Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Волгоградский государственный технический университет»

Факультет электроники и вычислительной техники

Кафедра «Программное обеспечение автоматизированных систем»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе**

по дисциплине «Объектно-ориентированный анализ и программирование»

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

(индивидуальное задание – вариант №25)

Студент: Серпинин Е.И.

Группа: ПрИн-366

Работа зачтена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «11»     июня   2022 г.

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

Волгоград 2022 г.

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего 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 \_\_\_\_\_\_\_\_\_\_ Орлова Ю.А.

**Задание**

**на курсовую работу**

Студент: Серпинин Е.И.

Группа: ПрИн-366

1. Тема: «Проектированиепрограммы с использованием объектно-ориентированного подхода» (индивидуальное задание – вариант №25)

Утверждена приказом от «24» января 2022г. № 101-ст

2. Срок представления работы к защите « 04 »   июня  2022 г.

3. Содержание пояснительной записки:

формулировка задания, требования к программе, структура программы, типовые процессы в программе, человеко-машинное взаимодействие, код программы и модульных тестов

4. Перечень графического материала:

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

5. Дата выдачи задания «13» февраля 2022 г.

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

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Серпинин Е.И.

«13» февраля 2022 г.

**Содержание**

[1 Формулировка задания 4](#_Toc95664339)

[2 Нефункциональные требования 5](#_Toc95664340)

[3 Первая итерация разработки 6](#_Toc95664341)

[3.1 Формулировка упрощенного варианта задания 6](#_Toc95664342)

[3.2 Функциональные требования (сценарии) 6](#_Toc95664343)

[3.3 Словарь предметной области 10](#_Toc95664344)

# 1 Формулировка задания

Правила игры «Следы»:

* Игровое поле состоит из шестиугольников, игрок располагается в одном из них.
* Игрок может перемещаться в любой из соседних шестиугольников, если тот проходим.
* При перемещении он оставляет за собой след (шестиугольник закрашивается определенным цветом).
* Игрок не может повторно наступать на шестиугольники, на которых оставлены следы того же цвета, что и у игрока.
* Цель - игрок должен достичь целевого шестиугольника.
* На поле могут быть разбросаны ключи, которые игрок должен собрать. Посещение целевого шестиугольника до сбора всех ключей не приводит к завершению игры.
* Игрок не оставляет следов на целевом шестиугольнике и может посещать его несколько раз.

Подвариант 1: необходимо предусмотреть в программе точки расширения, используя которые можно реализовать вариативную часть программы (в дополнение к базовой функциональности).

Вариативность: предусмотреть различные критерии окончания игры и их комбинации (через отношения И, ИЛИ, НЕ). Игроку должно сообщаться условие окончания игры.

НЕ изменяя ранее созданные классы, а используя точки расширения, реализовать: следующие критерии окончания игры: - собраны все ключи; - достигнут выход; - длина пройденного пути равна, больше или меньше заданной.

# 2 Нефункциональные требования

1. Программа должна быть реализована на языке Java SE 12 с использованием стандартных библиотек, в том числе, библиотеки Swing.
2. Форматирование исходного кода программы должно соответствовать Java Code Conventions, September 12, 1997.

# 3 Первая итерация разработки

### 3.1 Формулировка упрощенного варианта задания

Правила игры «Следы»:

* Игровое поле состоит из шестиугольников, игрок располагается в одном из них.
* Игрок может перемещаться в любой из соседних шестиугольников, если тот проходим.
* При перемещении он оставляет за собой след (шестиугольник закрашивается определенным цветом).
* Игрок не может повторно наступать на шестиугольники, на которых оставлены следы того же цвета, что и у игрока.
* Цель - игрок должен достичь целевого шестиугольника.
* На поле могут быть разбросаны ключи, которые игрок должен собрать. Посещение целевого шестиугольника до сбора всех ключей не приводит к завершению игры.
* Игрок не оставляет следов на целевом шестиугольнике и может посещать его несколько раз.

### 

### 3.2 Функциональные требования (сценарии)

**Сценарий «Игра»**

1. Пользователь инициирует начало игры
2. Игра создает карту из шестиугольных Ячеек и размещает на них ключи, Робота и Выход
3. Игра обрабатывает игровой цикл пока игра не завершена
   1. По желанию пользователя Робот делает шаг в одну из сторон шестиугольника
4. Игра завершается

**Сценарий «Игра не завершена»**

1. Игра проверяет доходимость Робота до ключей и выхода. Если пути нет, то она считает, что игрок проиграл
2. Игра проверяет находиться ли Робот на выходе, и остались ли на карте ключи. Если Робот на выходе и на карте нет ключей, то Игра считает, что игрок победил

**Сценарий «Робот делает шаг в Ячейку без своего следа»**

1. По указанию робота ячейка возвращает след
2. Если след не совпадает со следом робота
   1. Робот меняет свою координату на Поле, соответствующей пустой Ячейке
   2. Робот оставляет свой след на Ячейке, на которую наступил

**Сценарий «Робот делает шаг в Ячейку с ключем»**

1. Робот подбирает Ключ и убирает его с текущей Ячейки

**Сценарий «Робот делает шаг в ячейку Выхода»**

1. Робот меняет свою координату на Поле, соответствующей ячейке Выхода

### 3.3 Словарь предметной области

**Игра** – способна создавать карту из шестиугольников, размещать на ней ключи, Робота и Выход, так же проверяет условия победы игрока

**Поле** - прямоугольная область, состоящая из шестиугольных ячеек. Знает о роботе и ключах

**Ячейка** - шестиугольная область поля. Одновременно может содержать в себе только робота, или ключ, или быть пустой. Ячейки могут содержать на себе след

**Выход** - разновидность ячейки. На поле может быть только одна ячейка, являющаяся выходом. На ней не может быть оставлен след

**Робот** – умеет менять свою координату на поле, подбирать ключи, оставлять след на ячейке. Определяет по следу ячейки может ли он на неё наступить. Им управляет игрок.

