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 №27, подвариант №2)

Студент: Кременецкая А.Д.

Группа: ПрИн-367

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 2020 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 \_\_\_\_\_\_\_\_\_\_ Орлова Ю.А.

**Задание**

**на курсовой проект**

Студент: Кременецкая А.Д.

Группа: ПрИн-367

1. Тема: «Проектирование и реализация программы с использованием объектно-ориентированного подхода» (индивидуальное задание – вариант №0, подвариант №0)

Утверждена приказом от « » 20 г. №

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 « » 20 г.

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

Задание принял к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кременецкая А.Д.

«      »                         2020 г.

Содержание

[1 Формулировка задания](#_Toc440968681) 4

[2 Требования к программе](#_Toc440968682) 5

[3 Аналитическая модель](#_Toc440968683) 7

4 Проектная модель 11

[5 Паспорта классов 12](#_Toc440968684)

[6 Список использованной литературы и других источников 21](#_Toc440968685)

# 1 Формулировка задания

Правила игры «Запутанная схема».

На поле находится клубок состоящий из определенного количества узлов, соединенных нитями в хаотичном порядке. Цель игры – распутать клубок. Клубок можно распутать передвигая узлы таким образом, чтобы нити не пересекались друг с другом.

**Пример реализации:**

http://obigre.ru/logical/flashgamesyvf.html

**Подвариант 2**: необходимо предусмотреть в программе **точки расширения**, используя которые можно реализовать  вариативную часть программы (в дополнение к базовой функциональности).

**Вариативность**: предусмотреть появление ниток с различными свойствами растяжения и возможностью их порвать. Нитки должны быть визуально различимы, когда они готовы порваться. Порванная нить приводит к проигрышу.

**НЕ изменяя** ранее созданные классы, а используя **точки расширения**, **реализовать**:

* нитки, которые растягиваются не более чем на N процентов от своей исходной длины, но они не рвутся
* нитки, которые рвутся, при значительном растягивании

# 2 Требования к программе

**2.1 Функциональные требования**

Игра имеет следующие функциональные требования.

* Выбор уровня игры.
* Переигровка.
* Инициализация поля.
* Перетаскивание вершины игроком.
* Определение пересечения нитей.
* Определение победы.
* Ограничение максимальной длины нити при растягивании, если нить имеет данное ограничение.
* Разрыв нити при значительном растягивании, если нить рвущаяся.
* Завершение игры.

Примеры макетов окна игры "Запутанная схема"

