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 №07)

Студент: Воробьев А.С.

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: Воробьев А.С.

Группа: ПрИн-366

1. Тема: «Проектированиепрограммы с использованием объектно-ориентированного подхода» (индивидуальное задание – вариант №07)

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: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Воробьев А.С.

«13» февраля 2022 г.

**Содержание**

[Формулировка задания 3](#_Toc99976692)

[Нефункциональные требования 4](#_Toc99976693)

[Первая итерация разработки 5](#_Toc99976694)

# Формулировка задания

Правила игры «Глупый робот и умный робот»:

* имеется лабиринт, в котором действует маленький умный робот и большой глупый робот, их позиции задаются;
* маленьким роботом управляет человек, а большим - компьютер;
* игра ведется пошагово;
* в лабиринте имеются болота, а также вертикальные и горизонтальные препятствия (на стыке клеток);
* цель маленького робота – достичь выхода, цель большого робота – поймать маленького робота;
* большой робот рационально действует только, когда маленький робот попадает в поле его действия размером 2×2, в остальных случаях робот идет либо вправо, либо влево по направлению к маленькому роботу;
* маленькому роботу заходить в болото нельзя, а большой робот пропускает три хода.

Дополнительные требования:

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

Вариативность:

* предсоздание других ландшафтов, влияющих на перемещение обоих роботов (количество пройденных шагов и направление движения) и зависящих от сезона года. Ландшафты в разных состояниях должны визуально различаться.

Реализовать:

* болото, которое расширяется на новые клетки в период осени и весны (по окончанию дождливых периодов болото возвращается к исходным размерам), и замерзает в зимний период. По замерзшему болоту роботы передвигаются сразу на две клетки до препятствия, т.е. скользят.
* песок, который заставляет "буксовать" роботов, т.е. пропускать ход.

# 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.

# Первая итерация разработки

* + 1. **3.1 Формулировка упрощенного варианта задания**

Правила игры «Глупый робот и умный робот»:

* имеется лабиринт, в котором действует маленький умный робот и большой глупый робот, их позиции задаются случайным образом;
* маленьким роботом управляет человек, а большим - компьютер;
* игра ведется пошагово;
* в лабиринте имеются болота, а также вертикальные и горизонтальные препятствия (на стыке клеток);
* цель маленького робота – достичь выхода, цель большого робота – поймать маленького робота;
* большой робот рационально действует только, когда маленький робот попадает в поле его действия размером 2×2, в остальных случаях робот идет либо вправо, либо влево по направлению к маленькому роботу;
* маленькому роботу заходить в болото нельзя, а большой робот пропускает три хода.
  + 1. **3.2 Функциональные требования (сценарии)**

**1) Сценарий «Играть за маленького робота: выйти из лабиринта, не попавшись большому роботу»**

1. Пользователь инициирует начало игры.
2. Игра создаёт поле из ячеек и размещает на нем лабиринт из стен, болото(а), точку выхода и двух роботов: большого и маленького.
3. Игра делает активным маленького робота
4. Делать
   1. Робот перемещается на соседнюю ячейку.
   2. Игра делает активным следующего робота, который располагается на поле.

Пока маленький робот находится внутри поля.

1. Игра считает победителем игрока, если он “телепортирован” точкой выхода или компьютер, если маленький робот был уничтожен.

**2) Сценарий «игра создаёт поле из ячеек и размещает на нем лабиринт из стен, болото(а), точку выхода и двух роботов»**

1. Игра инициирует создание поля размером W на H ячеек.
2. Игра посредством лабиринта создает и расставляет последовательности стен по периметру поля.
3. Игра посредством лабиринта создает последовательности стен в середине поля.
4. Игра посредством лабиринта создает болото(а) случайным образом случайной формы.
5. Игра создаёт двух роботов.
6. Игра помещает на поле созданных роботов.
7. Игра создает и помещает на поле ячейку выхода.

**4) Сценарий «робот перемещается на соседнюю ячейку».**

1. Робот запрашивает у ячейки, в которой он находится, стену в направлении своего движения
2. Ячейка сообщает, что стены нет
3. Робот запрашивает у ячейки, в которой он находится, соседнюю ячейку в направлении своего движения
4. Ячейка сообщает о ячейке, с которой соседствует
5. Робот запрашивает у соседней ячейки робота (который возможно в ней находится)
6. Соседняя ячейка сообщает, что робот отсутствует в ней
7. Робот просит ячейку, в которой он находится, изъять его из нее
8. Ячейка извлекает робота из себя
9. Робот просит соседнюю ячейку поместить себя в нее
10. Ячейка помещает робота в себя, т.к. в ней нет другого робота
11. Робот определяет, находится ли в новой ячейке болото

**4.1) Альтернативный сценарий «ход большого робота»**

1. Сценарий начинается перед п. 1 Главного сценария 4
2. Большой робот вычисляет расстояние до маленького робота
3. Так как маленький робот находится в поле действия большого робота, большой робот составляет оптимальный маршрут до маленького робота
4. Большой робот выбирает направление движения по направлению к следующей клетке в построенном оптимальном пути
5. Сценарий переходит к п. 1 сценария 4

**4.1.1) Альтернативный сценарий «ход большого робота, пропуск хода»**

1. Сценарий начинается перед п. 1 альтернативного сценария «ход большого робота»
2. Так как робот все еще «оштрафован» за попадание в ячейку с болотом, он пропускает ход
3. Сценарий переходит к п. 1

**4.1.2) Альтернативный сценарий «ход большого робота, маленький робот вне поле действия большого робота»**

1. Сценарий начинается после п. 2 альтернативного сценария «ход большого робота»
2. Так как маленький робот находится вне поле действия большого робота, большой робот выбирает направление движения в сторону маленького робота из направлений “влево” и “вправо” или “пустое направление”
3. Сценарий переходит к п. 1 Главного сценария 4

**4.2) Альтернативный сценарий «в соседней ячейке находится робот»**

1. Сценарий начинается после п. 5 сценария 4
2. Соседняя ячейка сообщает, что в ней находится робот
3. Сценарий переходит к п 5.2. Главного сценария.