**Ключ** – объект в ячейке. Нужен для победы игрока

**След** – цвет ячейки

### 3.4 Структура программы на уровне классов

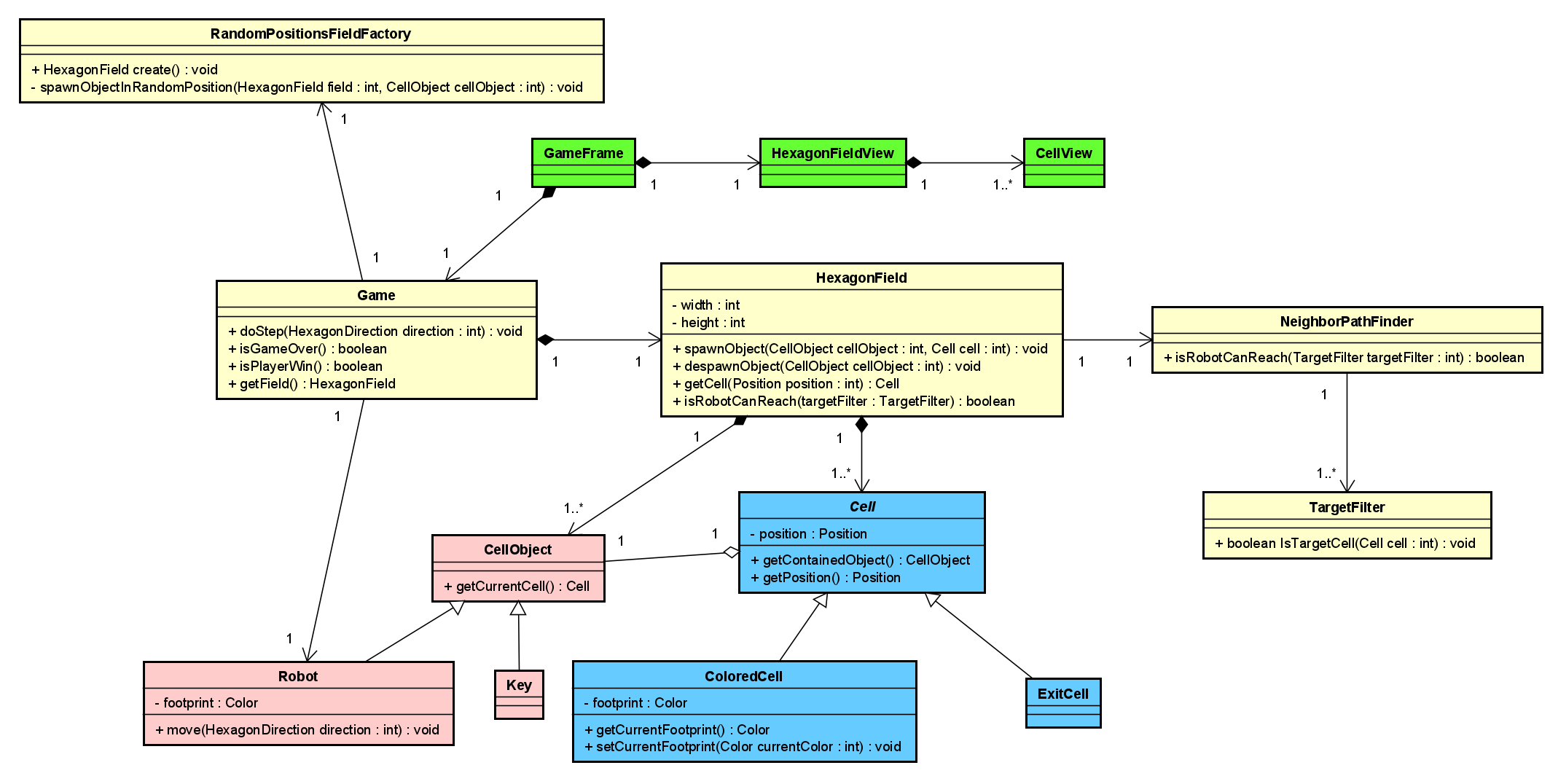


Диаграмма классов вычислительной модели