При запуске игры появляется окно с меню



Рис. 1 - Главное меню

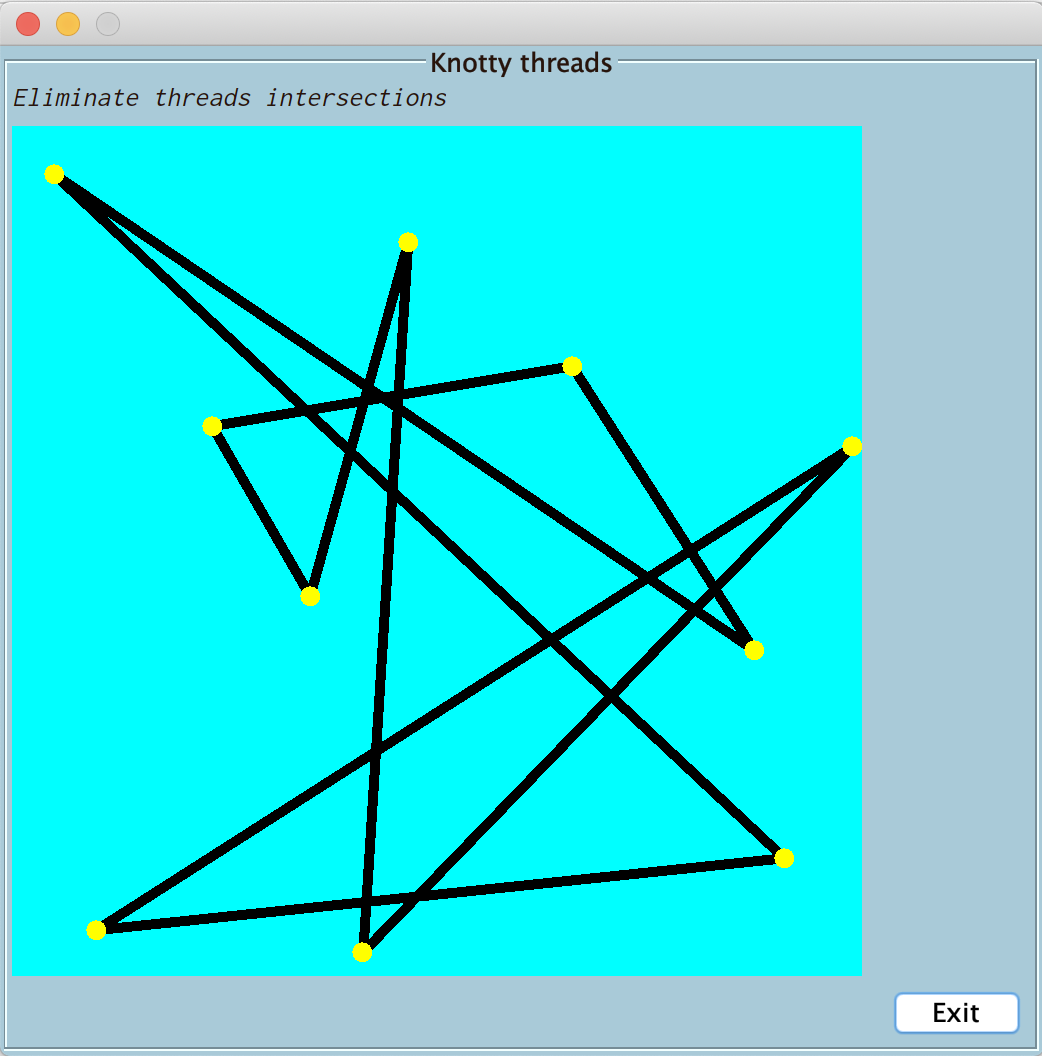


Рис. 2 - Общий вид главного окна программы

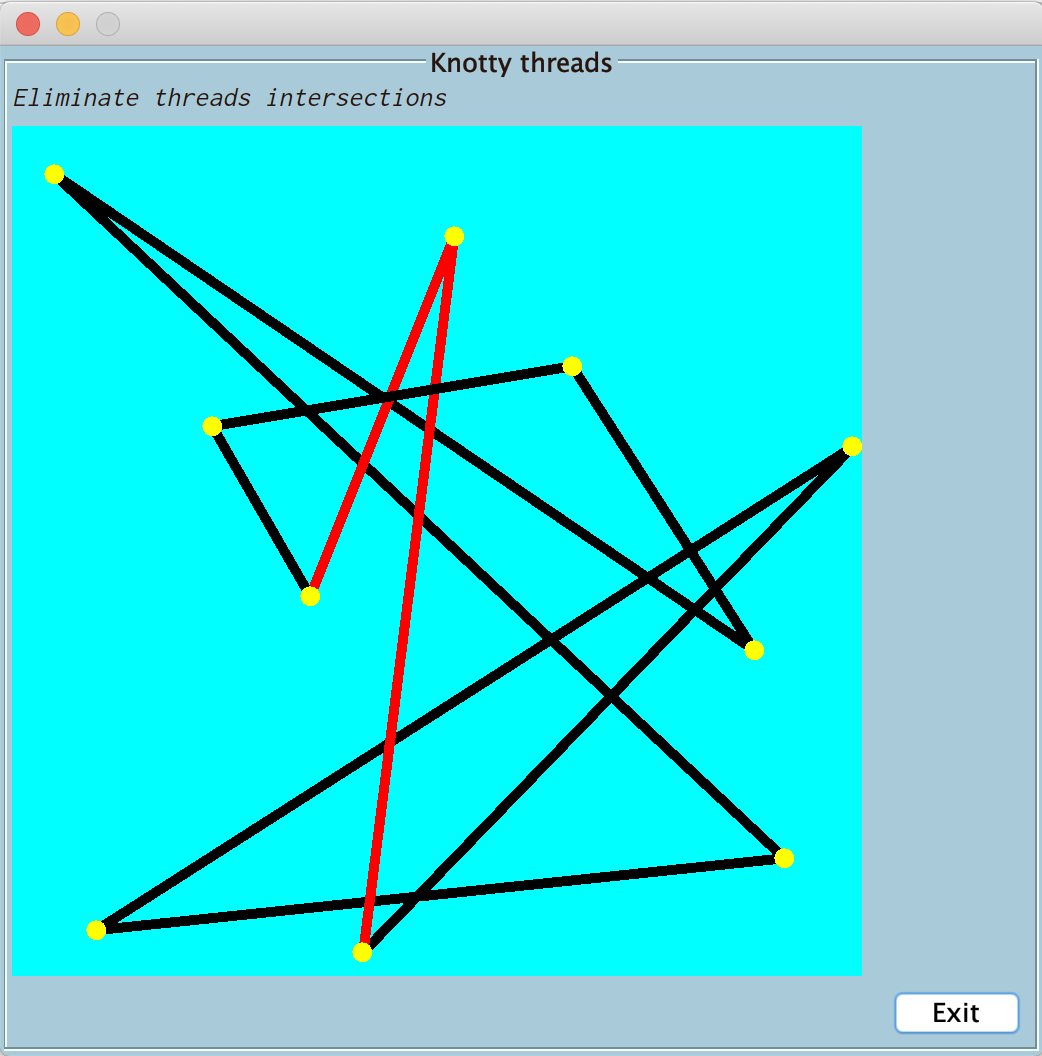


Рис. 3 - Вид схемы при максимальном натяжении рвущейся нити

При успешном прохождении игры появляется сообщение о выигрыше

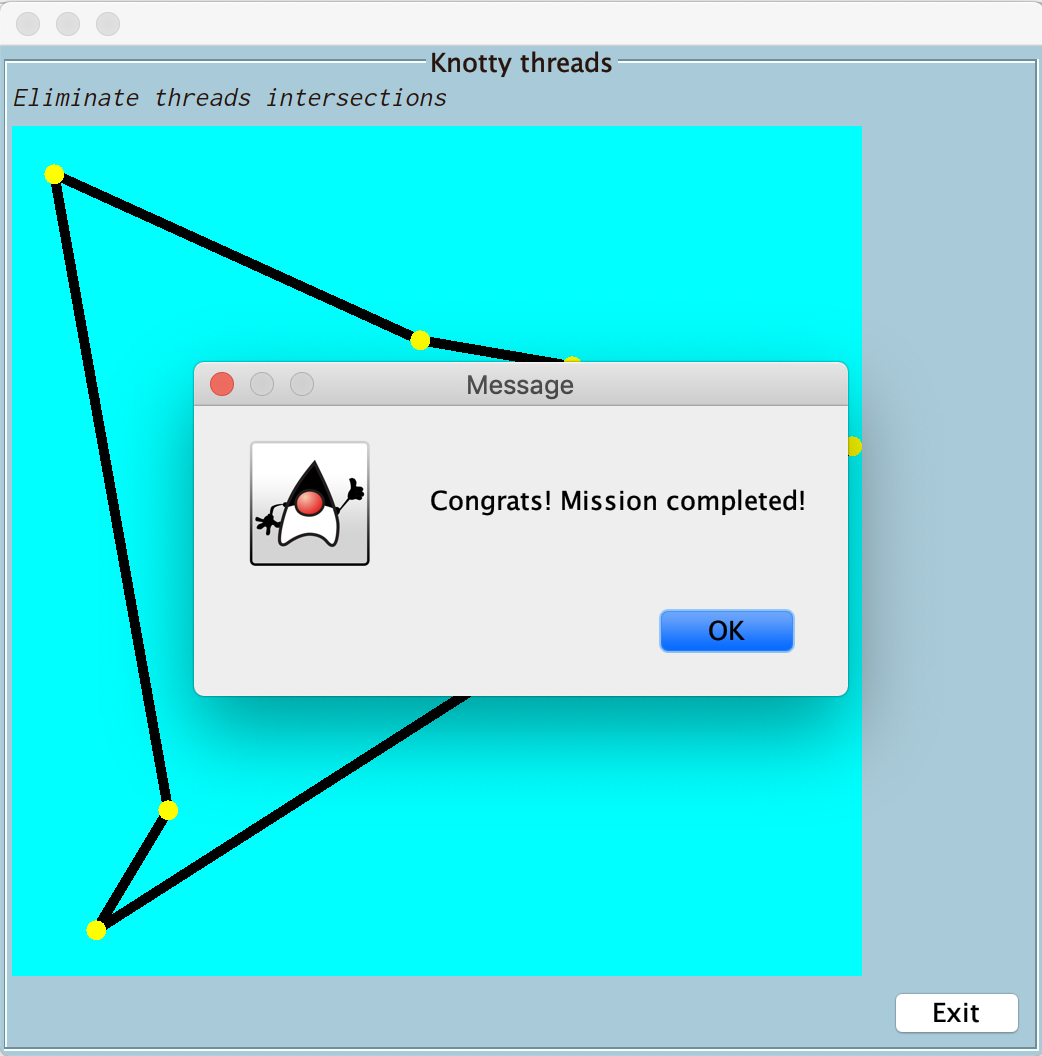


Рис. 4 - Вид главного окна при выигрыше

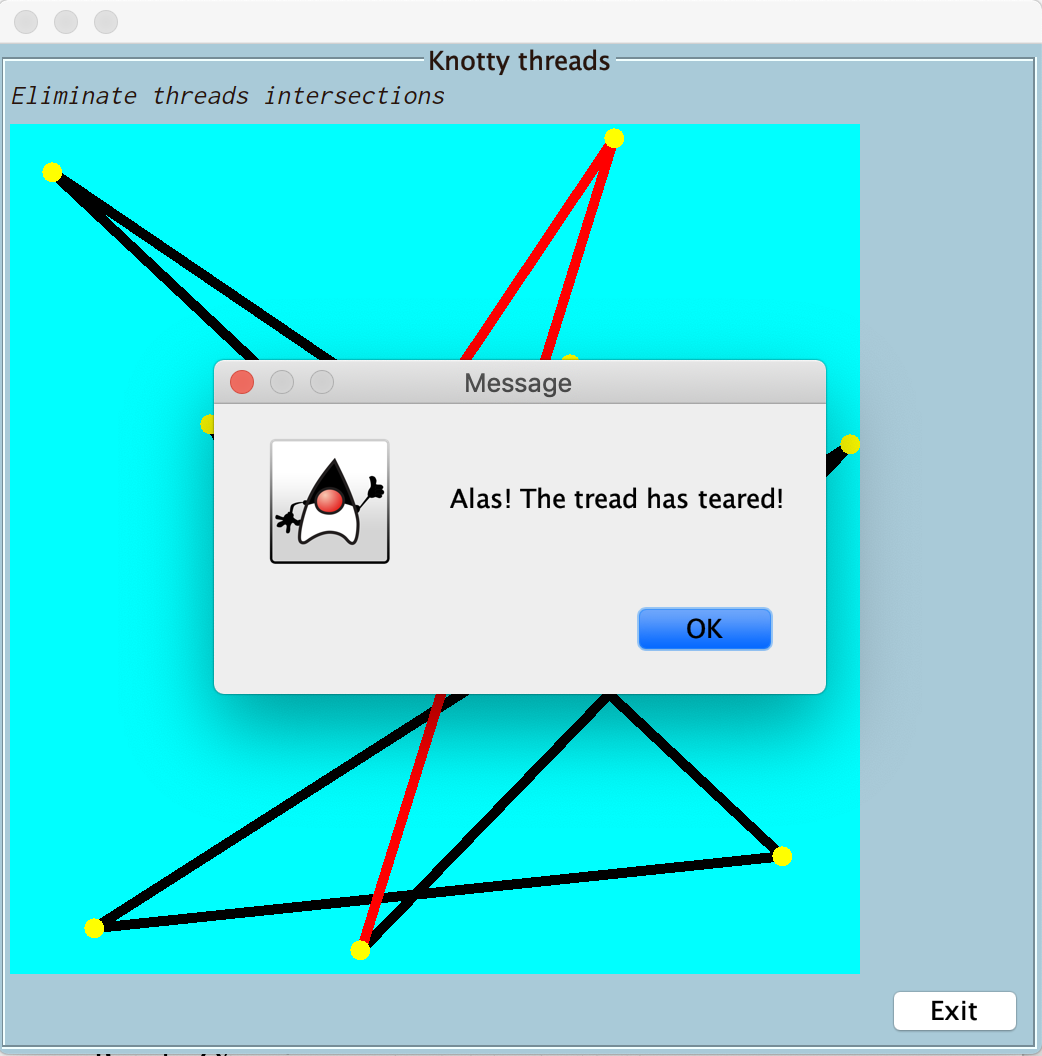


Рис. 5 - Вид главного окна при проигрыше

**Сценарии использования**

Согласно разработанной диаграмме вариантов использования был созданы сценарии для прецедента «Игрок победил» и «Игрок проиграл».