**4.2.1) Альтернативный сценарий «в соседней ячейке находится робот, ход большого робота»**

1. Сценарий начинается после п. 2 сценария 4.1
2. Большой робот инициирует уничтожение маленького робота
3. Сценарий переходит к п 5.1. Главного сценария

**4.3) Альтернативный сценарий «между соседней ячейкой и текущей находится стена»**

1. Сценарий начинается после п. 1 сценария 4
2. Ячейка сообщает, что стена имеется
3. Сценарий переходит к п. 5.2. Главного сценария

**4.4) Альтернативный сценарий «конец хода большого робота»**

1. Сценарий начинается после п. 4.1 Главного сценария
2. Большой робот анализирует расстояние до маленького робота
3. Сценарий переходит к п. 4.2 Главного сценария

**4.4.1) Альтернативный сценарий** **«конец хода большого робота, маленький робот в соседней клетке»**

1. Сценарий начинается после п. 2 альтернативного сценария «конец хода большого робота»
2. Так как маленький робот находится в соседней клетке, большой робот инициирует уничтожение маленького робота
3. Большой робот сообщает игре о том, что маленький робот уничтожен
4. Сценарий переходит к п. 5 Главного сценария

**4.5) Альтернативный сценарий** **«маленький робот попадает в болото»**

1. Сценарий начинается после п. 11 сценария «робот перемещается на соседнюю ячейку»
2. Так как маленький робот находится в ячейке, в которой есть болото, он инициирует свое уничтожение
3. Маленький робот сообщает игре о том, что он уничтожен
4. Сценарий переходит к п. 5 Главного сценария

**4.6) Альтернативный сценарий «большой робот попадает в болото»**

1. Сценарий начинается после п. 11 сценария «робот перемещается на соседнюю ячейку»
2. Так как большой робот находится в ячейке, в которой есть болото, робот пропускает три следующих хода
3. Сценарий переходит к п 4.2 Главного сценария

**5) Сценарий «победа - маленький робот телепортировался»**

1. Поле сообщает игре, что “телепортирован“ робот
2. Игра считает победителем “телепортированного” робота, то есть игрока.

**5.1) Альтернативный сценарий «проигрыш - большой робот, управляемый компьютером, “поймал” маленького»**

1. Сценарий начинается с п. 2 сценария 5
2. Большой робот сообщает игре, что он “поймал” маленького робота
3. Сценарий переходит к п. 5 Главного сценария

**6) Сценарий «досрочное завершение игры»**

1. Сценарий начинается в любой точке главного сценария, когда пользователь инициирует завершение игры.
2. Игра завершается без определения победителя.

**3.3 Словарь предметной области**

**Игра** - игра знает о поле. Игра управляет игровым циклом: определяет очередного игрока, определяет окончание, определяет победителя. Игра определяет начальную расстановку игровых элементов (роботы, стены, выход, батарейки) на поле. Игра “следит” за маленьким роботом, большим роботом, точкой выхода.

**Поле** - прямоугольная область, состоящая из ячеек. Между ячейками возможно расположение стен. По границе поля могут располагаться стены. Позволяет получить роботов, находящихся на поле, и роботов, “телепортированных” с поля.

**Ячейка** - квадратная область поля. Знает о четырёх соседних ячейках и граничащих с ней стен. Одновременно может содержать в себе только одного робота и только одну батарейку.

**Стена** - непроходимое препятствие для робота, располагающееся между ячейками.

**Лабиринт** - умеет создавать горизонтальные или вертикальные последовательности стен.

**Болото** - препятствие произвольного размера, располагающееся на ячейках поля. При попадании большого робота в болото, он пропускает три хода. Маленькому роботу заходить в болото нельзя.

**Точка выхода** - разновидность ячейки. На поле может быть только одна ячейка, являющаяся выходом. “Телепортирует” робота, находящегося на ней, с поля внутрь себя. Помнит всех “телепортированных роботов”.

**Робот** - умеет перемещаться в соседние ячейки. Не может переходить через стены.

**Маленький робот** - разновидность робота, управляемого игроком. Имеет все свойства робота. Может быть “телепортированным”, поместившись в “точку выхода”. Условие победы - достигнуть точки выхода.

**Большой робот** - разновидность робота, управляемого компьютером. Имеет все свойства робота. Действует рационально при приближенности к маленькому роботу не менее, чем на 2 клетки (идет оптимальным маршрутом), в противном случае передвигается только влево или вправо, в сторону маленького робота. Если находится вдалеке от маленького робота по оси Y, но на одной оси X, пропускает ход. Условие победы - уничтожить маленького робота.