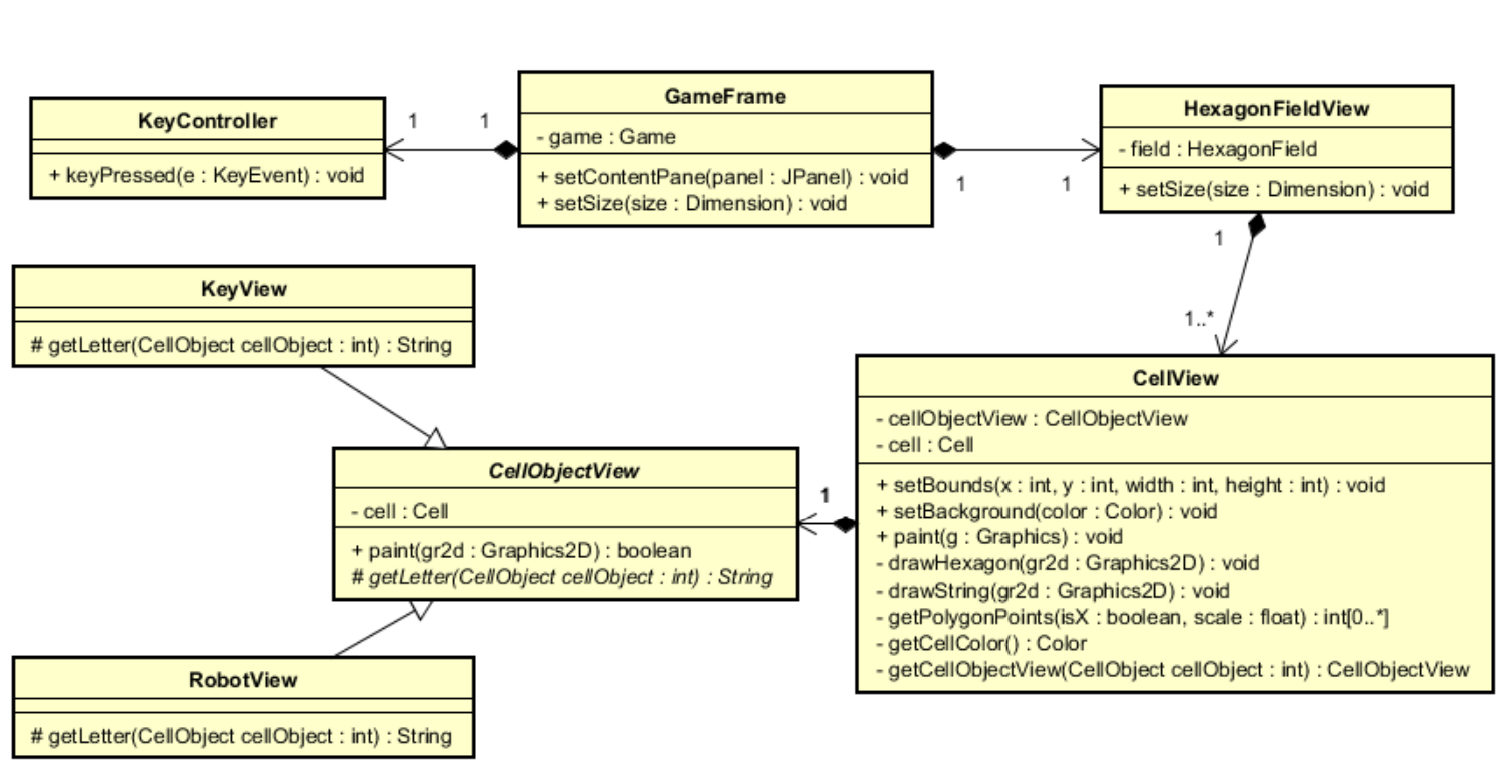
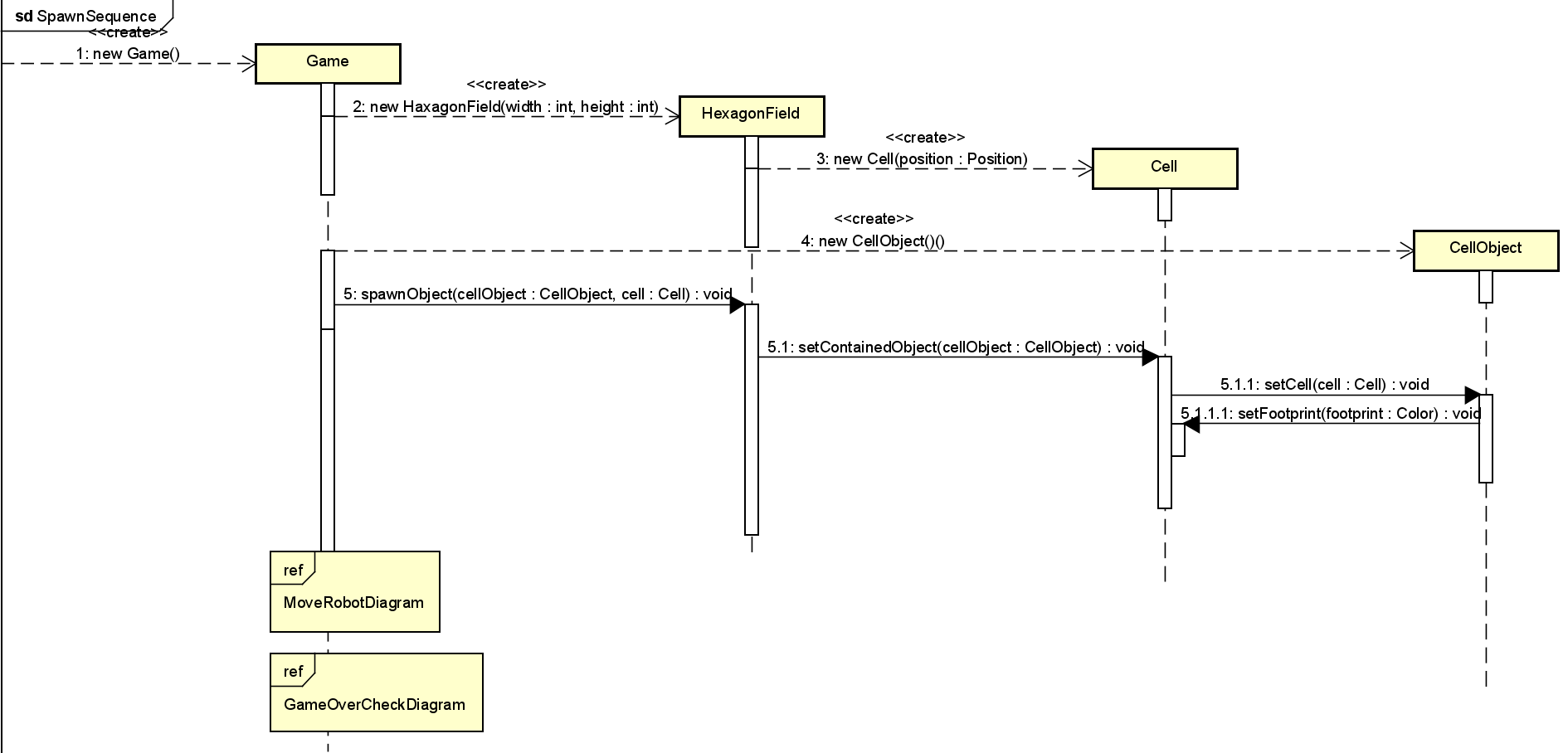
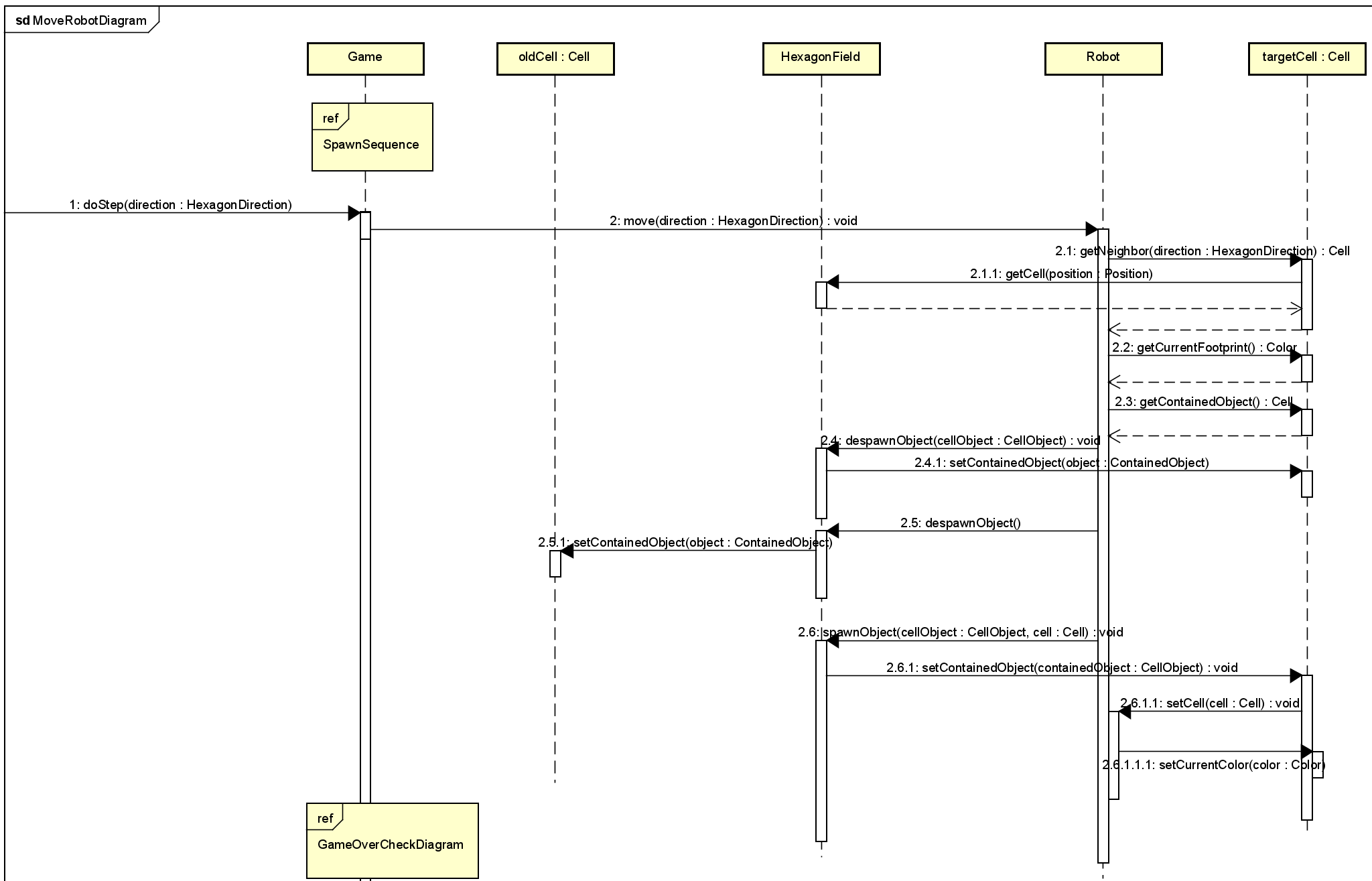


