.html) direction) |
| boolean | [**isPosEmpty**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#isPosEmpty-java.awt.Point-)(java.awt.Point pos)  Метод проверки позиции на незанятость. |
| boolean | [**isPosHasCollectable**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#isPosHasCollectable-java.awt.Point-)(java.awt.Point pos) |
| boolean | [**isPositionUnique**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#isPositionUnique-java.awt.Point-)(java.awt.Point pos)  Метод проверки позиции на уникальность - незанятость. |
| boolean | [**isSpellIdUnique**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#isSpellIdUnique-int-)(int id) |
| boolean | [**removeObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#removeObject-badmagic.model.gameobjects.GameObject-)(**[GameObject](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\gameobjects\\GameObject.html" \o "class in badmagic.model.gameobjects)** obj)  Метод удаления объекта с поля. |
| void | [**setHeight**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#setHeight-int-)(int height)  Метод установки высоты поля. |
| void | [**setSize**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#setSize-int-int-)(int width, int height)  Метод установления размера поля |
| void | [**setSpellIdsList**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#setSpellIdsList-java.util.ArrayList-)(java.util.ArrayList<java.lang.Integer> list) |
| void | [**setWidth**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html#setWidth-int-)(int width)  Метод установки ширины поля. |

**Class GameModel**

public class **GameModel**

extends java.lang.Object

Класс представляет модель игры. Отвечает за логику игры: загрузку уровней, режимы игры, переход между уровнями, определение завершенности уровня. сохранение и загрузка информации о прохождении

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private int | [**\_currentLevel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_currentLevel)  Номер текущего уровня |
| private [**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) | [**\_field**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_field)  Игровое поле |
| private [**GameModel.GameMode**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.GameMode.html) | [**\_gameMode**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_gameMode)  Текущий игровой режим |
| private static int | [**\_lastCompletedLevel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_lastCompletedLevel)  Номер последнего пройденного уровня |
| private java.util.ArrayList<java.lang.String> | [**\_levelNames**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_levelNames)  Список имен уровней игры |
| private static java.util.ArrayList<[**Level**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html)> | [**\_levels**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_levels)  Список уровней игры |
| private [**GameModel.LevelStatus**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.LevelStatus.html) | [**\_levelStatus**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_levelStatus)  Статус текущего уровня |
| private java.util.ArrayList | [**\_modelListenerList**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_modelListenerList)  Список слушателей модели |
| private [**GameModel.ObjectsObserver**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.ObjectsObserver.html) | [**\_objectsObserver**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_objectsObserver)  Слушатель игровых объектов |
| private [**Player**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html) | [**\_player**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#Z:Z_player)  Игрок |
| private static java.lang.String | [**PATH\_TO\_GAME\_PROGRESS\_FILE**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#PATH_TO_GAME_PROGRESS_FILE)  Путь к файлу с информацией о прогрессе игры |
| private static java.lang.String | [**PATH\_TO\_LEVELS\_INFO\_FILE**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#PATH_TO_LEVELS_INFO_FILE)  Путь к файлу с информацией об уровнях |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**GameModel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#GameModel--)()  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**addModelListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#addModelListener-badmagic.events.ModelListener-)(**[ModelListener](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\events\\ModelListener.html" \o "interface in badmagic.events)** l)  Метод добавления слушателя модели. |
| private void | [**checkIfLevelIsFinished**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#checkIfLevelIsFinished--)()  Метод проверки, завершен ли уровень. |
| void | [**continueCareer**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#continueCareer--)()  Метод продожения игры в режиме карьеры Загружает из номер последнего пройденного уровня, загружает этот уровень и дальше игра продолжается пока есть уровни. |
| protected void | [**fireLevelCompleted**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#fireLevelCompleted--)()  Метод испускания сигнала об успешном прохождении уровня. |
| protected void | [**fireLevelFailed**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#fireLevelFailed--)()  Метод испускания сигнала о неуспешном прохождении уровня. |
| [**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) | [**getField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#getField--)()  Метод получения текущего игрового поля. |
| [**GameModel.GameMode**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.GameMode.html) | [**getGameMode**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#getGameMode--)()  Метод получения текущего игрового режима. |
| java.lang.String | [**getLevelName**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#getLevelName--)()  Метод получения имени текущего уровня. |
| static java.util.ArrayList<java.lang.String> | [**getLevelNames**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#getLevelNames--)()  Статический метод получения имен уровней игры. |
| [**GameModel.LevelStatus**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.LevelStatus.html) | [**getLevelStatus**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#getLevelStatus--)()  Метод получения статуса текущего уровня. |
| int | [**getMoves**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#getMoves--)()  Метод получения количества ходов текущего уровня. |
| [**Player**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html) | [**getPlayer**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#getPlayer--)()  Метод получения текущего игрока. |
| private void | [**initializePlayer**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#initializePlayer--)()  Метод инициализации текущего игрока. |
| boolean | [**isLastLevel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#isLastLevel--)()  Метод,получения информации о том, является ли текущий уровень последним. |
| void | [**loadGameProgress**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#loadGameProgress--)()  Метод загрузки информации о последнем пройденном уровне из файла. |
| private java.util.ArrayList<java.lang.String> | [**loadLevelNames**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#loadLevelNames--)()  Статический метод загрузки информации об уровнях игры. |
| private void | [**loadLevels**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#loadLevels--)()  Метод загрузки уровней игры. |
| void | [**nextLevel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#nextLevel--)()  Метод перехода к следующему уровню игры. |
| void | [**oneLevelMode**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#oneLevelMode-int-)(int levelNumber)  Метод начала игры в режиме одного уровня. |
| void | [**removeModelListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#removeModelListener-badmagic.events.ModelListener-)(**[ModelListener](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\events\\ModelListener.html" \o "interface in badmagic.events)** l)  Метод удаления слушаетля модели. |
| private void | [**reset**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#reset--)()  Метод сброса модели.Очищает поля класса. |
| static void | [**saveGameProgress**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#saveGameProgress--)()  Метод сохранения информации о последнем пройденном уровне в файл. |
| private void | [**setCurrentLevel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#setCurrentLevel-int-)(int levelNumber)  Метод установки уровня номер leveNumber как текущего. |
| private void | [**startLevel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#startLevel-int-)(int levelNumber)  Метод начала уровня номер levelNumber. |
| void | [**startNewCareer**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#startNewCareer--)()  Метод начала новой игры в режиме карьеры. |
| void | [**tryAgain**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html#tryAgain--)()  Метод повторного прохождения уровня. |

**Class Bookshelf**

public class **Bookshelf**

extends [InteractiveObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\InteractiveObject.html)

Класс реализующий книжную полку (сундук-хранилище). Является наследником InteractiveObject. Полка может содержать одно заклинание (в виде численного идентификатора). После активации удаляется с поля, на своем месте оставляя объект-заклинание Имеет собственное графическое представление

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private int | [**\_spellId**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#Z:Z_spellId)  Хранимое заклинание (-1 если отсутствует) |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#PIC)  Путь к файлу с изображением |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Bookshelf**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#Bookshelf-badmagic.model.GameField-)(**[GameField](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\GameField.html" \o "class in badmagic.model)** field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| boolean | [**activate**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#activate--)()  Метод, активирующий объект. |
| protected void | [**loadPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#loadPic--)()  Метод загрузки изображения объекта. |
| void | [**paint**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#paint-java.awt.Graphics-java.awt.Point-)(java.awt.Graphics g, java.awt.Point pos)  Метод отрисовки объекта. |
| void | [**setLock**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#setLock-int-)(int key)  Метод, устанавливающий ключ для объекта |
| void | [**setSpell**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#setSpell-int-)(int spellId) |
| boolean | [**unlock**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Bookshelf.html#unlock-badmagic.model.gameobjects.CollectableObject-)(**[CollectableObject](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\gameobjects\\CollectableObject.html" \o "class in badmagic.model.gameobjects)** key)  Метод для открытия объекта (снятия с него замка) |

**Class CollectableObject**

public abstract class **CollectableObject**

extends [GameObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)

Абстрактный класс - подбираемый игровой объект. Наследники класса могут быть подобраны игроком. Их нельзя сдвинуть, но они могут быть перекрыты наследником MovableObject С ними нельзя взаимодействовать как с InteractiveObject. Чтобы подобрать наследника, игрок ддолжен пройти через него. Наследник класса GameObject.

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**CollectableObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\CollectableObject.html#CollectableObject-badmagic.model.GameField-)([**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) field)  Конструктор класса. |

**Class Elixir**

public class **Elixir**

extends [CollectableObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\CollectableObject.html)

Класс представляет игровой объект - элексир. Наследник класса CollectableObject. Реализует абстрактные методы. Имеет образ - изображение.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Elixir.html#PIC)  Путь к файлу с изображением |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Elixir**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Elixir.html#Elixir-badmagic.model.GameField-)(**[GameField](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\GameField.html" \o "class in badmagic.model)** field)  Конструктор класса. |

**Class GameObject**

public abstract class **GameObject**

extends java.lang.Object

Абстрактный класс - игровой объект.  
Хранит позицию игрового объекта и ссылку на класс игрового поля.  
Может посылать сообщения.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| protected [**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) | [**\_field**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#Z:Z_field)  Ссылка на игровое поле |
| protected java.awt.image.BufferedImage | [**\_image**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#Z:Z_image)  Изображение объекта |
| private java.util.ArrayList | [**\_listenerList**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#Z:Z_listenerList)  Список слушателей игрового объекта |
| protected java.awt.Point | [**\_position**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#Z:Z_position)  Позиция игрового объекта |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#GameObject-badmagic.model.GameField-)([**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**addObjectListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#addObjectListener-badmagic.events.GameObjectListener-)(**[GameObjectListener](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\events\\GameObjectListener.html" \o "interface in badmagic.events)** l)  Метод добавления слушателя игрового объекта. |
| protected void | [**fireObjectMoved**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#fireObjectMoved--)()  Метод испускания сигнала о перемещении объекта. |
| java.awt.Point | [**getPosition**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#getPosition--)()  Метод получения позиции объекта. |
| protected abstract void | [**loadPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#loadPic--)()  Абстрактный метод загрузки изображения объекта. |
| abstract void | [**paint**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#paint-java.awt.Graphics-java.awt.Point-)(java.awt.Graphics g, java.awt.Point pos)  Абстрактный метод отрисовки объекта. |
| void | [**removeObjectListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#removeObjectListener-badmagic.events.GameObjectListener-)(**[GameObjectListener](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\events\\GameObjectListener.html" \o "interface in badmagic.events)** l)  Метод удаления слушаетля игрового объекта. |
| boolean | [**setPosition**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#setPosition-java.awt.Point-)(java.awt.Point pos)  Метод установки позиции игрового объекта. |
| void | [**unsetPosition**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html#unsetPosition--)()  Метод обнуления позиции игрового объекта. |