**Активный робот** - робот, который может совершать действие в текущий ход.

**Игрок** - пользователь игры, управляющий маленьким роботом при помощи клавиатуры.

**Компьютер** - виртуальный оппонент игрока, управляющий большим роботом с заданным алгоритмом.

# Структура программы на уровне классов

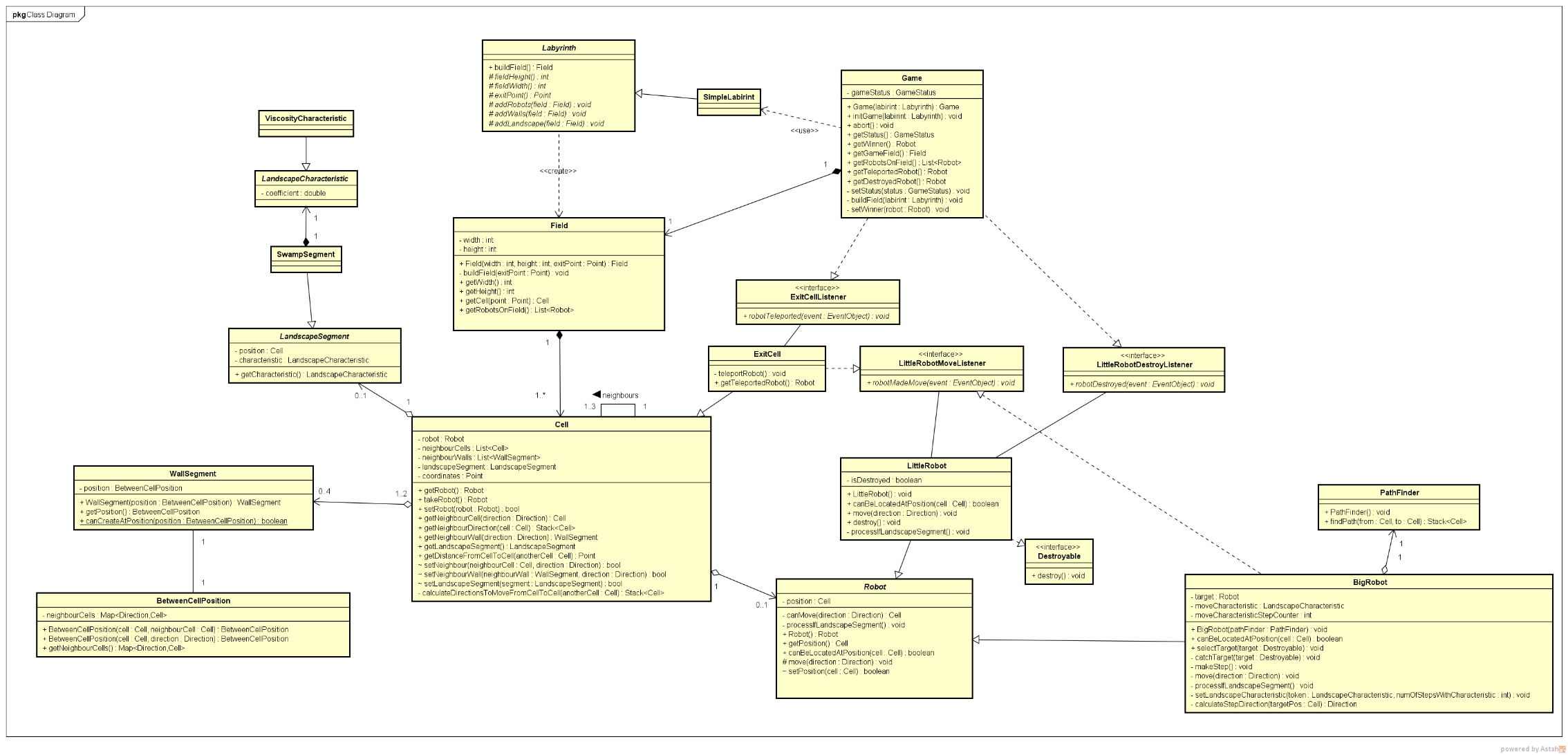
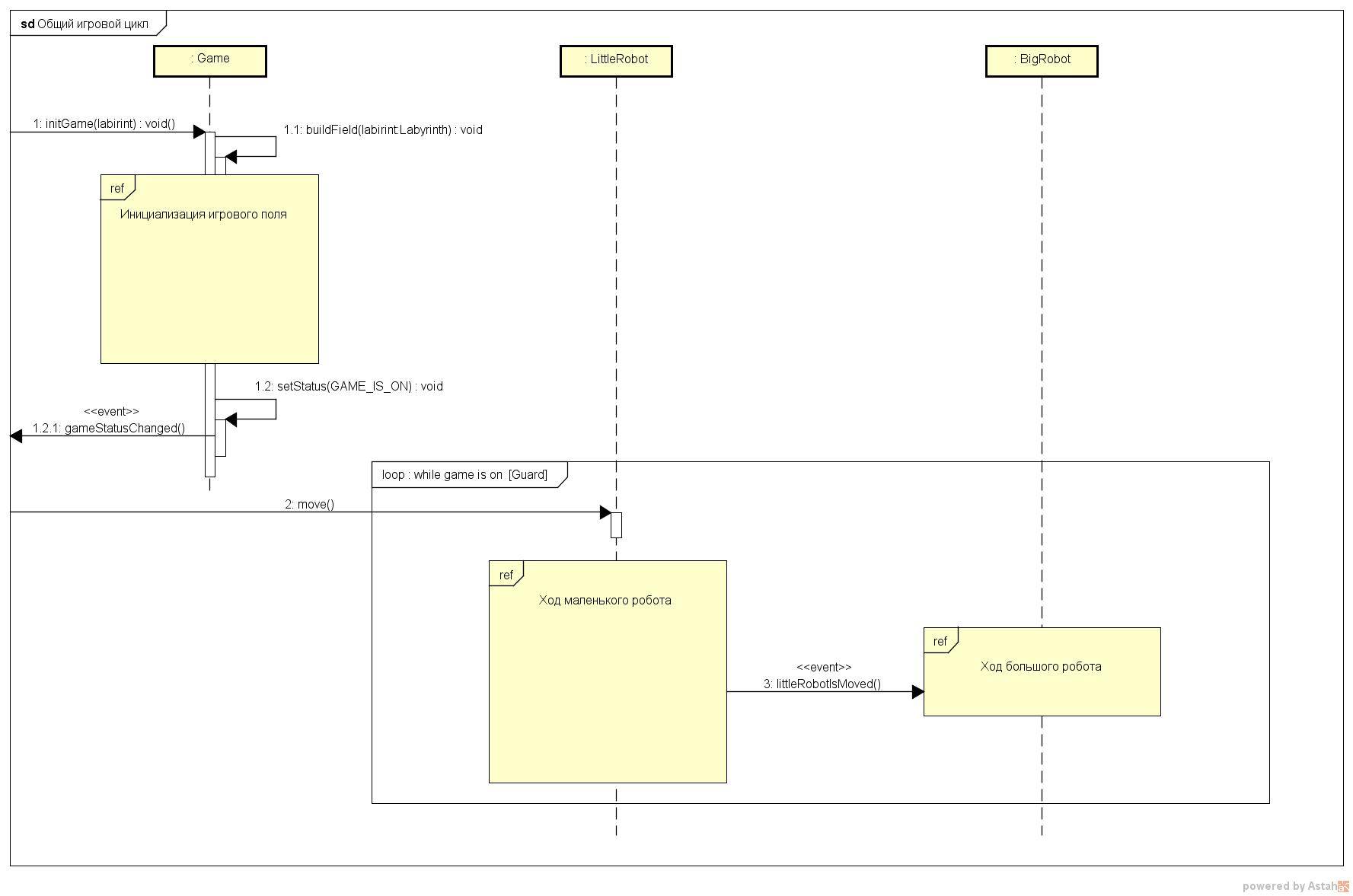
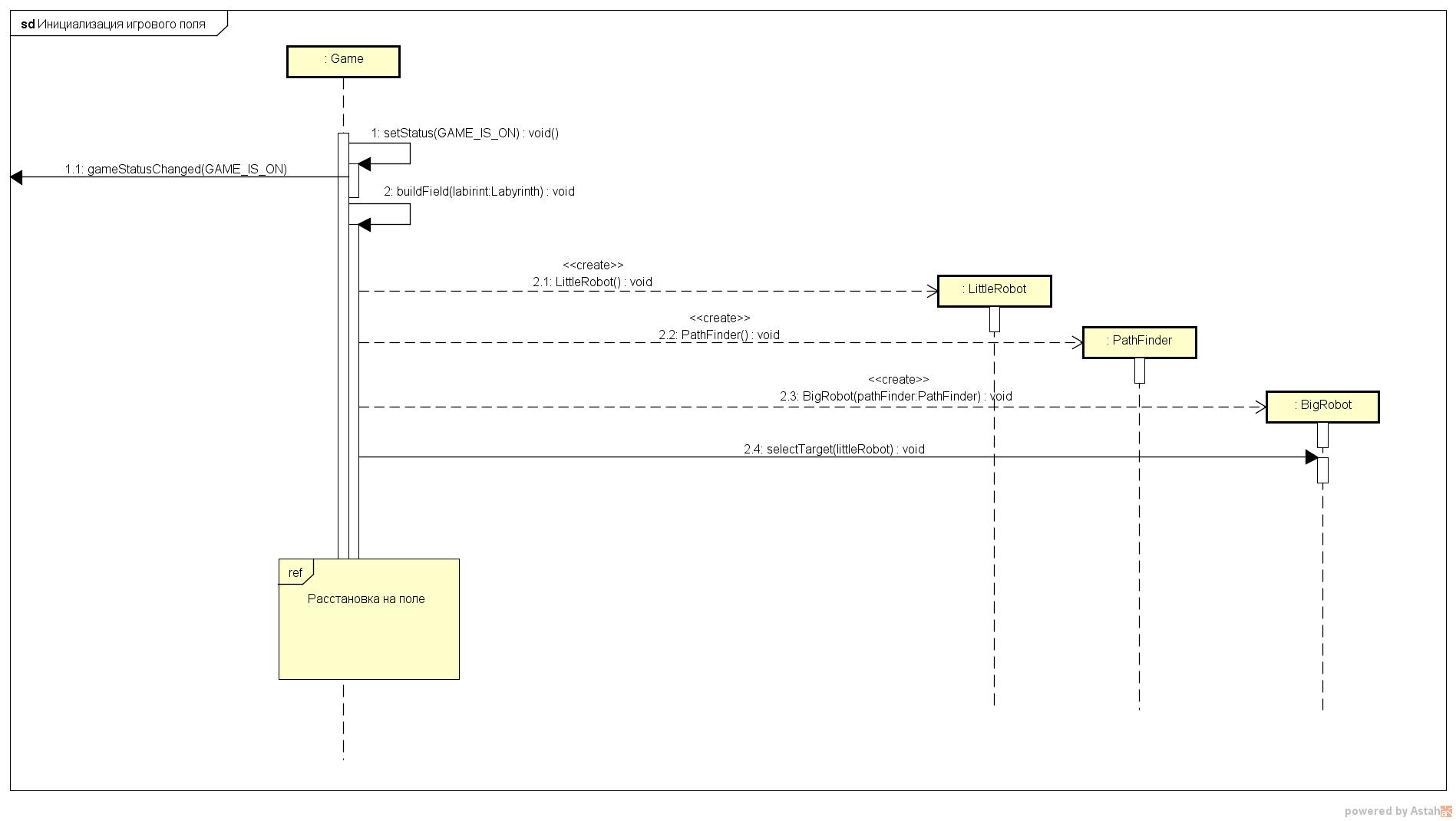