Диаграмма классов представления

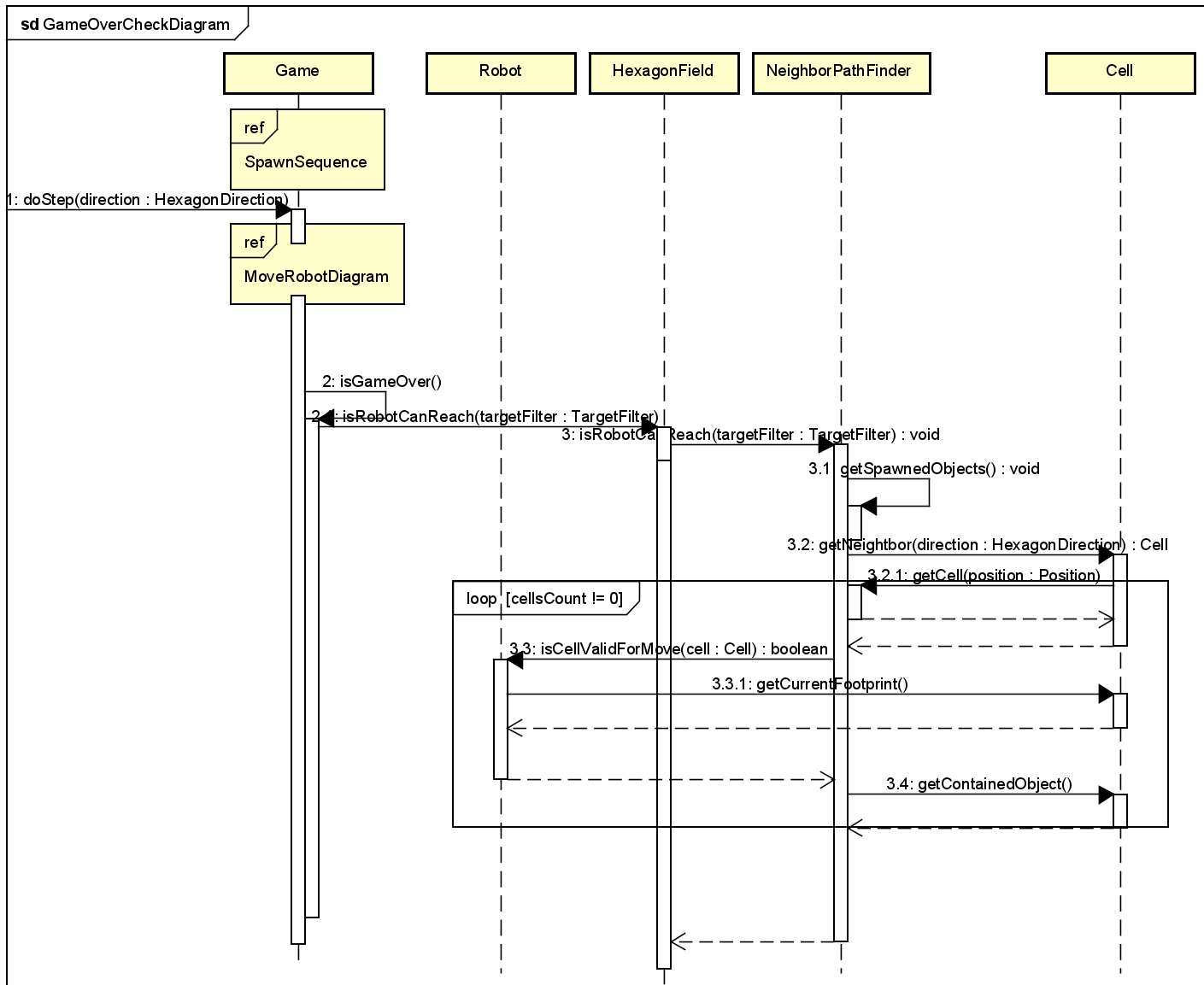
### 3.5 Типовые процессы в программе



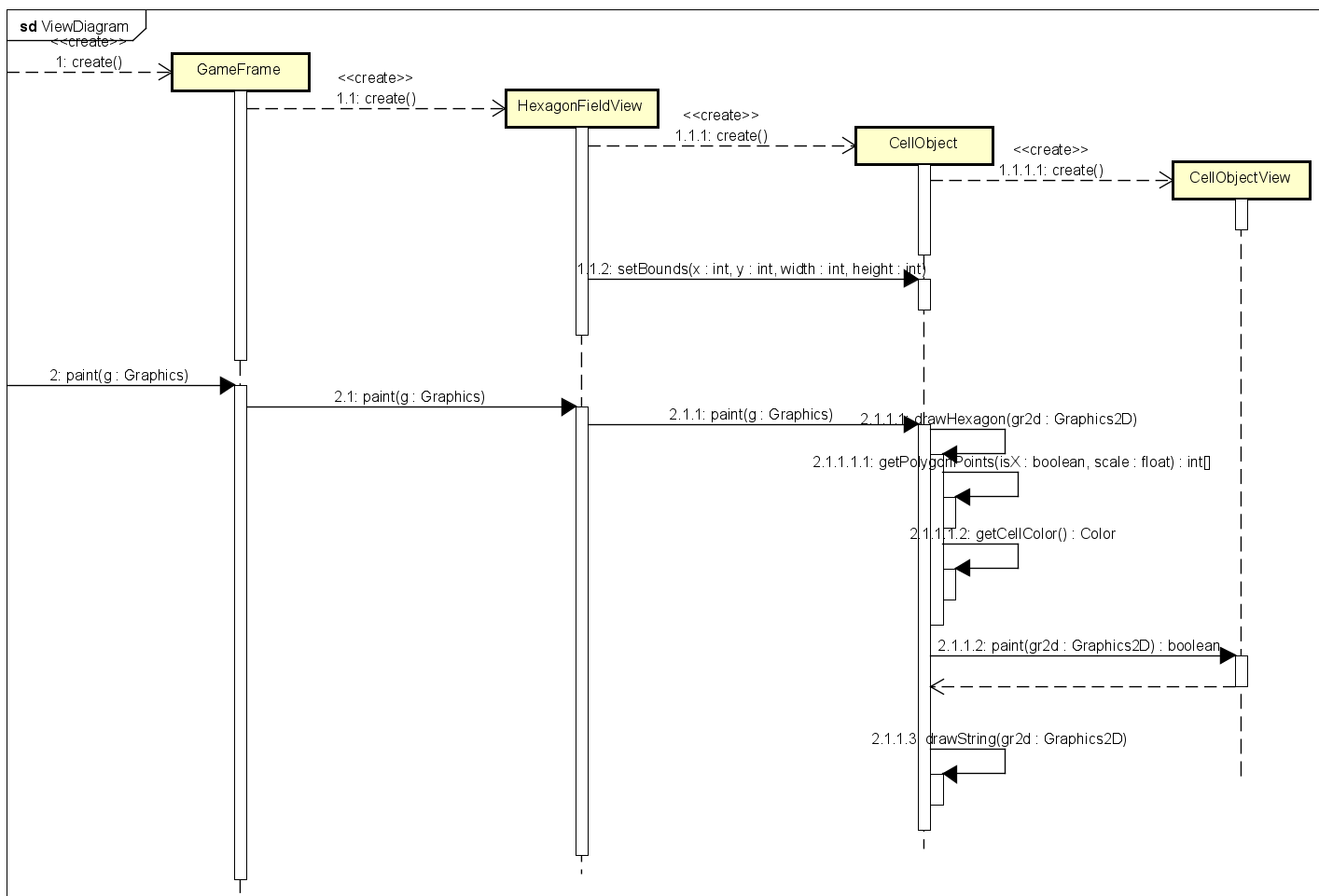
Создание игровой карты, расстановка объектов и игрока



Ход робота в клетку без следа с ключом



Проверка условий окончания игры

  
Создание и отрисовка представления

### 3.6 Реализация ключевых классов

### 3.7 Реализация ключевых тестовых случаев

# 4 Вторя итерация разработки

### 4.1 Функциональные требования (сценарии)

**Сценарий «Игра не завершена» при правиле «Собрать все ключи»**

1. Игра спрашивает у Обработчика правил игры, завершена ли игра
2. Обработчик правил игры спрашивает у каждого Правила завершена ли игра
3. Одно из Правил игры спрашивает у Поля доходимость Робота до Ключей. Если пути нет, то Игра считает, что игрок проиграл
4. Одно из Правил игры спрашивает у Поля остались ли на карте ключи. Если на карте нет ключей, то Игра считает, что игрок победил