**Class GamePanel**

public class **GamePanel**

extends javax.swing.JPanel

Класс представляет игровую панель. Отображает текущее игровое поле, показывает перемещения объектов, информацию о текущем уровне, выводит информирующие сообщения при прохождении уровней, сожержит элементы управления - кнопки.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private static java.awt.Font | [**\_buttonFont**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_buttonFont)  Шрифт кнопок |
| private java.awt.image.BufferedImage | [**\_cellPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_cellPic)  Изображение клетки |
| private [**GamePanel.ClickListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.ClickListener.html) | [**\_clickListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_clickListener)  Слушатель мыши |
| private int | [**\_fieldStartX**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_fieldStartX)  Координата начала отрисовки игрового поля по Х |
| private int | [**\_fieldStartY**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_fieldStartY)  Координата начала отрисовки игрового поля по Y |
| private java.awt.image.BufferedImage | [**\_infoPanel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_infoPanel)  Изображение информационной панели |
| private [**GamePanel.KeyHandler**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.KeyHandler.html) | [**\_keyHandler**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_keyHandler)  Слушатель клавиатуры |
| private static java.awt.Font | [**\_levelFont**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_levelFont)  Шрифт уровня |
| private java.util.ArrayList | [**\_listenerList**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_listenerList)  Список слушателей панели |
| private static java.awt.Rectangle | [**\_mainMenuBtn**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_mainMenuBtn)  "Кнопка" перехода в главное меню |
| private [**GameModel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html) | [**\_model**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_model)  Игровая модель |
| private static java.awt.Rectangle | [**\_nextActionBtn**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_nextActionBtn)  "Кнопка" следующего действия при прохождении уровня |
| private static java.awt.Rectangle | [**\_quitGameBtn**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_quitGameBtn)  "Кнопка" выхода из игры |
| private static java.awt.Font | [**\_simpleFont**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#Z:Z_simpleFont)  Обычный шрифт |
| private static java.awt.Color | [**BACKGROUND\_COLOR**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#BACKGROUND_COLOR)  Цвет фона |
| private static int | [**BUTTON\_HEIGHT**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#BUTTON_HEIGHT)  Высота кнопки |
| private static int | [**BUTTON\_WIDTH**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#BUTTON_WIDTH)  Ширина кнопки |
| private static int | [**CELL\_SIZE**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#CELL_SIZE)  Размер клетки |
| private static java.awt.Color | [**FONT\_COLOR**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#FONT_COLOR)  Цвет шрифта |
| private static float | [**FONT\_SIZE**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#FONT_SIZE)  Размер шрифта |
| private static java.lang.String | [**FONT\_TYPE**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#FONT_TYPE)  Тип шрифта |
| private static java.lang.String | [**INFO\_PANEL**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#INFO_PANEL)  Путь к файлу с изображением информационной панели |
| private static int | [**INFO\_PANEL\_WIDTH**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#INFO_PANEL_WIDTH)  Ширина информационной панели |
| private static java.lang.String | [**LEVEL\_FONT\_PATH**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#LEVEL_FONT_PATH)  Путь к шрифту |
| private static java.awt.Color | [**OBJECTS\_BORDER\_COLOR**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#OBJECTS_BORDER_COLOR)  Цвет рамок |
| private static int | [**OBJECTS\_BORDER\_WIDTH**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#OBJECTS_BORDER_WIDTH)  Ширина рамки |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#PIC)  Путь к файлу с изображением клетки |
| private static int | [**RES\_BOX\_BUTTON\_HEIGHT**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#RES_BOX_BUTTON_HEIGHT)  Высота кнопок блока результатов |
| private static int | [**RES\_BOX\_BUTTON\_WIDTH**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#RES_BOX_BUTTON_WIDTH)  Ширина кнопок блока результатов |
| private static int | [**RESULT\_BOX\_HEIGHT**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#RESULT_BOX_HEIGHT)  Высота блока результатов |
| private static int | [**RESULT\_BOX\_WIDTH**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#RESULT_BOX_WIDTH)  Ширина бока результатов |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**GamePanel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#GamePanel-badmagic.model.GameModel-)([**GameModel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameModel.html) model)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**addPanelListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#addPanelListener-badmagic.events.PanelListener-)(**[PanelListener](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\events\\PanelListener.html" \o "interface in badmagic.events)** l)  Метод добавления слушателя панели. |
| void | [**continueCareer**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#continueCareer--)()  Метод продолжения игры в режиме карьеры. |
| protected void | [**fireMainMenuClicked**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#fireMainMenuClicked--)()  Метод испускания сигнала о нажатии кнопки "Главное меню". |
| private java.awt.Point | [**getPanelPosition**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#getPanelPosition-java.awt.Point-)(java.awt.Point logPos)  Метод получения координат панели для отрисовки по логическим координатам. |
| private void | [**loadResources**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#loadResources--)()  Метод загрузки ресурсов панели. |
| void | [**oneLevelMode**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#oneLevelMode-int-)(int levelNumber)  Метод начала новой игры в режиме прохождения одного уровня. |
| private void | [**paintCenteredString**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#paintCenteredString-java.awt.Graphics-java.awt.Rectangle-java.lang.String-int-int-)(java.awt.Graphics g, java.awt.Rectangle area, java.lang.String string, int yStart, int xStart)  Метод отрисоки строки посередине прямоугольной области. |
| void | [**paintComponent**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#paintComponent-java.awt.Graphics-)(java.awt.Graphics g)  Метод отрисовки компонентов панели. |
| private void | [**paintGrid**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#paintGrid-java.awt.Graphics-)(java.awt.Graphics g)  Метод отрисовки сетки игрового поля. |
| private void | [**paintInfoPanel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#paintInfoPanel-java.awt.Graphics-)(java.awt.Graphics g)  Метод отрисовки информационной панели. |
| private void | [**paintObjects**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#paintObjects-java.awt.Graphics-)(java.awt.Graphics g)  Метод отрисовки игровых объектов поля. |
| private void | [**paintResultBox**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#paintResultBox-java.awt.Graphics-)(java.awt.Graphics g)  Метод отрисовки блока результатов прохождения уровня. |
| void | [**removePanelListener**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#removePanelListener-badmagic.events.PanelListener-)(**[PanelListener](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\events\\PanelListener.html" \o "interface in badmagic.events)** l)  Метод удаления слушателя панели. |
| void | [**startListenToPeriphery**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#startListenToPeriphery--)()  Метод подключения слушателей мыши и клавиатуры. |
| void | [**startNewCareer**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#startNewCareer--)()  Метод начала новой игры в режиме карьеры. |
| void | [**stopListenToPeriphery**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\view\GamePanel.html#stopListenToPeriphery--)()  Метод отключения слушателей мыши и клавиатуры. |

**Class InteractiveObject**

public abstract class **InteractiveObject**

extends GameObject

Абстрактный класс, реализующий интерактвный объект. Наследник GameObject. Наследники класса представляют собой интерактивный объект, который игрок может активировать. Наследник может быть закрыт на замок. В таком случае для его активации, игроку понадобится найти ключ и открыть наследника.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| (package private) int | [**openId**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\InteractiveObject.html#openId) |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**InteractiveObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\InteractiveObject.html#InteractiveObject-badmagic.model.GameField-)(**[GameField](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\GameField.html" \o "class in badmagic.model)** field) |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| abstract boolean | [**activate**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\InteractiveObject.html#activate--)()  Абстрактный метод, активирующий объект |
| int | [**needKey**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\InteractiveObject.html#needKey--)()  Метод проверки требования к ключу. |
| abstract boolean | [**unlock**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\InteractiveObject.html#unlock-badmagic.model.gameobjects.CollectableObject-)(**[CollectableObject](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\gameobjects\\CollectableObject.html" \o "class in badmagic.model.gameobjects)** key)  Абстрактный метод, открывающий объект |

**Class Level**

public class **Level**

extends java.lang.Object

Класс представляет игровой уровень. Считывает уровень из json файла и хранит информацию о нем.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private [**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) | [**\_field**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#Z:Z_field)  Игровое поле уровня |
| private java.lang.String | [**\_history**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#Z:Z_history)  История уровня |
| private boolean | [**\_isCompleted**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#Z:Z_isCompleted)  Флаг - пройден ли уровень |
| private java.lang.String | [**\_levelPath**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#Z:Z_levelPath)  Путь к файлу с уровнем |
| private int | [**\_moves**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#Z:Z_moves)  Количество ходов уровня |
| private java.lang.String | [**\_name**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#Z:Z_name)  Имя уровня |
| private static int | [**MAX\_HEIGHT**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#MAX_HEIGHT)  Максимальная высота поля - клеток |
| private static int | [**MAX\_WIDTH**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#MAX_WIDTH)  Максимальная ширина поля - клеток |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Level**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#Level-java.lang.String-)(java.lang.String levelPath)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| [**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) | [**getField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#getField--)()  Метод получения поля уровня. |
| java.lang.String | [**getHistory**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#getHistory--)()  Метод получения истории уровня. |
| int | [**getMoves**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#getMoves--)()  Метод получения количества ходов уровня. |
| java.lang.String | [**getName**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#getName--)()  Метод получения имени уровня. |
| boolean | [**isCompleted**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#isCompleted--)()  Метод получения флага - пройден ли уровень. |
| void | [**loadLevel**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#loadLevel-java.lang.String-)(java.lang.String levelPath)  Загрузить уровень из файла. |
| private void | [**parseFieldSize**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#parseFieldSize-org.json.simple.JSONObject-)(org.json.simple.JSONObject json)  Метод разбора и установки размеров полей. |
| private java.util.ArrayList<java.lang.Integer> | [**parseIntegerArray**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#parseIntegerArray-org.json.simple.JSONArray-)(org.json.simple.JSONArray array)  Метод парсит JSON массив, содержащий целые числа и возвращает их в виде списка |
| private void | [**parseObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#parseObject-org.json.simple.JSONObject-java.lang.String-)(org.json.simple.JSONObject obj, java.lang.String objClassName)  \* Метод разбора и установки одного типа игрового объектоа. |
| private void | [**parseObjects**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#parseObjects-org.json.simple.JSONObject-)(org.json.simple.JSONObject json)  Метод разбора и установки игровых объектов. |
| private java.util.ArrayList<java.awt.Point> | [**parsePointArray**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#parsePointArray-org.json.simple.JSONArray-)(org.json.simple.JSONArray array)  Метод парсит JSON массив, содержащий объекты типа Point и проверяет их на ошибки |
| (package private) static java.util.ArrayList<java.lang.Integer> | [**removeIntegerDuplicates**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#removeIntegerDuplicates-java.util.ArrayList-)(java.util.ArrayList<java.lang.Integer> list)  Метод удаляет дубликаты из списка |
| void | [**setIsCompleted**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\Level.html#setIsCompleted-boolean-)(boolean isCompleted)  Метод установки флага - пройден ли уровень. |

**Class MovableObject**

public abstract class **MovableObject**

extends [GameObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)

Абстрактный класс - передвигаемый игровой объект. Наследник класса GameObject. Содержит метод перемещения объекта.