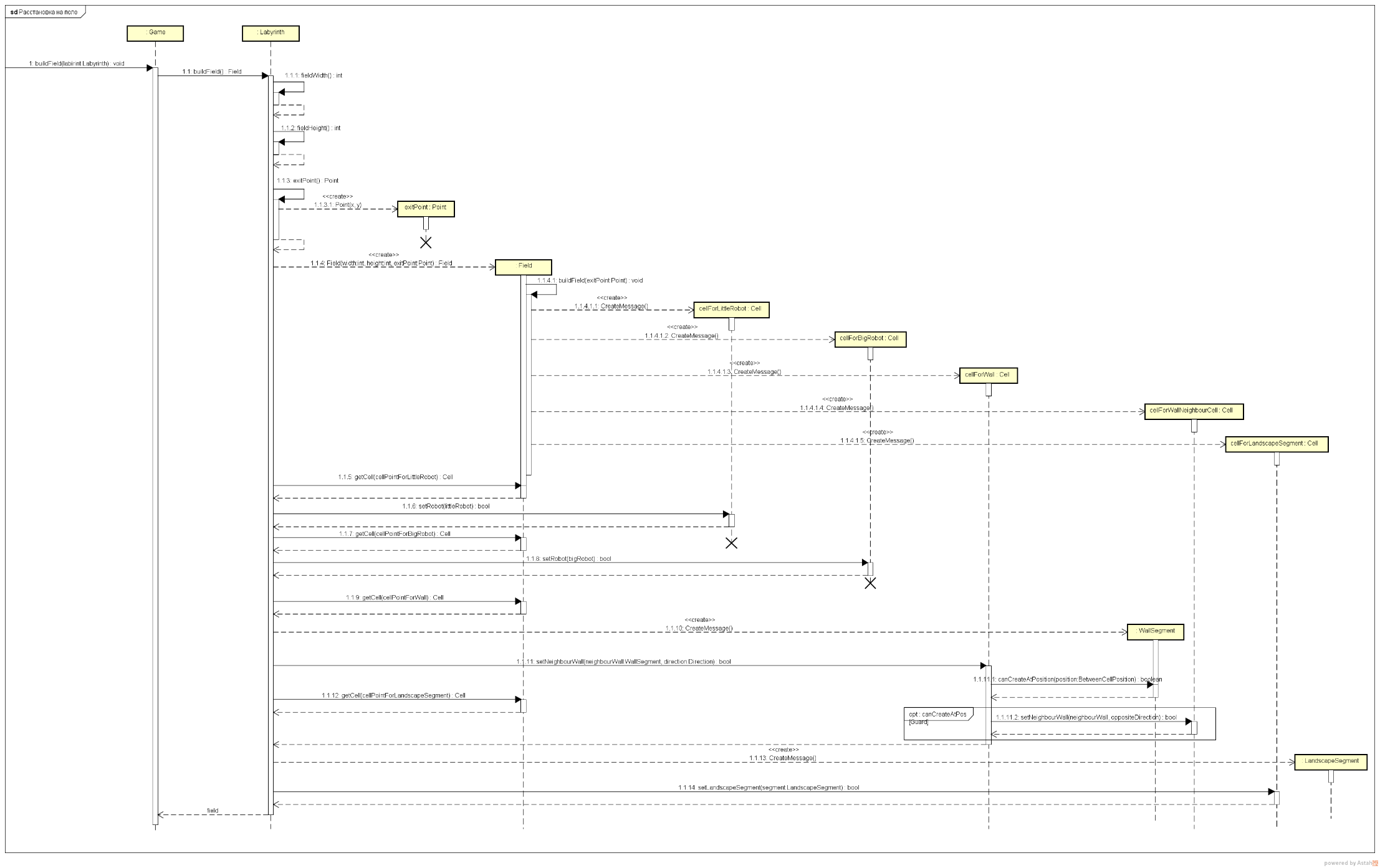
Диаграмма классов вычислительной модели



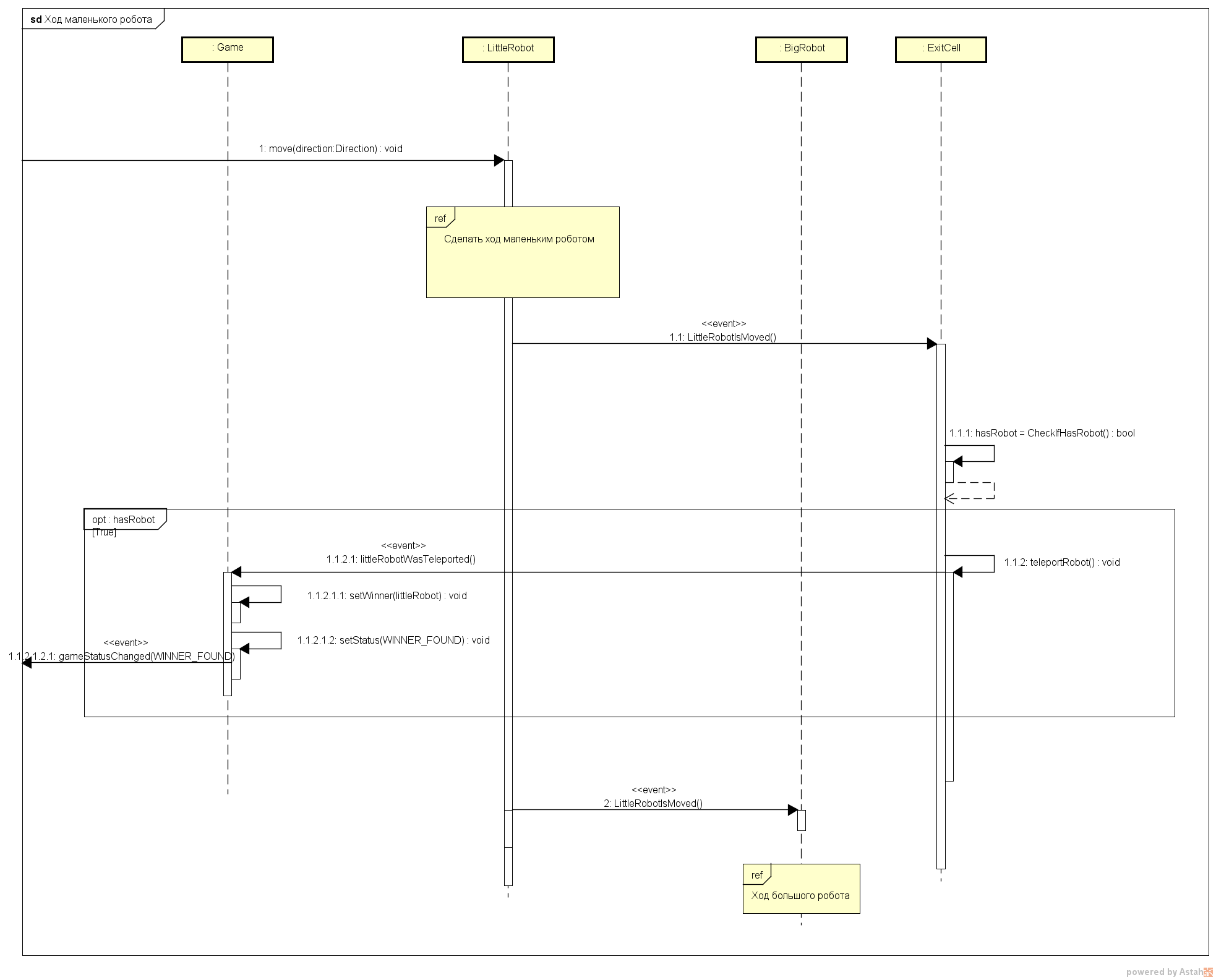