**Сценарий «Игра не завершена» при правиле «Шагов меньше заданного и выход достигнут»**

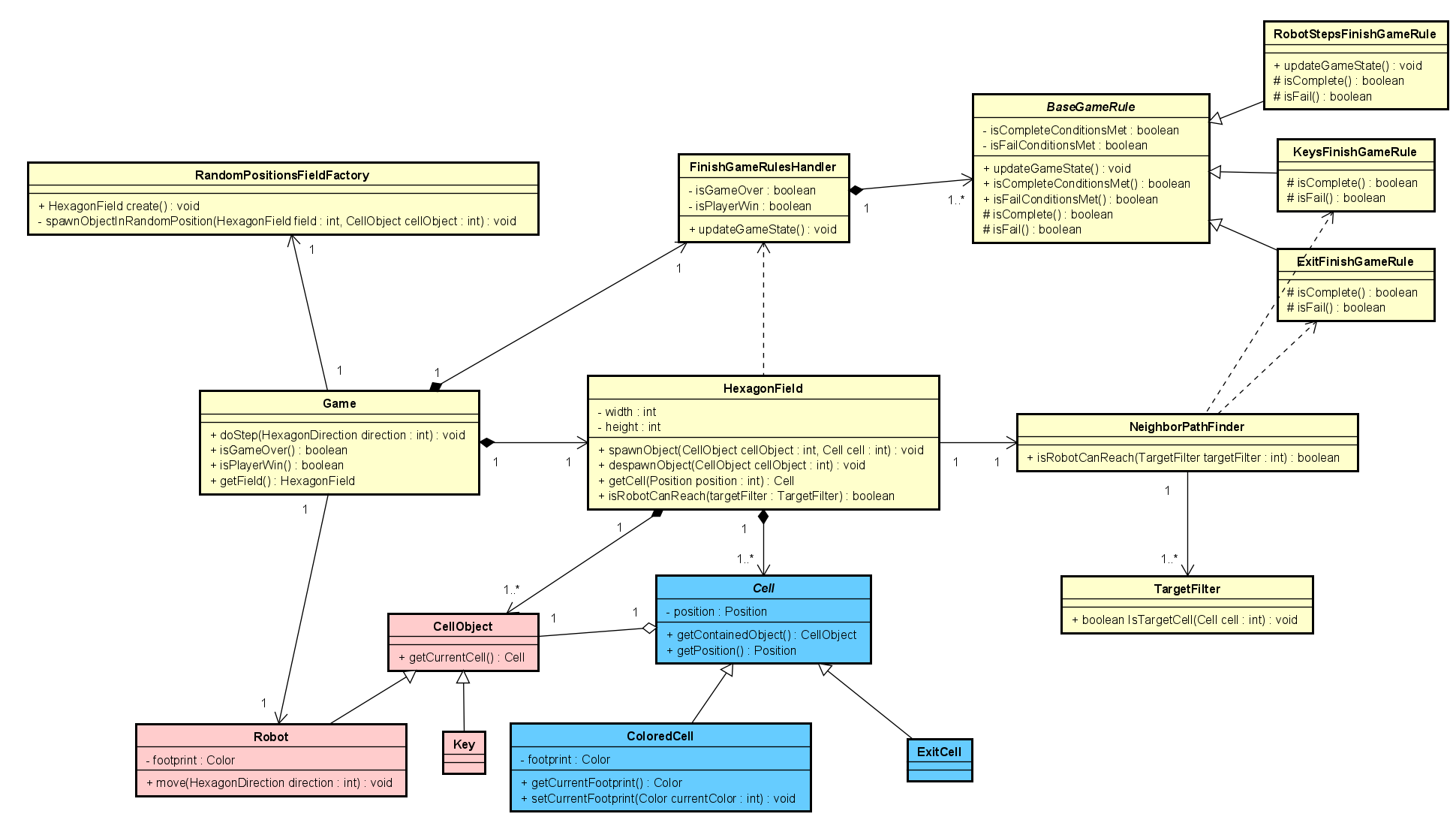
1. Игра спрашивает у Обработчика правил игры, завершена ли игра
2. Обработчик правил игры спрашивает у каждого Правила завершена ли игра
3. Одно из Правил игры спрашивает у Поля доходимость Робота до Выхода. Если пути нет, то Игра считает, что игрок проиграл
4. Одно из Правил игры спрашивает у Поля количество закрашенных клеток роботом. Если их кол-во превышает или равняется заданному значению, то Игра считает, что игрок проиграл

### 4.2 Словарь предметной области

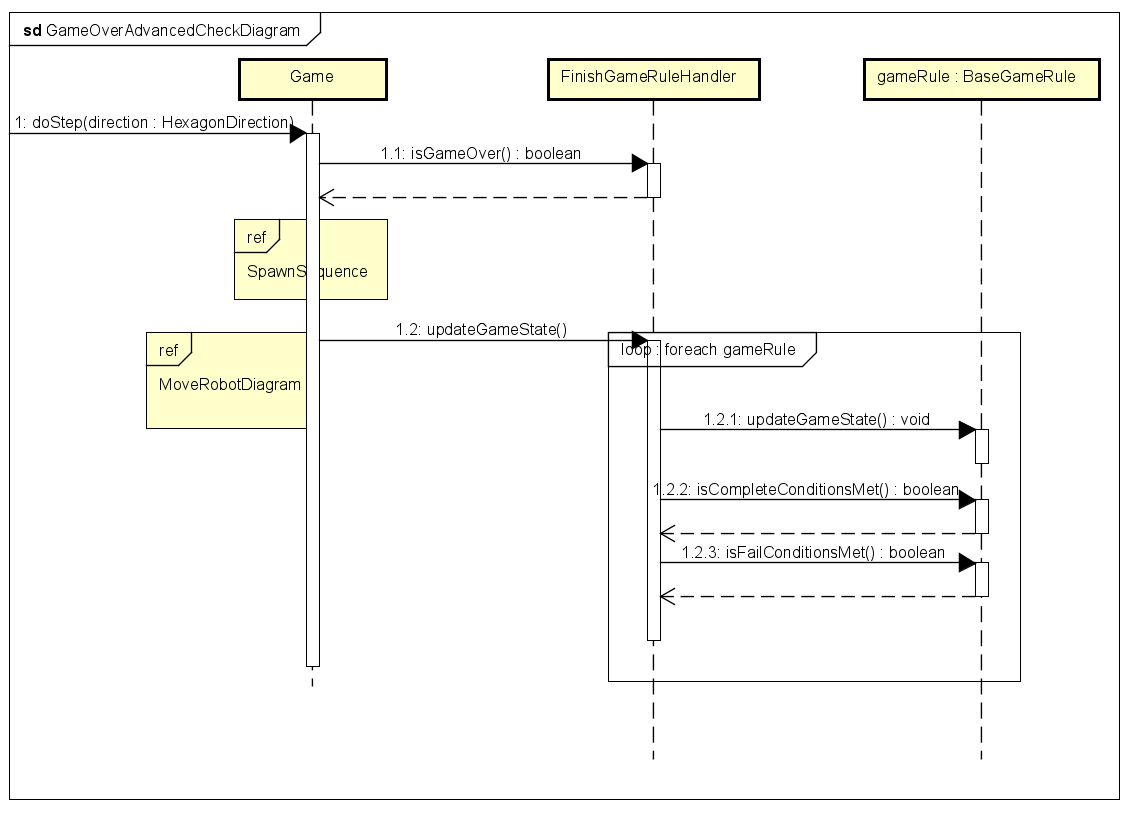
**Правило** – атомарное условие, которое должно выполняться или не выполняться для победы игрока