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**MovableObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\MovableObject.html#MovableObject-badmagic.model.GameField-)([**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**move**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\MovableObject.html#move-badmagic.navigation.Direction-)(**[Direction](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\navigation\\Direction.html" \o "class in badmagic.navigation)** moveDirection)  Метод перемещения объекта в направлении. |

**Class Player**

public class **Player**

extends [GameObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)

Класс представляет игрока. Наследник класса GameObject. Реализует абстрактные методы. Имеет образ - изображение, направление взгляда и количество ходов.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private [**Direction**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\navigation\Direction.html) | [**\_gazeDirection**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#Z:Z_gazeDirection)  Направление взгляда игрока |
| private java.util.ArrayList<[**GameObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)> | [**\_inventory**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#Z:Z_inventory)  Инвентарь |
| private int | [**\_moves**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#Z:Z_moves)  Количество ходов игрока |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#PIC)  Путь к файлу с изображением |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Player**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#Player-badmagic.model.GameField-)([**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| void | [**activateObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#activateObject--)()  Метод, активирующий объект в клетке по направлению |
| void | [**clearInventory**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#clearInventory--)()  Метод очистки инвентаря |
| private void | [**getItemFromField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#getItemFromField-java.awt.Point-)(java.awt.Point pos)  Метод, убирающий предмет с поля в инвентарь |
| int | [**getMoves**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#getMoves--)()  Метод получения количества ходов игрока. |
| boolean | [**haveElixirInInventory**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#haveElixirInInventory--)()  Метод проверки наличия эликсира в инвентаре |
| protected void | [**loadPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#loadPic--)()  Метод загрузки изображения объекта. |
| void | [**move**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#move-badmagic.navigation.Direction-)(**[Direction](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\navigation\\Direction.html" \o "class in badmagic.navigation)** moveDirection)  Метод перехода игрока на соседнюю клетку в направлении. |
| private void | [**move**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#move-badmagic.navigation.Direction-boolean-)([**Direction**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\navigation\Direction.html) moveDirection, boolean needToChangeDirection)  Метод перехода игрока на соседнюю клетку в направлении. |
| void | [**moveObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#moveObject-badmagic.navigation.Direction-)(**[Direction](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\navigation\\Direction.html" \o "class in badmagic.navigation)** moveDirection)  Метод перемещения объекта на соседнюю клетку. |
| void | [**paint**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#paint-java.awt.Graphics-java.awt.Point-)(java.awt.Graphics g, java.awt.Point pos)  Метод отрисовки объекта. |
| private java.awt.geom.AffineTransform | [**rotate**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#rotate-badmagic.navigation.Direction-java.awt.Point-)([**Direction**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\navigation\Direction.html) direction, java.awt.Point pos)  Метод поворота изображения игрока. |
| void | [**setMoves**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Player.html#setMoves-int-)(int moves)  Метод установки количества ходов игрока. |

**Class Spell**

public class **Spell**

extends [CollectableObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\CollectableObject.html)

Класс, реализующий заклинание - ключ. Является наследником CollectableObject. Имеет собственный численный идентификатор. Имеет собственное графическое представление.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private int | [**id**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Spell.html#id)  Идентификатор открываемого объекта |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Spell.html#PIC)  Путь к файлу с изображением |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Spell**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Spell.html#Spell-badmagic.model.GameField-)(**[GameField](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\GameField.html" \o "class in badmagic.model)** field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| int | [**getId**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Spell.html#getId--)()  Метод, возвращающий текущий идентификатор заклинания. |
| protected void | [**loadPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Spell.html#loadPic--)()  Метод загрузки изображения объекта. |
| void | [**paint**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Spell.html#paint-java.awt.Graphics-java.awt.Point-)(java.awt.Graphics g, java.awt.Point pos)  Метод отрисовки объекта. |
| void | [**setId**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Spell.html#setId-int-)(int \_id)  Метод, устанавливающий идентификатор для заклинания. |

**Class Teleport**

public class **Teleport**

extends [InteractiveObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\InteractiveObject.html)

Класс, представляющий собой телепортатор. Является наследником InteractiveObject Имеет собственное графическое представление Хранит позицию для телепортации

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private java.awt.Point | [**\_tPosition**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#Z:Z_tPosition)  Позиция телепортации |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#PIC)  Путь к файлу с изображением |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Teleport**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#Teleport-badmagic.model.GameField-)([**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| boolean | [**activate**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#activate--)()  Метод активации объекта. |
| protected void | [**loadPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#loadPic--)()  Метод загрузки изображения объекта. |
| void | [**paint**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#paint-java.awt.Graphics-java.awt.Point-)(java.awt.Graphics g, java.awt.Point pos)  Метод отрисовки объекта. |
| void | [**setLock**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#setLock-int-)(int key)  Метод, устанавливающий ключ для объекта |
| void | [**setTPosition**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#setTPosition-java.awt.Point-)(java.awt.Point tPosition)  Метод, устанавливающий позицию для телепортации. |
| boolean | [**unlock**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Teleport.html#unlock-badmagic.model.gameobjects.CollectableObject-)(**[CollectableObject](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\gameobjects\\CollectableObject.html" \o "class in badmagic.model.gameobjects)** key)  Метод для открытия объекта (снятия с него замка) |

**Class UnmovableObject**

public abstract class **UnmovableObject**

extends [GameObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\GameObject.html)

Абстрактный класс - непередвигаемый игровой объект. Наследник класса GameObject.

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**UnmovableObject**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\UnmovableObject.html#UnmovableObject-badmagic.model.GameField-)([**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) field)  Конструктор класса. |

**Class Wall**

public class **Wall**

extends [UnmovableObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\UnmovableObject.html)

Класс представляет игровой объект - стену. Наследник класса UnmovableObject. Реализует абстрактные методы. Имеет образ - изображение.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Wall.html#PIC)  Путь к файлу с изображением |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**Wall**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Wall.html#Wall-badmagic.model.GameField-)(**[GameField](file:///D:\\MEGA\\Institute\\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\\Semestry\\Game\\BadMagic\\dist\\javadoc\\badmagic\\model\\GameField.html" \o "class in badmagic.model)** field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| protected void | [**loadPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Wall.html#loadPic--)()  Метод загрузки изображения объекта. |
| void | [**paint**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\Wall.html#paint-java.awt.Graphics-java.awt.Point-)(java.awt.Graphics g, java.awt.Point pos)  Метод отрисовки объекта. |

**Class WoodenTable**

public class **WoodenTable**

extends [MovableObject](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\MovableObject.html)

Класс представляет игровой объект - деревянный стол. Наследник класса MovableObject. Реализует абстрактные методы. Имеет образ - изображение.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| private static java.lang.String | [**PIC**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\WoodenTable.html#PIC)  Путь к файлу с изображением |

|  |
| --- |
| **Constructors** |
| **Constructor and Description** |
| [**WoodenTable**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\WoodenTable.html#WoodenTable-badmagic.model.GameField-)([**GameField**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\GameField.html) field)  Конструктор класса. |

|  |  |
| --- | --- |
| **All Methods** | |
| **Modifier and Type** | **Method and Description** |
| protected void | [**loadPic**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\WoodenTable.html#loadPic--)()  Метод загрузки изображения объекта. |
| void | [**paint**](file:///D:\MEGA\Institute\%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D0%B8%D0%B5%D0%BD%D1%82%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%BD%D0%BE%D0%B5%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5\Semestry\Game\BadMagic\dist\javadoc\badmagic\model\gameobjects\WoodenTable.html#paint-java.awt.Graphics-java.awt.Point-)(java.awt.Graphics g, java.awt.Point pos)  Метод отрисовки объекта. |

# 5 Список использованной литературы и других источников

1. Бадд Т. Объектно-ориентированное программирование в действии / Перев. с англ. – Спб.: Питер, 1997. – 464 с.
2. Фаулер, Мартин; Скотт, Кендалл. UML. Основы. - Пер. с англ. - СПб.: Символ-Плюс, 2002. - 192 с.
3. Эккель, Б. Философия Java. Библиотека программиста. 4-е изд. – СПб.: Питер, 2009 – 640с.
4. Справочная служба Javadoc [Электронный ресурс] – Режим доступа: http://docs.oracle.com/javase/7/docs/.

# Приложение A. Исходный код программы

Класс GameModel

package badmagic.model;

import badmagic.BadMagic;

import badmagic.events.GameObjectListener;

import badmagic.events.ModelListener;

import badmagic.model.Level;

import badmagic.model.gameobjects.CollectableObject;

import badmagic.model.gameobjects.Elixir;

import badmagic.model.gameobjects.GameObject;

import badmagic.model.gameobjects.Player;

import java.awt.Point;

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.EventObject;

import java.util.logging.Logger;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.JSONValue;

import org.json.simple.parser.JSONParser;

/\*\*

\* Класс представляет модель игры.

\*

\* Отвечает за логику игры: загрузку уровней, режимы игры, переход между

\* уровнями, определение завершенности уровня. сохранение и загрузка информации

\* о прохождении

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public class GameModel {

/\*\*

\* Конструктор класса.

\*

\* Загружает уровни игры.

\*

\* @throws Exception при возникновении ошибок загрузки уровней.

\*/

public GameModel() throws Exception {

loadLevels();

}

/\*\*

\* Метод начала новой игры в режиме карьеры.

\*

\* Загружает первый уровень и дальше игра продолжается пока есть уровни.

\*

\* @throws Exception при возникновении ошибок загрузки уровней.

\*/

public void startNewCareer() throws Exception {

startLevel(0);

\_gameMode = GameMode.CAREER;

}

/\*\*

\* Метод продожения игры в режиме карьеры

\*

\* Загружает из номер последнего пройденного уровня, загружает этот уровень

\* и дальше игра продолжается пока есть уровни.

\*

\* @throws Exception при возникновении ошибок загрузки уровней.

\*/

public void continueCareer() throws Exception {

/\* Попытка загрузить данные о прохождении \*/

try {

loadGameProgress();

} catch (Exception ex) {

ex.printStackTrace();

}

startLevel(\_lastCompletedLevel);

\_gameMode = GameMode.CAREER;

}

/\*\*

\* Метод начала игры в режиме одного уровня.

\*

\* Загружает выбранный уровень и дальше игра продолжается пока есть уровни.

\*

\* @param levelNumber номер уровня

\* @throws Exception при возникновении ошибок загрузки уровней.

\*/

public void oneLevelMode(int levelNumber) throws Exception {

startLevel(levelNumber);

\_gameMode = GameMode.ONE\_LEVEL;

}

/\*\*

\* Метод перехода к следующему уровню игры.

\*/

public void nextLevel() {

/\* Текущий уровень - следующий \*/

setCurrentLevel(++\_currentLevel);

\_lastCompletedLevel = \_currentLevel;

/\* Инициализируем игрока \*/

initializePlayer();

/\* Статус уровня \*/

\_levelStatus = LevelStatus.PLAYING;

}

/\*\*

\* Метод повторного прохождения уровня.

\*/

public void tryAgain() {

/\* Загрузить уровень снова \*/

Level sameLevel;

try {

sameLevel = new Level(\_levelNames.get(\_currentLevel));

} catch (Exception ex) {

ex.printStackTrace();

return;

}

\_levels.set(\_currentLevel, sameLevel);

/\* Текущий уровень - этот же \*/

setCurrentLevel(\_currentLevel);

/\* Инициализируем игрока \*/

initializePlayer();

/\* Статус уровня \*/

\_levelStatus = LevelStatus.PLAYING;

}

/\*\*

\* Метод получения текущего игрового поля.

\*

\* @return GameField - текущее игровое поле.

\*/

public GameField getField() {

return \_field;

}

/\*\*

\* Метод получения текущего игрока.

\*

\* @return Player - текущий игрок

\*/

public Player getPlayer() {

return \_player;

}

/\*\*

\* Метод получения статуса текущего уровня.