Общий игровой цикл



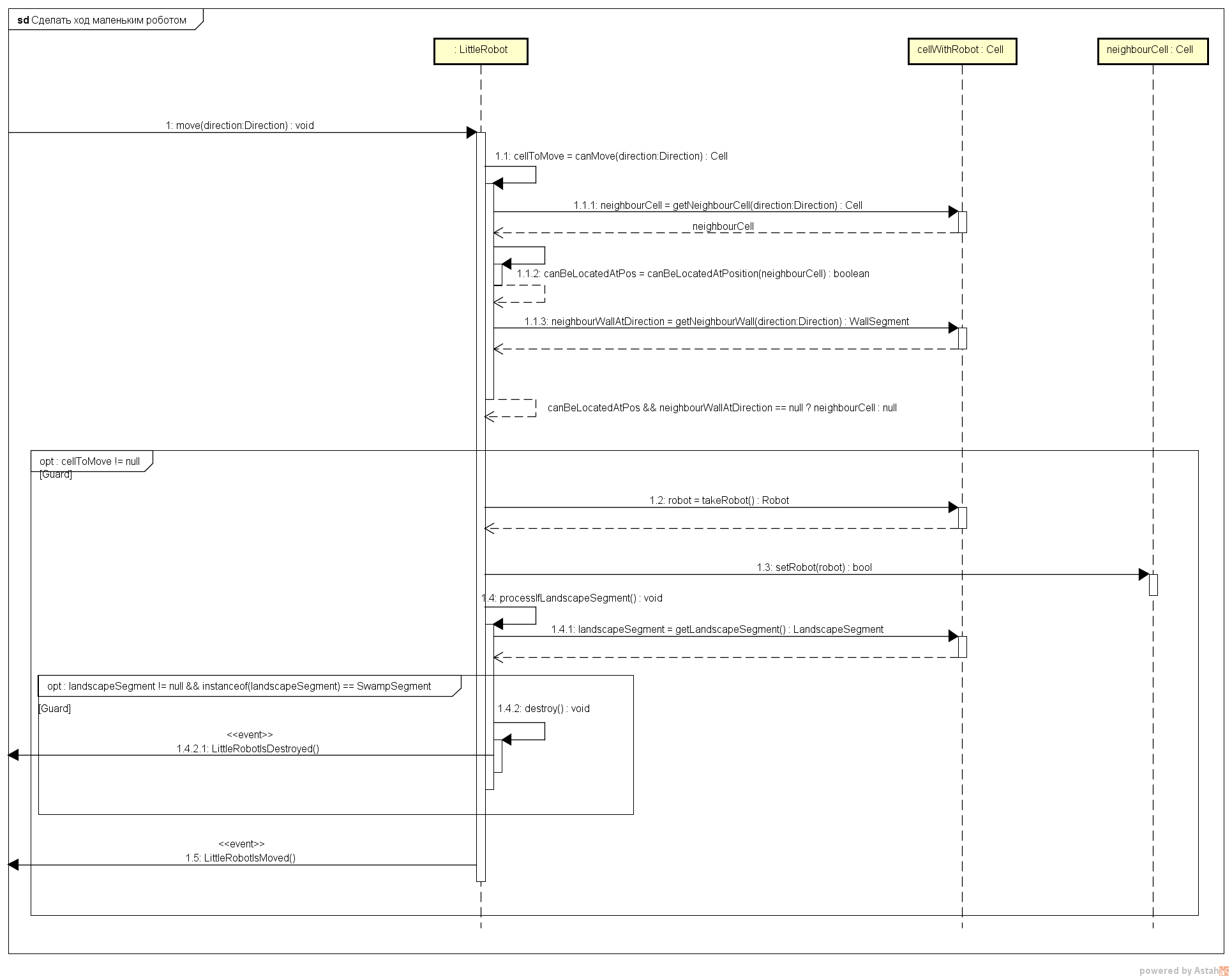
Инициализация игрового поля



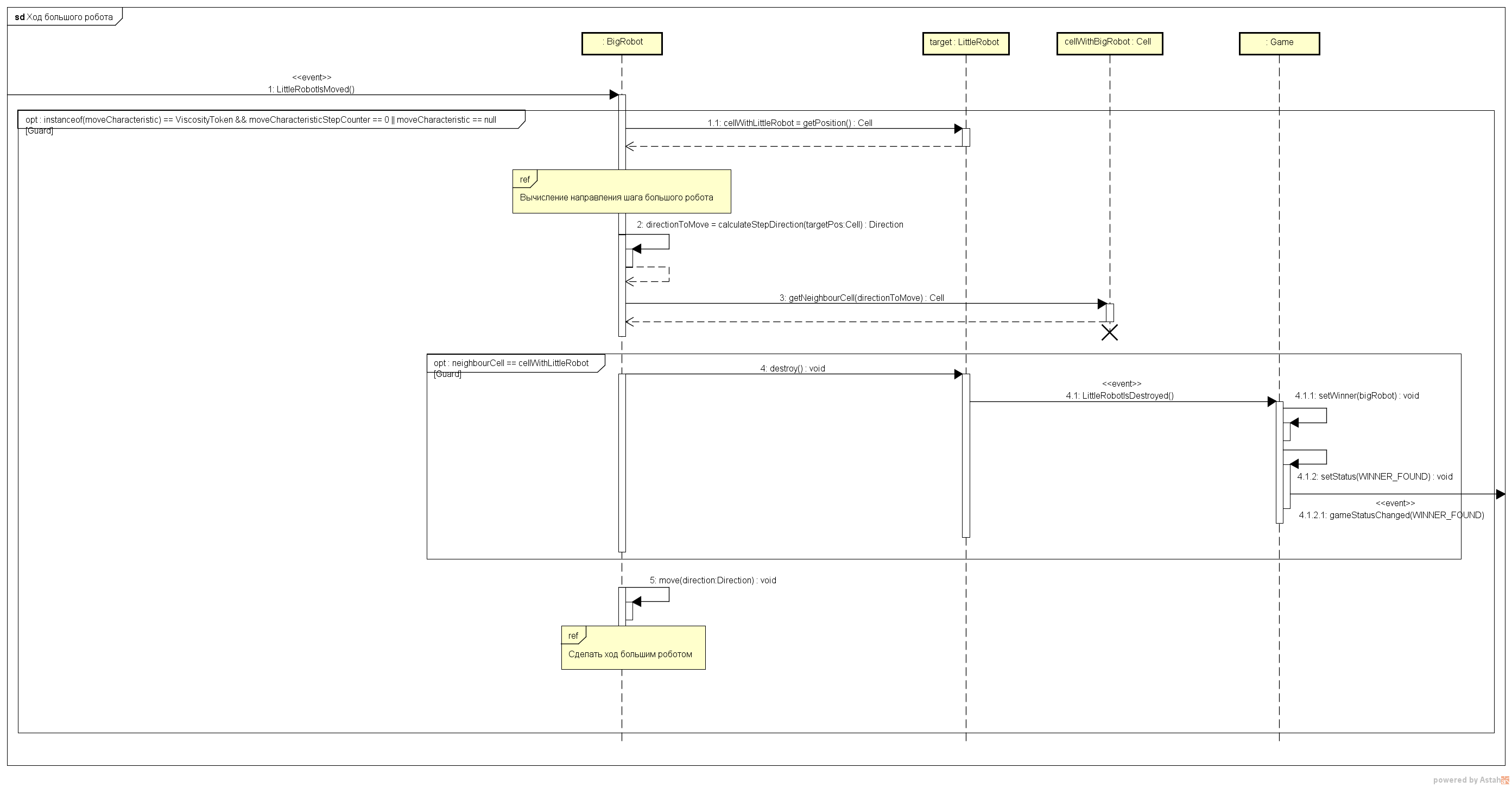