**Правило игры** – совокупность Правил, связанных логическими операторами И, ИЛИ и НЕ (без вложенности), которые формируют окончательное правило победы игрока в игровой сессии

### 4.3 Структура программы на уровне классов



### 4.4 Типовые процессы в программе



# 5 Человеко-машинное взаимодействие

Общий вид главного экрана программы представлен ниже. На нём располагается игровое поле, на котором изображен робот, 3 ключа и ячейка выхода, а так же 2е закрашенные ячейки.

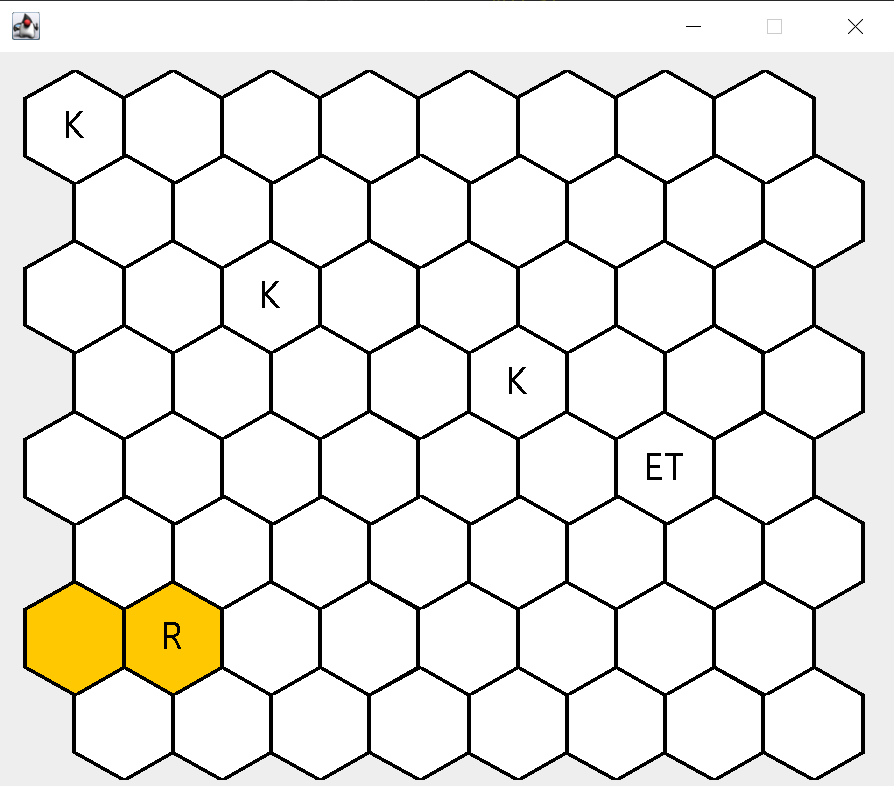


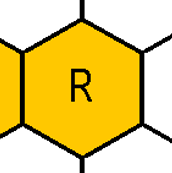
Рис. 1. Общий вид главного экрана программы

Управление активным роботом пользователь осуществляет с помощью клавиатуры.

Q – движение влево-вверх

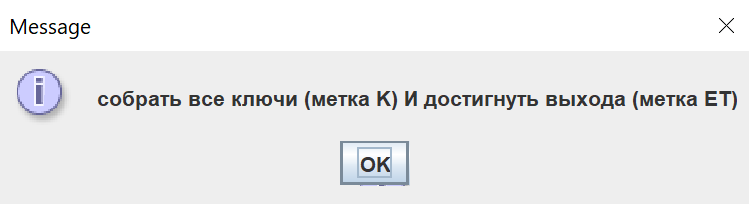
W – движение вправо-вверх  
 A – движение влево  
 S – движение вправо  
 Z – движение влево-вниз  
 X – движение вправо-вниз

Изображение робота представлено на рисунке 2. Под ним ячейка закрашена цветом его следа.

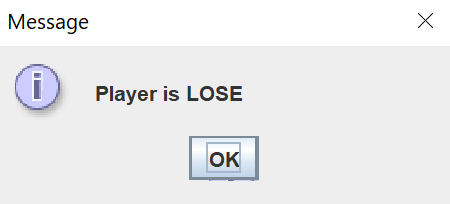
  
Рис. 2. Изображение робота

Ключ представлен аналогичным образом, но буквой “K”. Ячейка выхода буквами “ET”.

При запуске игры (Рис. 3) появляется условия победы. Каждый запуск выдаются случайные из определенного набора.

  
Рис. 3. Сообщение с условиями победы

Когда игрок проигрывает или выигрывает, то появляется подобное окно (Рис. 4) с сообщением о победе или проигрыше.

  
Рис. 4. Сообщение о проигрыше