\*

\* @return LevelStatus - статус уровня.

\*/

public LevelStatus getLevelStatus() {

return \_levelStatus;

}

/\*\*

\* Метод получения текущего игрового режима.

\*

\* @return GameMode - текущий игровой режим

\*/

public GameMode getGameMode() {

return \_gameMode;

}

/\*\*

\* Метод получения имени текущего уровня.

\*

\* @return String - имя текущего уровня.

\*/

public String getLevelName() {

return \_levels.get(\_currentLevel).getName();

}

/\*\*

\* Статический метод получения имен уровней игры.

\*

\* @return ArrayList - список имен уровней игры.

\*/

static public ArrayList<String> getLevelNames() {

ArrayList<String> levelNames = new ArrayList<>();

for (Level lvl : \_levels) {

levelNames.add(lvl.getName());

}

return levelNames;

}

/\*\*

\* Метод получения количества ходов текущего уровня.

\*

\* @return int - количество ходов.

\*/

public int getMoves() {

return \_player.getMoves();

}

/\*\*

\* Метод,получения информации о том, является ли текущий уровень последним.

\*

\* @return boolean - флаг - последний ли уровень

\*/

public boolean isLastLevel() {

return (\_levels.size() - 1) == \_currentLevel;

}

/\*\*

\* Метод загрузки уровней игры.

\*

\* @throws Exception при возникновении ошибок загрузки.

\*/

private void loadLevels() throws Exception {

Level level = null;

\_levels.clear();

/\* Попытка загрузить имена файлов с уровнями \*/

\_levelNames = loadLevelNames();

/\* Загружаем данные уровней \*/

for (Object i : \_levelNames) {

level = new Level(i.toString());

\_levels.add(level);

}

}

/\*\*

\* Метод начала уровня номер levelNumber.

\*

\* @param levelNumber номер уровня.

\* @throws Exception при возникновении ошибок загрузки уровня.

\*/

private void startLevel(int levelNumber) throws Exception {

reset();

loadLevels();

/\* Текущий уровень \*/

setCurrentLevel(levelNumber);

/\* Инициализируем игрока \*/

initializePlayer();

/\* Статус уровня \*/

\_levelStatus = LevelStatus.PLAYING;

}

/\*\*

\* Метод установки уровня номер leveNumber как текущего.

\*

\* @param levelNumber номер уровня.

\*/

private void setCurrentLevel(int levelNumber) {

if (levelNumber < 0 || levelNumber > (\_levels.size() - 1)) {

return;

}

\_currentLevel = levelNumber;

\_field = \_levels.get(\_currentLevel).getField();

}

/\*\*

\* Метод инициализации текущего игрока.

\*/

private void initializePlayer() {

/\* Устанавливаем игрока \*/

try {

\_player = (Player) \_field.getObjects(

Class.forName("badmagic.model.gameobjects.Player")).get(0);

\_player.setMoves(\_levels.get(\_currentLevel).getMoves());

\_player.clearInventory();

\_player.addObjectListener(\_objectsObserver);

} catch (ClassNotFoundException ex) {

ex.printStackTrace();

}

}

/\*\*

\* Метод проверки, завершен ли уровень.

\*

\* Проверяет, достиг ли игрок игрового объекта Elixir. Если да - посылает

\* сигнал об успешном прохождении уровня. Если нет - проверяет оставшееся

\* количество ходов и если оно 0 - посылает сигнал о неуспешном прохождении

\* уровня.

\*/

private void checkIfLevelIsFinished() {

if (\_player.haveElixirInInventory()) {

fireLevelCompleted();

\_field.clearSpellIdsList();

\_levelStatus = LevelStatus.COMPLETED;

}

if (\_player.getMoves() == 0) {

fireLevelFailed();

\_levelStatus = LevelStatus.FAILED;

}

}

/\*\*

\* Метод сохранения информации о последнем пройденном уровне в файл.

\*/

public static void saveGameProgress() {

JSONObject obj = new JSONObject();

obj.put("LastCompletedLevel", \_lastCompletedLevel);

try (

FileWriter file = new FileWriter(PATH\_TO\_GAME\_PROGRESS\_FILE)) {

file.write(obj.toJSONString());

} catch (IOException ex) {

ex.printStackTrace();

}

}

/\*\*

\* Метод загрузки информации о последнем пройденном уровне из файла.

\*

\* @throws Exception при возникновении ошибок загрузки.

\*/

public void loadGameProgress() throws Exception {

JSONParser parser = new JSONParser();

Object object = parser.parse(new FileReader(PATH\_TO\_GAME\_PROGRESS\_FILE));

JSONObject obj = (JSONObject) object;

if (obj.containsKey("LastCompletedLevel")) {

String tmp = obj.get("LastCompletedLevel").toString();

int value = Integer.parseInt(tmp);

if (value > (\_levels.size() - 1)) {

throw new Exception("Не корректно задана информация о "

+ "последнем пройденном уровне.");

}

\_lastCompletedLevel = Integer.parseInt(tmp);

} else {

throw new Exception("Не задана информация о "

+ "последнем пройденном уровне.");

}

}

/\*\*

\* Метод сброса модели.Очищает поля класса.

\*/

private void reset() {

if (\_field != null) {

\_field.clear();

}

\_player = null;

\_currentLevel = 0;

\_levels.clear();

\_levelNames.clear();

}

/\*\*

\* Статический метод загрузки информации об уровнях игры.

\*

\* Загружает пути к файлам уровней из специального файла с информацией об

\* уровнях игры.

\*

\* @return ArrayList - список путей к файлам уровней.

\* @throws Exception при возникновении ошибок загрузки.

\*/

private ArrayList<String> loadLevelNames() throws Exception {

ArrayList<String> levels = new ArrayList();

JSONParser parser = new JSONParser();

Object object = parser.parse(new FileReader(PATH\_TO\_LEVELS\_INFO\_FILE));

JSONArray array = (JSONArray) object;

if (array.isEmpty()) {

throw new Exception("Не заданы файлы уровней.");

}

for (Object i : array) {

levels.add(i.toString());

}

return levels;

}

/\*\*

\* Внутренний класс - слушатель игровых объектов.

\*

\* Вызывает метод проверки, завершен ли уровень при получении сигнала о том,

\* что игровой объект перемещен.

\*/

private class ObjectsObserver implements GameObjectListener {

/\*\*

\* Метод, обрабатывающий получение сигнала о перемещении объекта.

\*

\* Проверяет, завершен ли уровень.

\*

\* @param e событие.

\*/

@Override

public void objectMoved(EventObject e) {

checkIfLevelIsFinished();

}

}

////////////////////////// События модели /////////////////////////////////

/\*\*

\* Метод добавления слушателя модели.

\*

\* @param l слушатель.

\*/

public void addModelListener(ModelListener l) {

\_modelListenerList.add(l);

}

/\*\*

\* Метод удаления слушаетля модели.

\*

\* @param l слушатель.

\*/

public void removeModelListener(ModelListener l) {

\_modelListenerList.remove(l);

}

/\*\*

\* Метод испускания сигнала об успешном прохождении уровня.

\*/

protected void fireLevelCompleted() {

EventObject event = new EventObject(this);

for (Object listener : \_modelListenerList) {

((ModelListener) listener).levelCompleted(event);

}

}

/\*\*

\* Метод испускания сигнала о неуспешном прохождении уровня.

\*/

protected void fireLevelFailed() {

EventObject event = new EventObject(this);

for (Object listener : \_modelListenerList) {

((ModelListener) listener).levelFailed(event);

}

}

/\*\*

\* Список слушателей модели

\*/

private ArrayList \_modelListenerList = new ArrayList();

//////////////////////////// Данные ///////////////////////////////////////

/\*\*

\* Игровое поле

\*/

private GameField \_field;

/\*\*

\* Номер текущего уровня

\*/

private int \_currentLevel;

/\*\*

\* Игрок

\*/

private Player \_player;

/\*\*

\* Текущий игровой режим

\*/

private GameMode \_gameMode;

/\*\*

\* Статус текущего уровня

\*/

private LevelStatus \_levelStatus;

/\*\*

\* Номер последнего пройденного уровня

\*/

private static int \_lastCompletedLevel = 0;

/\*\*

\* Список уровней игры

\*/

private static ArrayList<Level> \_levels = new ArrayList();

/\*\*

\* Список имен уровней игры

\*/

private ArrayList<String> \_levelNames = new ArrayList();

/\*\*

\* Слушатель игровых объектов

\*/

private ObjectsObserver \_objectsObserver = new ObjectsObserver();

/\*\*

\* Путь к файлу с информацией об уровнях

\*/

private static final String PATH\_TO\_LEVELS\_INFO\_FILE

= "src/badmagic/resources/levels/levelsinfo.json";

/\*\*

\* Путь к файлу с информацией о прогрессе игры

\*/

private static final String PATH\_TO\_GAME\_PROGRESS\_FILE

= "src/badmagic/resources/gameprogress.json";

/\*\*

\* Статус уровня

\*/

public enum LevelStatus {

COMPLETED,

FAILED,

PLAYING

}

/\*\*

\* Игровой режим

\*/

public enum GameMode {

CAREER,

ONE\_LEVEL

}

}

Класс GameField

package badmagic.model;

import badmagic.model.gameobjects.CollectableObject;

import badmagic.model.gameobjects.GameObject;

import badmagic.navigation.Direction;

import java.awt.Point;

import java.util.ArrayList;

import java.util.LinkedHashMap;

import java.util.Map;

/\*\*

\* Класс представляет игровую поле.

\*

\* Построен на основе полиморфного контейнера игровых объектов

\* с добавлением логики поля.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public class GameField {

/\*\*

\* Конструктор игрового поля с параметрами.

\*

\* Устанавливает размер поля.

\*

\* @param width ширина поля.

\* @param height высота поля.

\*/

public GameField(int width,int height) {

setSize(width,height);

}

/////////////////// Работа с клетками поля ////////////////////////////////

/\*\*

\* Метод получения следующей позиции в указанном направлнеии.

\*

\* @param currentPos текущая позиция.

\* @param direction направление.

\* @return Point - следующая позиция.

\*/

public Point getNextPos(Point currentPos, Direction direction) {

/\* Проверка на невалидные аргументы \*/

if( currentPos == null || direction == null ) {

return null;

}

/\* Находим следующую позицию \*/

Point nextPos = null;

int x = currentPos.x;

int y = currentPos.y;

if( direction.equals(Direction.north())

&& (y - 1) >= 1 ) {

nextPos = new Point(x,y - 1);

} else if ( direction.equals(Direction.south())

&& (y+1) <= \_height ) {

nextPos = new Point(x,y + 1);

} else if ( direction.equals(Direction.west())

&& (x-1) >= 1 ) {

nextPos = new Point(x - 1,y);

} else if( direction.equals(Direction.east())

&& (x+1) <= \_width ) {

nextPos = new Point(x + 1,y);

}

return nextPos;

}

/\*\*

\* Метод проверки позиции на незанятость.

\*

\* @param pos позиция, которая проверяется.

\* @return boolean - флаг - занята ли позиция.