**Вариант использования «Игрок победил»:**

**Краткое описание**

Данный вариант использования происходит в случае, если игрок распутал нитки, при этом не порвав ни одной из них.

**Основной поток событий**

1. Начинается сеанс игры по указанию игрока
2. Игра генерирует поле со схемой, содержащей узлы, связанные пересекающимися нитями, при этом каждый узел связан с двумя и более нитями
3. Делать

3.1. Игрок передвигает узел в пределах игрового поля

3.2. Если при передвижении узла, нить достигла своей максимальной длины

3.2.1. Нить окрашивается в красный цвет

3.2. Узел инициирует растяжение или сжатие, связанных с ним нитей в зависимости от своего нового положения

Пока нити в схеме пересекаются

1. Игра успешна

**Альтернативные потоки**

***2А. В процессе передвижения конца нити, нить не выдержала натяжения***

1. Система выводит сообщение о проигрыше

2. Управление переходит на шаг 1 основного потока.

***3А. Игрок закончил игру досрочно***

1. Система завершает процесс игры без вывода сообщений

***4А. Игрока не устроил результат и он решил переиграть***

Управление переходит на шаг 1 основного потока.

Инициация нового сеанса игры

Новый сеанс игры инициируется игроком посредством выбора уровня сложности из главного меню (рис. 1).

Инициация переигровки

Инициируется игроком посредством нажатия на кнопку "exit" в основном окне игры (рис. 2) и выбором уровня сложности в появившемся меню (рис.1).

**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**

**Игра** - система, генерирующая условия игры, отвечающая за игровой цикл и определяющая исход

**Игрок** - пользователь игры

**Схема** - сгенерированный игрой набор нитей, связанный узлами. Конечная цель схемы - проверить нити на наличие пересечений

**Поле** - плоское размеченное пространство, содержащее схему

**Нить** - объект, принадлежащий двум узлам. Способен проверять сам себя на наличие пересечений с другими нитями, имеет цвет

**Растяжимая нить** - нить, имеющая ограниченную длину. Способна рваться. При достижении максимальной длины меняет цвет

**Нерастяжимая нить** - нить, не имеющая понятия длины. Рваться не может

**Нить с ограниченной длиной** - нить, имеющая ограниченную длину. При растяжении на N процентов от своей исходной длины перестает растягиваться.