Расстановка на поле



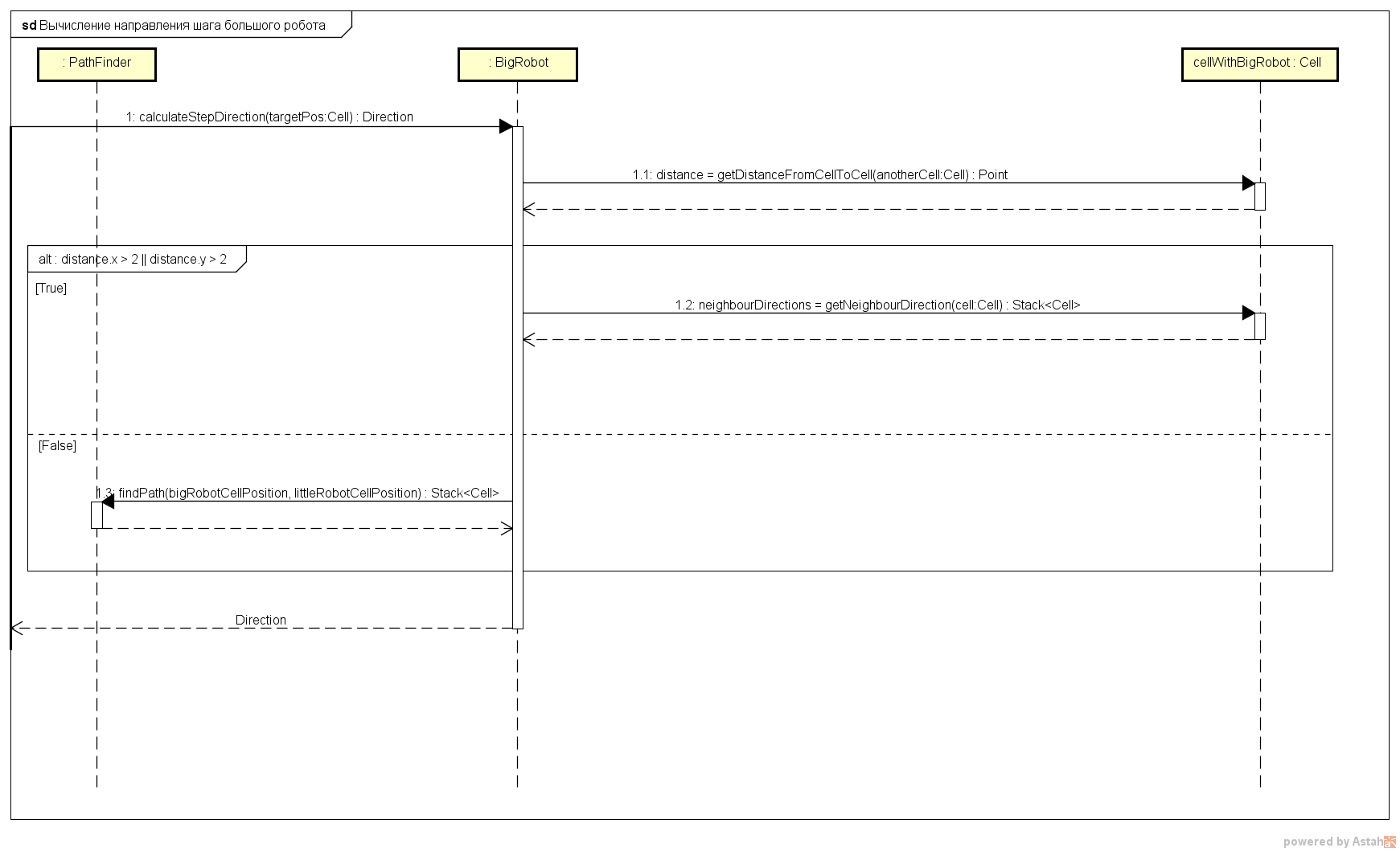
Ход маленького робота



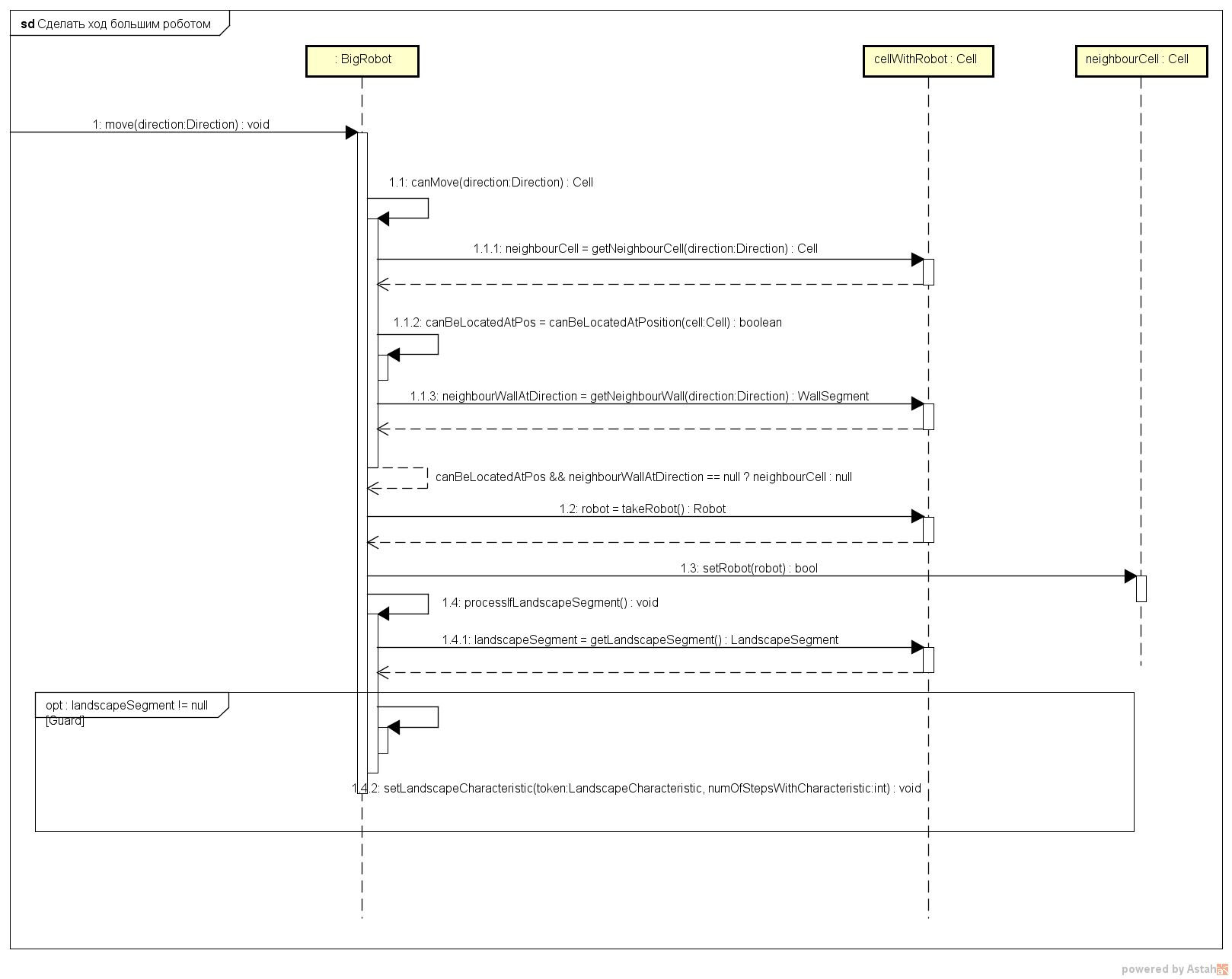
Сделать ход маленьким роботом



Ход большого робота



Вычисление направления шага большого робота



Сделать ход большим роботом