\*/

public boolean isPosEmpty(Point pos) {

/\* Проверка переданных координат на принадлежность полю \*/

if( pos == null ||

((pos.x < 1 || pos.x > \_width ) ||

(pos.y < 1 || pos.y > \_height)) ) {

return false;

}

/\* Поиск объектов с такой же позицией \*/

boolean isEmpty = true;

ArrayList<GameObject> fieldObjects = getObjects();

for(GameObject object : fieldObjects) {

if( object.getPosition().equals(pos) &&

!(object instanceof CollectableObject)) {

isEmpty = false;

}

}

return isEmpty;

}

public boolean isPosHasCollectable(Point pos) {

/\* Проверка переданных координат на принадлежность полю \*/

if( pos == null ||

((pos.x < 1 || pos.x > \_width ) ||

(pos.y < 1 || pos.y > \_height)) ) {

return false;

}

/\* Поиск объектов с такой же позицией \*/

boolean isEmpty = false;

ArrayList<GameObject> fieldObjects = getObjects(pos);

for(GameObject object : fieldObjects) {

if(object instanceof CollectableObject) {

isEmpty = true;

}

}

return isEmpty;

}

/\*\*

\* Метод проверки следующей позиции в направлении на незанятость.

\*

\* @param currentPos текущая позиция.

\* @param direction направление.

\* @return boolean - флаг - занята ли следующая позиция.

\*/

public boolean isNextPosEmpty(Point currentPos, Direction direction) {

Point nextPos = getNextPos(currentPos,direction);

return isPosEmpty(nextPos);

}

public boolean isNextPosHasCollectable(Point currentPos, Direction direction){

Point nextPos = getNextPos(currentPos,direction);

return isPosHasCollectable(nextPos);

}

/////////////////// Работа с размерами поля ///////////////////////////////

/\*\*

\* Метод установления размера поля

\*

\* @param width ширина

\* @param height высота

\*/

public void setSize(int width,int height) {

setWidth(width);

setHeight(height);

}

/\*\*

\* Метод получения ширины поля

\*

\* @return int - ширина поля.

\*/

public int getWidth() {

return \_width;

}

/\*\*

\* Метод установки ширины поля.

\*

\* @param width ширина.

\*/

public void setWidth(int width) {

\_width = width;

}

/\*\*

\* Метод получения высоты поля

\*

\* @return int - высота поля.

\*/

public int getHeight() {

return \_height;

}

/\*\*

\* Метод установки высоты поля.

\*

\* @param height высота.

\*/

public void setHeight(int height) {

\_height = height;

}

/////////////////// Работа с объектами поля ///////////////////////////////

/\*\*

\* Метод добавления объекта на поле.

\*

\* @param pos позиция объекта.

\* @param obj объект добавления.

\* @return boolean - флаг - успешност добавления.

\*/

public boolean addObject(Point pos, GameObject obj) {

Class objClass = obj.getClass();

if ( obj.setPosition(pos) ) {

if( \_gameObjects.containsKey(objClass) ) {

\_gameObjects.get(objClass).add(obj);

} else {

ArrayList<GameObject> objList = new ArrayList<>();

objList.add(obj);

\_gameObjects.put(objClass, objList);

}

return true;

}

return false;

}

/\*\*

\* Метод удаления объекта с поля.

\*

\* @param obj объект поля.

\* @return boolean - флаг - успешность удаления.

\*/

public boolean removeObject(GameObject obj) {

boolean isSuccessful = false;

Class objClass = obj.getClass();

if( \_gameObjects.containsKey(objClass) ) {

isSuccessful = \_gameObjects.get(objClass).remove(obj);

if( isSuccessful ) {

obj.unsetPosition();

}

}

return isSuccessful;

}

/\*\*

\* Метод получения объектов поля.

\*

\* @return ArrayList - все объекты поля.

\*/

public ArrayList<GameObject> getObjects() {

ArrayList<GameObject> objList = new ArrayList<>();

for( Map.Entry<Class, ArrayList<GameObject>> entry :

\_gameObjects.entrySet()) {

objList.addAll(entry.getValue());

}

return objList;

}

/\*\*

\* Метод получения объетов поля с заданной позицией.

\*

\* @param pos позиция объектов.

\* @return ArrayList - все объекты поля с заданной позицией.

\*/

public ArrayList<GameObject> getObjects(Point pos) {

ArrayList<GameObject> objList = new ArrayList<>();

for( Map.Entry<Class, ArrayList<GameObject> > entry :

\_gameObjects.entrySet()) {

for( GameObject obj : entry.getValue() ) {

if( obj.getPosition().equals(pos) ) {

objList.add(obj);

}

}

}

return objList;

}

/\*\*

\* Метод получения объектов поля заданного типа.

\*

\* @param objType тип объекта получения.

\* @return ArrayList - все объекты поля заданного типа.

\*/

public ArrayList<GameObject> getObjects(Class objType) {

ArrayList<GameObject> objList = new ArrayList<>();

if( \_gameObjects.containsKey(objType) ) {

objList.addAll( \_gameObjects.get(objType) );

}

return objList;

}

/\*\*

\* Метод получения объектов заданного типа и позиции.

\*

\* @param objType тип объекта получения.

\* @param pos позиция объекта получения.

\* @return ArrayList - все объекты поля заданного типа и позиции.

\*/

public ArrayList<GameObject> getObjects(Class objType, Point pos) {

ArrayList<GameObject> objList = new ArrayList<>();

if( \_gameObjects.containsKey(objType) ) {

for( GameObject obj : \_gameObjects.get(objType) ) {

if( obj.getPosition().equals(pos) ) {

objList.add(obj);

}

}

}

return objList;

}

/\*\*

\* Метод проверки позиции на уникальность - незанятость.

\*

\* @param pos позиция, которая проверяется.

\* @return boolean - флаг - не занята ли позиция.

\*/

public boolean isPositionUnique(Point pos) {

ArrayList<GameObject> objects = getObjects();

for(GameObject obj : objects) {

if(pos.equals(obj.getPosition())) {

return false;

}

}

return true;

}

public boolean isSpellIdUnique(int id){

return !(\_spellIds.contains(id));

}

public void setSpellIdsList(ArrayList<Integer> list){

\_spellIds = new ArrayList<Integer> (list);

}

public ArrayList<Integer> spellIdsList(){

return \_spellIds;

}

public void clearSpellIdsList(){

\_spellIds.clear();

}

/\*\*

\* Метод очистки игрового поля.

\*

\* Очищает игровые объекты поля.

\*/

public void clear() {

\_gameObjects.clear();

}

//////////////////////////////// Данные ///////////////////////////////////

/\*\* Ширина поля \*/

private int \_width;

/\*\* Высота поля \*/

private int \_height;

private ArrayList<Integer> \_spellIds = new ArrayList<>();

/\*\* Игровые объекты поля \*/

private Map<Class, ArrayList<GameObject>> \_gameObjects = new LinkedHashMap<>();

}

Класс Level

package badmagic.model;

import badmagic.BadMagic;

import badmagic.model.gameobjects.Bookshelf;

import badmagic.model.gameobjects.GameObject;

import badmagic.model.gameobjects.WoodenTable;

import java.awt.Point;

import java.io.File;

import java.io.FileReader;

import java.io.IOException;

import java.io.FileInputStream;

import java.lang.reflect.Constructor;

import java.net.URL;

import java.util.HashMap;

import java.util.Map;

import org.json.simple.JSONArray;

import org.json.simple.JSONObject;

import org.json.simple.parser.JSONParser;

import badmagic.model.gameobjects.GameObjectsFactory;

import badmagic.model.gameobjects.Spell;

import badmagic.model.gameobjects.Teleport;

import java.util.ArrayList;

import java.util.HashSet;

import java.util.Set;

import java.util.logging.Logger;

/\*\*

\* Класс представляет игровой уровень.

\*

\* Считывает уровень из json файла и хранит информацию о нем.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public class Level {

/\*\*

\* Конструктор класса.

\*

\* Загружает уровнень игры.

\*

\* @param levelPath путь к файлу уровня.

\* @throws Exception при возникновении ошибок загрузки или парсинга файла.

\*/

Level(String levelPath) throws Exception {

loadLevel(levelPath);

}

/\*\*

\* Загрузить уровень из файла.

\*

\* Загружает json файл уровня, считывает его данные и инициализирует поля

\* класса.

\*

\* @param levelPath путь к файлу уровня.

\* @throws Exception при возникновении ошибок загрузки или парсинга.

\*/

public void loadLevel(String levelPath) throws Exception {

\_levelPath = levelPath;

Point size;

JSONParser parser = new JSONParser();

Object object = parser.parse(new FileReader(levelPath));

JSONObject json = (JSONObject) object;

/\* Название уровня \*/

if (json.containsKey("LevelName")) {

\_name = (String) json.get("LevelName");

}

/\* История уровня \*/

if (json.containsKey("LevelHistory")) {

\_history = (String) json.get("LevelHistory");

}

/\* Количество ходов игрока \*/

if (json.containsKey("PlayerMoves")) {

\_moves = Integer.parseInt(json.get("PlayerMoves").toString());

}

/\* Размер поля \*/

parseFieldSize(json);

/\* Объекты поля \*/

parseObjects(json);

}

/\*\*

\* Метод получения количества ходов уровня.

\*

\* @return int - количество ходов уровня.

\*/

public int getMoves() {

return \_moves;

}

/\*\*

\* Метод получения поля уровня.

\*

\* @return GameField - поле уровня.

\*/

public GameField getField() {

return \_field;

}

/\*\*

\* Метод получения имени уровня.

\*

\* @return String - имя уровня.

\*/

public String getName() {

return \_name;

}

/\*\*

\* Метод получения флага - пройден ли уровень.

\*

\* @return boolean - флаг - пройден ли уровень.

\*/

public boolean isCompleted() {

return \_isCompleted;

}

/\*\*

\* Метод установки флага - пройден ли уровень.

\*

\* @param isCompleted пройден ли уровень.

\*/

public void setIsCompleted(boolean isCompleted) {

\_isCompleted = isCompleted;

}

/\*\*

\* Метод получения истории уровня.

\*

\* @return String - история уровня.

\*/

public String getHistory() {

return \_history;

}

/\*\*

\* Метод разбора и установки размеров полей.

\*

\* Находит в json объекте данные о размерах поля и устанавливает их как

\* размеры поля \_field.

\*

\* @param json объект файла уровня.

\* @throws Exception при возникновении ошибок разбора.

\*/

private void parseFieldSize(JSONObject json) throws Exception {

if (json.containsKey("FieldSize")) {

JSONArray tmp = (JSONArray) json.get("FieldSize");

int width = Integer.parseInt(tmp.get(0).toString());

int height = Integer.parseInt(tmp.get(1).toString());

if (width > MAX\_WIDTH || height > MAX\_HEIGHT) {

Exception ex = new Exception("Размер поля ("

+ tmp.get(0).toString() + ";"

+ tmp.get(1).toString()

+ ") превышет "

+ "максимально допустимый ("

+ MAX\_WIDTH + ";" + MAX\_HEIGHT + ")");

throw ex;

} else if (width < 1 || height < 1) {

Exception ex = new Exception("Размер поля ("

+ tmp.get(0).toString() + ";"

+ tmp.get(1).toString()

+ ") не допустим");

throw ex;

}

\_field = new GameField(width, height);

} else {

Exception ex = new Exception("Не задан размер поля в файле уровня");

throw ex;

}

}

/\*\*

\* Метод разбора и установки игровых объектов.

\*

\* Находит в json объекте данные об игровых объектах и устанавливает их как

\* объекты поля \_field.

\*

\* @param json объект файла уровня.

\* @throws Exception при возникновении ошибок разбора.

\*/

private void parseObjects(JSONObject json) throws Exception {

if (json.containsKey("FieldObjects")) {

JSONArray objects = (JSONArray) json.get("FieldObjects");

/\* Парсинг объектов в массиве objects \*/

for (Object i : objects) {

JSONObject obj = (JSONObject) i;

Set keys = obj.keySet();

/\*

\* Ключ - конкретный тип объекта,

\* должен соответствовать имени класса

\*/

for (Object key : keys) {

JSONObject tmp = (JSONObject) obj.get(key.toString());

parseObject(tmp, key.toString());

}

}

} else {

Exception ex = new Exception("Не заданы объекты поля в файле уровня");

throw ex;

}

}

/\*\*

\* \* Метод разбора и установки одного типа игрового объектоа.

\*

\* Находит в json объекте данные об игровом объекте objClassName и объект к

\* объектам поля \_field.

\*

\* @param obj json объект с информацией об игровом объекте.

\* @param objClassName имя класса игрового объекта.

\* @throws Exception при возникновении ошибок разбора.

\*/

private void parseObject(JSONObject obj, String objClassName) throws Exception {

/\*Список позиций объекта типа objClassName\*/

ArrayList<Point> positions = new ArrayList<>();

/\*Список идентификаторов заклинаний\*/

ArrayList<Integer> spell\_ids = new ArrayList<>();

/\*Список заклинаний-содержимого книжных полок\*/

ArrayList<Integer> bs\_contains\_ids = new ArrayList<>();

/\*Список заклинаний-ключей книжных полок\*/

ArrayList<Integer> bs\_keys\_ids = new ArrayList<>();

/\*Список заклинаний-ключей телепортов\*/

ArrayList<Integer> tp\_keys\_ids = new ArrayList<>();

/\*Список позиций для телепорта\*/

ArrayList<Point> tp\_positions = new ArrayList<>();

if (obj.containsKey("Positions")) {

positions = parsePointArray((JSONArray) obj.get("Positions"));

}

/\*Для заклинаний\*/

/\*Парсинг доступных id\*/

if (obj.containsKey("AvailableIds")) {

spell\_ids = parseIntegerArray((JSONArray) obj.get("AvailableIds"));

spell\_ids = removeIntegerDuplicates(spell\_ids);

getField().setSpellIdsList(spell\_ids);

}

/\*Парсинг заклинаний на поле и вне объектов\*/

if (obj.containsKey("OnField")) {

/\*Несовпадение количества позиций и idшников на поле\*/

if (((JSONArray) obj.get("OnField")).size() != positions.size()) {

String error = "Число заклинаний вне объектов не совпадает с числом позиций";

throw new Exception(error);

}

spell\_ids = parseIntegerArray((JSONArray) obj.get("OnField"));

/\*Проверка на неизвестные идентификаторы\*/

for (Integer value : spell\_ids){

if (\_field.isSpellIdUnique(value) || value == -1) {

String error = "Неизвестный идентификатор заклинания вне объекта: " + value;

throw new Exception(error);

}

}

}

/\*Книжная полка с заклинаниями\*/

/\*Содержимое полок\*/

if (obj.containsKey("Contains")) {

if (((JSONArray) obj.get("Contains")).size() != positions.size()) {

String error = "Число заклинаний-содержимых не совпадает с числом позиций";

throw new Exception(error);

}

bs\_contains\_ids = parseIntegerArray((JSONArray) obj.get("Contains"));

/\*Проверка на неизвестные идентификаторы\*/

for (Integer value : bs\_contains\_ids){

if (\_field.isSpellIdUnique(value) && value != -1) {

String error = "Неизвестный идентификатор заклинания-содержимого: " + value;

throw new Exception(error);

}

}

}

/\*Ключи\*/

if (obj.containsKey("Keys")) {

if (((JSONArray) obj.get("Keys")).size() != positions.size()) {

String error = "Число заклинаний-ключей не совпадает с числом позиций";

throw new Exception(error);

}

bs\_keys\_ids = parseIntegerArray((JSONArray) obj.get("Keys"));

/\*Проверка на неизвестные идентификаторы\*/

for (Integer value : bs\_keys\_ids){

if (\_field.isSpellIdUnique(value) && value != -1) {

String error = "Неизвестный идентификатор заклинания-ключа: " + value;

throw new Exception(error);

}

}

}

/\*Телепорты\*/

/\*Позиции для телепортации\*/

if (obj.containsKey("TeleportPositions")) {

tp\_positions = parsePointArray((JSONArray) obj.get("TeleportPositions"));

}

/\*Ключи\*/

if (obj.containsKey("TeleportKeys")) {

if (((JSONArray) obj.get("TeleportKeys")).size() != positions.size()) {

String error = "Число заклинаний-ключей не совпадает с числом позиций";

throw new Exception(error);

}

tp\_keys\_ids = parseIntegerArray((JSONArray) obj.get("TeleportKeys"));

/\*Проверка на неизвестные идентификаторы\*/

for (Integer value : bs\_keys\_ids){

if (\_field.isSpellIdUnique(value) && value != -1) {

String error = "Неизвестный идентификатор заклинания-ключа: " + value;

throw new Exception(error);

}

}

}

/\*Создание объектов\*/

for (int i = 0; i < positions.size(); ++i) {

GameObject o = new GameObjectsFactory().createGameObject(objClassName, getField());

if (o instanceof Spell) {

/\*Присваивание идентификтора\*/

((Spell) o).setId(spell\_ids.get(i));

}

if (o instanceof Bookshelf) {

/\*Присваивание ключа и содержания\*/

((Bookshelf) o).setLock(bs\_keys\_ids.get(i));

((Bookshelf) o).setSpell(bs\_contains\_ids.get(i));

}

if (o instanceof Teleport) {

/\*Присваивание ключа и позиции телепортирования\*/

((Teleport) o).setTPosition(tp\_positions.get(i));

((Teleport) o).setLock(tp\_keys\_ids.get(i));

}

getField().addObject(positions.get(i), o);

}

}

/\*\*

\* Метод парсит JSON массив, содержащий объекты типа Point и проверяет их на ошибки

\* @param array - массив Point

\* @return массив объектов Point в виде листа

\* @throws Exception ошибка, возникающая при некорректных данных

\*/

private ArrayList<Point> parsePointArray(JSONArray array) throws Exception {

ArrayList<Point> resultList = new ArrayList<>();

for (Object j : array) {

JSONArray positionData = (JSONArray) j;

if (positionData.size() != 2) {

String error = "Позиция " + (resultList.size() + 1) + "-го элемента некорректна."

+ "Ожидаемое количество координат: 2, Полученное количество: "

+ positionData.size();

throw new Exception(error);

}

Point position = new Point(Integer.parseInt(positionData.get(0).toString()),

Integer.parseInt(positionData.get(1).toString()));

if (!\_field.isPositionUnique(position)) {

String error = "Позиция " + (resultList.size() + 1)

+ "-го элемента (" + position.x

+ ";" + position.y

+ ") уже занята.";

throw new Exception(error);

} else if ((position.x < 1 || position.x > \_field.getWidth())

|| (position.y < 1 || position.y > \_field.getHeight())) {

String error = "Позиция " + (resultList.size() + 1)

+ "-го элемента (" + position.x

+ ";" + position.y

+ ") не может быть размещен, "

+ "т.к. его позиция за пределами поля.";

throw new Exception(error);

}

resultList.add(position);

}

return resultList;

}

/\*\*

\* Метод парсит JSON массив, содержащий целые числа и возвращает их в виде списка

\* @param array - JSON-массив чисел

\* @return массив чисел в виде листа

\* @throws Exception ошибка, возникающая при некорректных данных

\*/

private ArrayList<Integer> parseIntegerArray(JSONArray array) throws Exception {

ArrayList<Integer> resultList = new ArrayList<>();

for (Object j : array) {

int value = Integer.parseInt(j.toString());

resultList.add(value);

}

return resultList;

}

/\*\*

\* Метод удаляет дубликаты из списка

\* @param list - список из которого нужно удалить дубликаты

\* @return очищенный от дубликатов список

\*/

static ArrayList<Integer> removeIntegerDuplicates(ArrayList<Integer> list) {

ArrayList<Integer> result = new ArrayList<>();

HashSet<Integer> set = new HashSet<>();

for (Integer item : list) {

if (!set.contains(item)) {

result.add(item);

set.add(item);

}

}

return result;

}

////////////////////////////// Данные /////////////////////////////////////

/\*\* Максимальная ширина поля - клеток \*/

private static final int MAX\_WIDTH = 16;

/\*\*

\* Максимальная высота поля - клеток

\*/

private static final int MAX\_HEIGHT = 11;

/\*\*

\* Количество ходов уровня

\*/

private int \_moves = 100;

/\*\*

\* Игровое поле уровня

\*/

private GameField \_field;

/\*\*

\* Имя уровня

\*/

private String \_name = "default level name";

/\*\*

\* История уровня

\*/

private String \_history = "default level history";

/\*\*

\* Путь к файлу с уровнем

\*/

private String \_levelPath;

/\*\*

\* Флаг - пройден ли уровень

\*/

private boolean \_isCompleted = false;

}

Класс GameObject

package badmagic.model.gameobjects;

import badmagic.events.GameObjectListener;

import badmagic.model.GameField;

import java.awt.Graphics;

import java.awt.Point;

import java.awt.image.BufferedImage;

import java.util.ArrayList;

import java.util.EventObject;

/\*\*

\* Абстрактный класс - игровой объект.<br>

\*

\* Хранит позицию игрового объекта и ссылку на класс игрового поля.<br>

\* Может посылать сообщения.<br>

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public abstract class GameObject {

/\*\*

\* Конструктор класса.<br>

\*

\* Инициализирует поля класса.

\*

\* @param field ссылка на игровое поле

\*/

public GameObject(GameField field) {

\_field = field;

}

/\*\*

\* Метод получения позиции объекта.<br>

\*

\* @return Point - позиция объекта.

\*/

public Point getPosition() {

return \_position;

}

/\*\*

\* Метод обнуления позиции игрового объекта.

\*/

public void unsetPosition() {

\_position = null;

}

/\*\*

\* Метод установки позиции игрового объекта.<br>

\* Позиция будет установлена, если передаваемый

\* параметр не выходит за границы поля и не null.

\*

\* @param pos позиция объекта.

\* @return boolean - успешность установки.

\*/

public boolean setPosition(Point pos) {

if( pos != null && \_field.getObjects(pos).isEmpty() ) {

\_position = pos;

return true;

}

return false;

}

/\*\*

\* Абстрактный метод отрисовки объекта.

\* @param g среда отрисовки.

\* @param pos позиция отрисовки

\*/

public abstract void paint(Graphics g,Point pos);

/\*\*

\* Абстрактный метод загрузки изображения объекта.

\*/

protected abstract void loadPic();

///////////////////// События игрового объекта ////////////////////////////

/\*\*

\* Метод добавления слушателя игрового объекта.

\* @param l слушатель.

\*/

public void addObjectListener(GameObjectListener l) {

\_listenerList.add(l);

}

/\*\*

\* Метод удаления слушаетля игрового объекта.

\* @param l слушатель.

\*/

public void removeObjectListener(GameObjectListener l) {

\_listenerList.remove(l);

}

/\*\*

\* Метод испускания сигнала о перемещении объекта.

\*/

protected void fireObjectMoved() {

EventObject event = new EventObject(this);

for (Object listener : \_listenerList) {

((GameObjectListener) listener).objectMoved(event);

}

}

/\*\* Список слушателей игрового объекта \*/

private ArrayList \_listenerList = new ArrayList();

///////////////////////////// Данные //////////////////////////////////////

/\*\* Позиция игрового объекта \*/

protected Point \_position;

/\*\* Ссылка на игровое поле \*/

protected GameField \_field;

/\*\* Изображение объекта \*/

protected BufferedImage \_image;

}

Класс MovableObject

package badmagic.model.gameobjects;

import badmagic.model.GameField;

import badmagic.navigation.Direction;

/\*\*

\* Абстрактный класс - передвигаемый игровой объект.

\*

\* Наследник класса GameObject.

\* Содержит метод перемещения объекта.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public abstract class MovableObject extends GameObject {

/\*\*

\* Конструктор класса.

\*

\* Вызывает конструктор базового класса.

\*

\* @param field ссылка на игровое поле.

\*/

public MovableObject(GameField field) {

super(field);

}

/\*\*

\* Метод перемещения объекта в направлении.

\*

\* Осуществляет перемещение объекта на соседнюю клетку в

\* указанном направлении, если эта клетка свободна.

\*

\* @param moveDirection направление перемещения.

\*/

public void move(Direction moveDirection) {

if( \_field.isNextPosEmpty(\_position,moveDirection) ) {

\_position = \_field.getNextPos(\_position,moveDirection);

}

}

}

Класс UnmovableObject

package badmagic.model.gameobjects;

import badmagic.model.GameField;

/\*\*

\* Абстрактный класс - непередвигаемый игровой объект.

\*

\* Наследник класса GameObject.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public abstract class UnmovableObject extends GameObject {

/\*\*

\* Конструктор класса.

\*

\* Вызывает конструктор базового класса.

\*

\* @param field ссылка на игровое поле.

\*/

public UnmovableObject(GameField field) {

super(field);

}

}

Класс CollectableObject

package badmagic.model.gameobjects;

import badmagic.model.GameField;

/\*\*

\* Абстрактный класс - подбираемый игровой объект.

\* Наследники класса могут быть подобраны игроком.

\* Их нельзя сдвинуть, но они могут быть перекрыты наследником MovableObject

\* С ними нельзя взаимодействовать как с InteractiveObject.

\* Чтобы подобрать наследника, игрок ддолжен пройти через него.

\*

\* Наследник класса GameObject.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public abstract class CollectableObject extends GameObject {

/\*\*

\* Конструктор класса.

\*

\* Вызывает конструктор базового класса.

\*

\* @param field ссылка на игровое поле.

\*/

public CollectableObject(GameField field) {

super(field);

}

}

Класс InteractiveObject

package badmagic.model.gameobjects;

import badmagic.model.GameField;

/\*\*

\* Абстрактный класс, реализующий интерактвный объект.

\* Наследник GameObject.

\* Наследники класса представляют собой интерактивный объект, который игрок может активировать.

\* Наследник может быть закрыт на замок. В таком случае для его активации,

\* игроку понадобится найти ключ и открыть наследника.

\* @author Alexander Lyashenko

\*/

public abstract class InteractiveObject extends GameObject {

public InteractiveObject(GameField field) {

super(field);

}

/\*\*

\* Абстрактный метод, открывающий объект

\* @param key - ключ

\* @return Флаг успеха действия

\*/

public abstract boolean unlock(CollectableObject key);

/\*\*

\* Абстрактный метод, активирующий объект

\* @return Флаг успеха действия

\*/

public abstract boolean activate();

/\*\*

\* Метод проверки требования к ключу.

\*

\* Возвращает требуемый id или -1 - в случае, если ключ не нужен

\* @return идентификатор ключа

\*/

public int needKey(){

return openId;

}

/\*Идентификатор заклинания, необходимый для активации\*/

int openId;

}

Класс Player

package badmagic.model.gameobjects;

import badmagic.BadMagic;

import badmagic.model.GameField;

import badmagic.navigation.Direction;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.Point;

import java.awt.geom.AffineTransform;

import java.io.IOException;

import java.util.ArrayList;

import javax.imageio.ImageIO;

/\*\*

\* Класс представляет игрока.

\*

\* Наследник класса GameObject. Реализует абстрактные методы. Имеет образ -

\* изображение, направление взгляда и количество ходов.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public class Player extends GameObject {

/\*\*

\* Конструктор класса.

\*

\* Инициализирует поля. Загружает изображение объекта.

\*

\* @param field ссылка на игровое поле.

\*/

public Player(GameField field) {

super(field);

\_inventory = new ArrayList<GameObject>();

loadPic();

}

/\*\*

\* Метод проверки наличия эликсира в инвентаре

\* @return Флаг наличия эликсира в инвентаре

\*/

public boolean haveElixirInInventory() {

for (GameObject item : \_inventory) {

/\*В инвентаре есть эликсир\*/

if (item instanceof Elixir) {

return true;

}

}

return false;

}

/\*\*

\* Метод очистки инвентаря

\*/

public void clearInventory() {

\_inventory.clear();

}

/\*\*

\* Метод получения количества ходов игрока.

\*

\* @return int - количество ходов

\*/

public int getMoves() {

return \_moves;

}

/\*\*

\* Метод установки количества ходов игрока.

\*

\* @param moves количество ходов.

\*/

public void setMoves(int moves) {

\_moves = moves;

}

/\*\*

\* Метод перехода игрока на соседнюю клетку в направлении.

\*

\* Обертка над одноименным методом. Флаг изменения направления взгляда

\* установлен в true.

\*

\* @param moveDirection направление перемещения.

\*/

public void move(Direction moveDirection) {

move(moveDirection, true);

}

/\*\*

\* Метод перемещения объекта на соседнюю клетку.

\*

\* Перемещает игрока и объект на который он смотрит на соседние клетки в

\* направлении, если они свободны.

\*

\* @param moveDirection направление перемещения.

\*/

public void moveObject(Direction moveDirection) {

GameObject nextObject = null;

ArrayList<GameObject> objects = null;

Point nextPos = \_field.getNextPos(\_position, \_gazeDirection);

/\* Получаем объекты в этой позиции и ищем тот, который можно сдвинуть \*/

objects = \_field.getObjects(nextPos);

if (!objects.isEmpty()) {

for (GameObject object : objects) {

if (object instanceof MovableObject) {

nextObject = object;

}

}

}

/\* Пробуем сдвинуть \*/

if (nextObject != null) {

if (\_gazeDirection.equals(moveDirection)) {

/\* Толкаем предмет вперед \*/

((MovableObject) nextObject).move(moveDirection);

this.move(moveDirection, false);

BadMagic.log.info("Толкаем предмет вперед.");

} else if (\_gazeDirection.isOpposite(moveDirection)) {

/\* Тянем предмет на себя \*/

this.move(moveDirection, false);

((MovableObject) nextObject).move(moveDirection);

BadMagic.log.info("Тянем предмет на себя.");

} else {

\_gazeDirection = moveDirection;

BadMagic.log.info("Нельзя сдвинуть в этом направлении.");

}

} else {

\_gazeDirection = moveDirection;

BadMagic.log.info("Нечего двигать.");

}

}

/\*\*

\* Метод отрисовки объекта.

\*

\* @param g среда отрисовки.

\* @param pos позиция отрисоки.

\*/

@Override

public void paint(Graphics g, Point pos) {

AffineTransform at = rotate(\_gazeDirection, pos);

Graphics2D g2d = (Graphics2D) g;

g2d.drawImage(\_image, at, null);

}

/\*\*

\* Метод загрузки изображения объекта.

\*/

@Override

protected void loadPic() {

try {

\_image = ImageIO.read(getClass().getResource(PIC));

} catch (IOException ex) {

ex.printStackTrace();

}

}

/\*\*

\* Метод поворота изображения игрока.

\*

\* @param direction направление поворота.

\* @param pos позиция объекта.

\* @return AffineTransform - объект трансформации.

\*/

private AffineTransform rotate(Direction direction, Point pos) {

int angle = direction.getAngle();

AffineTransform at = AffineTransform.getTranslateInstance(pos.x, pos.y);

at.rotate(Math.toRadians(angle),

\_image.getWidth() / 2,

\_image.getHeight() / 2);

return at;

}

/\*\*

\* Метод перехода игрока на соседнюю клетку в направлении.

\*

\* Перемещает игрока на соседнюю клетку в заданном направлении, если она

\* свободна, изменяет направление взгляда, уменьшает количество ходов,

\* испускает сигнал о перемещении.

\*

\* @param moveDirection направление перемещения.

\* @param needToChangeDirection флаг - менять ли направление взгляда.

\*/

private void move(Direction moveDirection, boolean needToChangeDirection) {

if (needToChangeDirection) {

\_gazeDirection = moveDirection;

BadMagic.log.info("Направление взгляда изменено.");

}

Point newPosition = \_field.getNextPos(\_position, moveDirection);

if (\_field.isPosEmpty(newPosition)) {

if (\_field.isPosHasCollectable(newPosition)) {

/\*Take item to inventory\*/

getItemFromField(newPosition);

}

\_position = newPosition;

\_moves--;

fireObjectMoved();

BadMagic.log.info("Переход на клетку ("

+ \_position.x + ";" + \_position.y + ").");

} else {

BadMagic.log.info("Невозможно перейти на клетку.");

}

}

/\*\*

\* Метод, убирающий предмет с поля в инвентарь

\* @param pos - позиция, с которой нужно собрать предметы

\*/

private void getItemFromField(Point pos) {

ArrayList<GameObject> itemList = \_field.getObjects(pos);

for (GameObject obj : itemList) {

if (obj instanceof CollectableObject) {

\_inventory.add(obj);

\_field.removeObject(obj);

}

}

}

/\*\*

\* Метод, активирующий объект в клетке по направлению

\*/

public void activateObject() {

Point objectPosition = \_field.getNextPos(\_position, \_gazeDirection);

for (GameObject obj : \_field.getObjects(objectPosition)) {

if (obj instanceof InteractiveObject) {

/\*Запрос и проверка ключа\*/

/\*Попробовать открыть\*/

if (!((InteractiveObject) obj).activate()) {

for (GameObject o : \_inventory) {

if (((InteractiveObject) obj).unlock((CollectableObject) o)) {

\_inventory.remove(o);

((InteractiveObject) obj).activate();

break;

}

}

break;

}

}

}

}

///////////////////////////// Данные //////////////////////////////////////

/\*\*

\* Количество ходов игрока

\*/

private int \_moves;

/\*\*

\* Направление взгляда игрока

\*/

private Direction \_gazeDirection = Direction.north();

/\*\*

\* Путь к файлу с изображением

\*/

private static final String PIC = "/badmagic/resources/goat.png";

/\*\*

\* Инвентарь

\*/

private ArrayList<GameObject> \_inventory;

}

Класс Elixir

package badmagic.model.gameobjects;

import badmagic.model.GameField;

import java.awt.Graphics;

import java.awt.Point;

import java.io.IOException;

import javax.imageio.ImageIO;

/\*\*

\* Класс представляет игровой объект - элексир.

\*

\* Наследник класса CollectableObject. Реализует абстрактные методы.

\* Имеет образ - изображение.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public class Elixir extends CollectableObject {

/\*\*

\* Конструктор класса.

\*

\* Инициализирует поля. Загружает изображение объекта.

\*

\* @param field ссылка на игровое поле.

\*/

public Elixir(GameField field) {

super(field);

loadPic();

}

/\*\*

\* Метод отрисовки объекта.

\*

\* @param g среда отрисовки.

\* @param pos позиция отрисоки.

\*/

@Override

public void paint(Graphics g, Point pos) {

g.drawImage(\_image, pos.x, pos.y, null);

}

/\*\*

\* Метод загрузки изображения объекта.

\*/

@Override

protected void loadPic() {

try {

\_image = ImageIO.read(getClass().getResource(PIC));

} catch ( IOException ex ) {

ex.printStackTrace();

}

}

///////////////////////////// Данные //////////////////////////////////////

/\*\* Путь к файлу с изображением \*/

private static final String PIC = "/badmagic/resources/Elixir.png";

}

Класс WoodenTable

package badmagic.model.gameobjects;

import badmagic.model.GameField;

import java.awt.Graphics;

import java.awt.Point;

import java.io.IOException;

import javax.imageio.ImageIO;

/\*\*

\* Класс представляет игровой объект - деревянный стол.

\*

\* Наследник класса MovableObject. Реализует абстрактные методы.

\* Имеет образ - изображение.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public class WoodenTable extends MovableObject {

/\*\*

\* Конструктор класса.

\*

\* Инициализирует поля. Загружает изображение объекта.

\*

\* @param field ссылка на игровое поле.

\*/

public WoodenTable(GameField field) {

super(field);

loadPic();

}

/\*\*

\* Метод отрисовки объекта.

\*

\* @param g среда отрисовки.

\* @param pos позиция отрисоки.

\*/

@Override

public void paint(Graphics g, Point pos) {

g.drawImage(\_image, pos.x, pos.y, null);

}

/\*\*

\* Метод загрузки изображения объекта.

\*/

@Override

protected void loadPic() {

try {

\_image = ImageIO.read(getClass().getResource(PIC));

} catch ( IOException ex ) {

ex.printStackTrace();

}

}

///////////////////////////// Данные //////////////////////////////////////

/\*\* Путь к файлу с изображением \*/

private static final String PIC = "/badmagic/resources/woodentable.png";

}

Класс Wall

package badmagic.model.gameobjects;

import badmagic.model.GameField;

import java.awt.Graphics;

import java.awt.Point;

import java.io.IOException;

import javax.imageio.ImageIO;

/\*\*

\* Класс представляет игровой объект - стену.

\*

\* Наследник класса UnmovableObject. Реализует абстрактные методы.

\* Имеет образ - изображение.

\*

\* @author Alexander Petrenko, Alexander Lyashenko

\*/

public class Wall extends UnmovableObject {

/\*\*

\* Конструктор класса.

\*

\* Инициализирует поля. Загружает изображение объекта.

\*

\* @param field ссылка на игровое поле.

\*/

public Wall(GameField field) {

super(field);

loadPic();

}

/\*\*

\* Метод отрисовки объекта.

\*

\* @param g среда отрисовки.

\* @param pos позиция отрисоки.

\*/

@Override

public void paint(Graphics g, Point pos) {

g.drawImage(\_image, pos.x, pos.y, null);

}

/\*\*

\* Метод загрузки изображения объекта.

\*/

@Override

protected void loadPic() {

try {

\_image = ImageIO.read(getClass().getResource(PIC));

} catch ( IOException ex ) {

ex.printStackTrace();

}

}

///////////////////////////// Данные //////////////////////////////////////

/\*\* Путь к файлу с изображением \*/

private static final String PIC = "/badmagic/resources/Wall.png";

}

Класс Bookshelf

package badmagic.model.gameobjects;

import badmagic.model.GameField;

import java.awt.Graphics;

import java.awt.Point;

import java.io.IOException;

import javax.imageio.ImageIO;

/\*\*

\* Класс реализующий книжную полку (сундук-хранилище).

\* Является наследником InteractiveObject.

\* Полка может содержать одно заклинание (в виде численного идентификатора).

\* После активации удаляется с поля, на своем месте оставляя объект-заклинание

\* Имеет собственное графическое представление

\* @author Alexander Lyashenko

\*/

public class Bookshelf extends InteractiveObject {

/\*\*

\* Конструктор класса.

\*

\* Инициализирует поля. Загружает изображение объекта.

\*

\* @param field ссылка на игровое поле.

\*/

public Bookshelf(GameField field) {

super(field);

openId = -1;

loadPic();

}

public void setSpell(int spellId){

\_spellId = spellId;

}

/\*\*

\* Метод, устанавливающий ключ для объекта

\* @param key идентификатор ключа

\*/

public void setLock (int key){

openId = key;

}

/\*\*

\* Метод для открытия объекта (снятия с него замка)

\* @param key - ключ

\* @return Флаг - открылся ли объект

\*/

@Override

public boolean unlock(CollectableObject key){

if (((Spell)key).getId() == openId){

openId = -1;

return true;

}

else {

return false;

}

}

/\*\*

\* Метод, активирующий объект.

\* Полка убирает себя с поля, и оставляет на своем месте заклинание.

\* @return Флаг успеха действия

\*/

@Override

public boolean activate() {

if (openId == -1){

/\*Удаление полки с поля\*/

Point temp = new Point(\_position);

\_field.removeObject(this);

/\*Добавление заклинания на его место\*/

if (\_spellId != -1){

GameObject newSpell = new Spell(\_field);

((Spell)newSpell).setId(\_spellId);

\_field.addObject(temp, (Spell)newSpell);

}

return true;

}

else{

return false;

}

}

/\*\*

\* Метод отрисовки объекта.

\*

\* @param g среда отрисовки.

\* @param pos позиция отрисоки.

\*/

@Override

public void paint(Graphics g, Point pos) {

g.drawImage(\_image, pos.x, pos.y, null);

}

/\*\*

\* Метод загрузки изображения объекта.

\*/

@Override

protected void loadPic() {

try {

\_image = ImageIO.read(getClass().getResource(PIC));

} catch ( IOException ex ) {

ex.printStackTrace();

}

}

///////////////////////////// Данные //////////////////////////////////////

/\*\* Путь к файлу с изображением \*/

private static final String PIC = "/badmagic/resources/bookshelf.png";

/\*\* Хранимое заклинание (-1 если отсутствует)\*/

private int \_spellId = -1;

}

Класс Spell

package badmagic.model.gameobjects;

import badmagic.model.GameField;

import java.awt.Graphics;

import java.awt.Point;

import java.io.IOException;

import javax.imageio.ImageIO;

/\*\*

\* Класс, реализующий заклинание - ключ.

\* Является наследником CollectableObject.

\* Имеет собственный численный идентификатор.

\* Имеет собственное графическое представление.

\*

\* @author Alexander Lyashenko

\*/

public class Spell extends CollectableObject {

/\*\*

\* Конструктор класса.

\*

\* Инициализирует поля. Загружает изображение объекта.

\*

\* @param field ссылка на игровое поле.

\*/

public Spell(GameField field) {

super(field);

loadPic();

}

/\*\*

\* Метод, устанавливающий идентификатор для заклинания.

\*

\* @param \_id идентификатор для заклинания

\*/

public void setId (int \_id){

id = \_id;

}

/\*\*

\* Метод, возвращающий текущий идентификатор заклинания.

\* @return текущий идентификатор заклинания

\*/

public int getId (){

return id;

}

/\*\*

\* Метод отрисовки объекта.

\*

\* @param g среда отрисовки.

\* @param pos позиция отрисоки.

\*/

@Override

public void paint(Graphics g, Point pos) {

g.drawImage(\_image, pos.x, pos.y, null);

}

/\*\*

\* Метод загрузки изображения объекта.

\*/

@Override

protected void loadPic() {

try {

\_image = ImageIO.read(getClass().getResource(PIC));

} catch ( IOException ex ) {

ex.printStackTrace();

}

}

///////////////////////////// Данные //////////////////////////////////////

/\*\*

\* Путь к файлу с изображением

\*/

private static final String PIC = "/badmagic/resources/spell.png";

/\*\*

\* Идентификатор открываемого объекта

\*/

private int id = -1;

}

Класс Teleport

package badmagic.model.gameobjects;

import badmagic.model.GameField;

import java.awt.Graphics;

import java.awt.Point;

import java.io.IOException;

import javax.imageio.ImageIO;

/\*\*

\* Класс, представляющий собой телепортатор.

\* Является наследником InteractiveObject

\* Имеет собственное графическое представление

\* Хранит позицию для телепортации

\*

\* @author Alexander Lyashenko

\*/

public class Teleport extends InteractiveObject {

/\*\*

\* Конструктор класса.

\*

\* Инициализирует поля. Загружает изображение объекта.

\*

\* @param field ссылка на игровое поле.

\*/

public Teleport(GameField field) {

super(field);

loadPic();

openId = -1;

}

/\*\*

\* Метод, устанавливающий позицию для телепортации.

\* Позиция должна быть в пределах игрового поля

\*

\* @param tPosition позиция для телепортации

\*/

public void setTPosition(Point tPosition){

\_tPosition = new Point(tPosition);

}

/\*\*

\* Метод, устанавливающий ключ для объекта

\* @param key идентификатор ключа

\*/

public void setLock (int key){

openId = key;

}

/\*\*

\* Метод отрисовки объекта.

\*

\* @param g среда отрисовки.

\* @param pos позиция отрисоки.

\*/

@Override

public void paint(Graphics g, Point pos) {

g.drawImage(\_image, pos.x, pos.y, null);

}

/\*\*

\* Метод загрузки изображения объекта.

\*/

@Override

protected void loadPic() {

try {

\_image = ImageIO.read(getClass().getResource(PIC));

} catch (IOException ex) {

ex.printStackTrace();

}

}

/\*\*

\* Метод для открытия объекта (снятия с него замка)

\* @param key - ключ

\* @return Флаг - открылся ли объект

\*/

@Override

public boolean unlock(CollectableObject key) {

if (((Spell) key).getId() == openId) {

openId = -1;

return true;

} else {

return false;

}

}

/\*\*

\* Метод активации объекта.

\* Для конкретного объекта - телепортация игрока.

\* @return Флаг, возвращающий успех действия

\*/

@Override

public boolean activate() {

if (openId == -1){

for (GameObject o : \_field.getObjects()){

if (o instanceof Player){

((Player)o).setPosition(\_tPosition);

break;

}

}

return true;

}

else{

return false;

}

}

///////////////////////////// Данные //////////////////////////////////////

/\*\*

\* Путь к файлу с изображением

\*/

private static final String PIC = "/badmagic/resources/teleport.png";

/\*\*

\* Позиция телепортации

\*/

private Point \_tPosition;

}