**Узел** - точка на поле, соединяющая две и более нитей между собой. Для победы в игре необходимо передвинуть узлы так, чтобы не осталось пересекающихся нитей

**3.2 Структура предметной области**

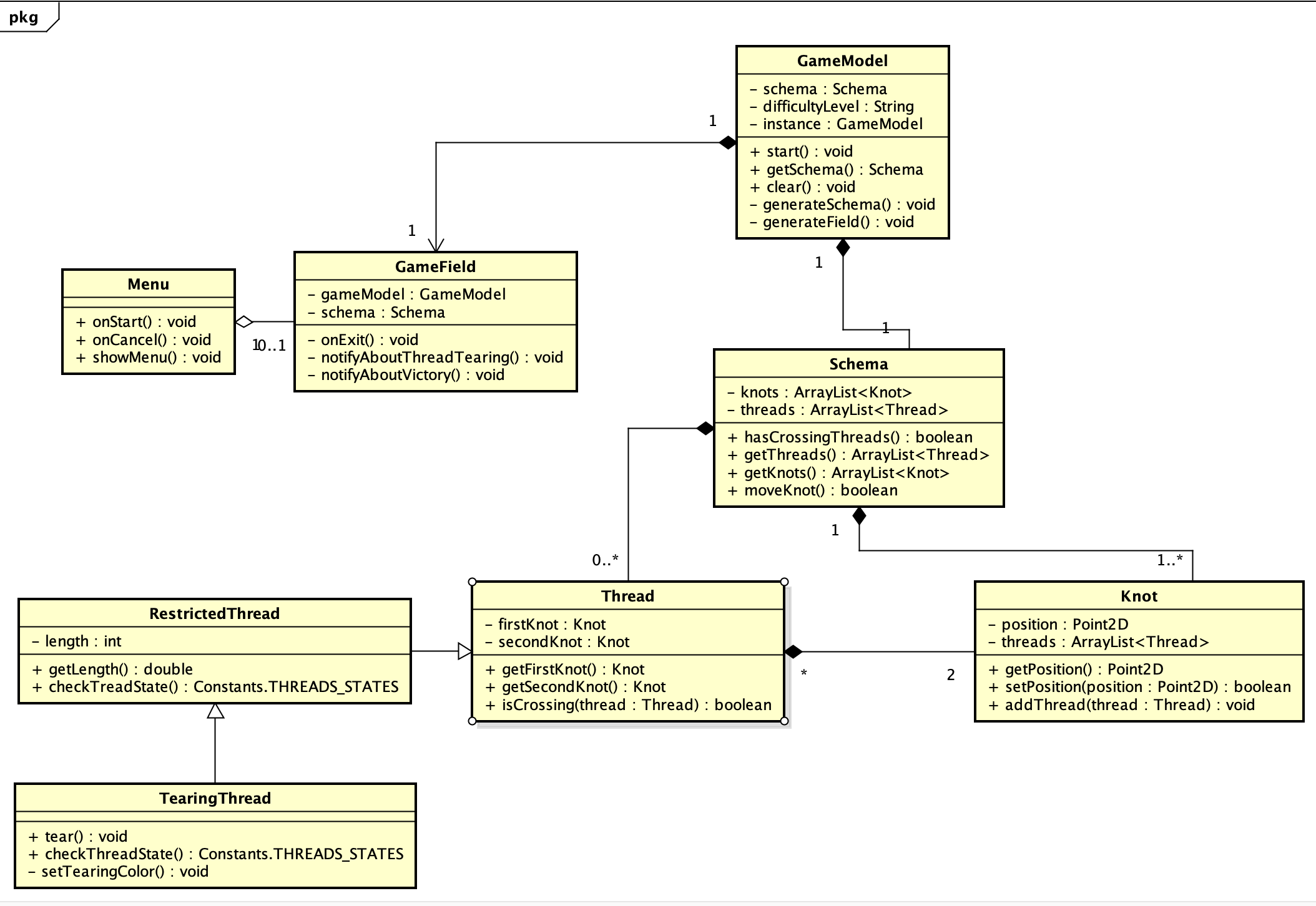
****

Диаграмма 1 «Структура предметной области»

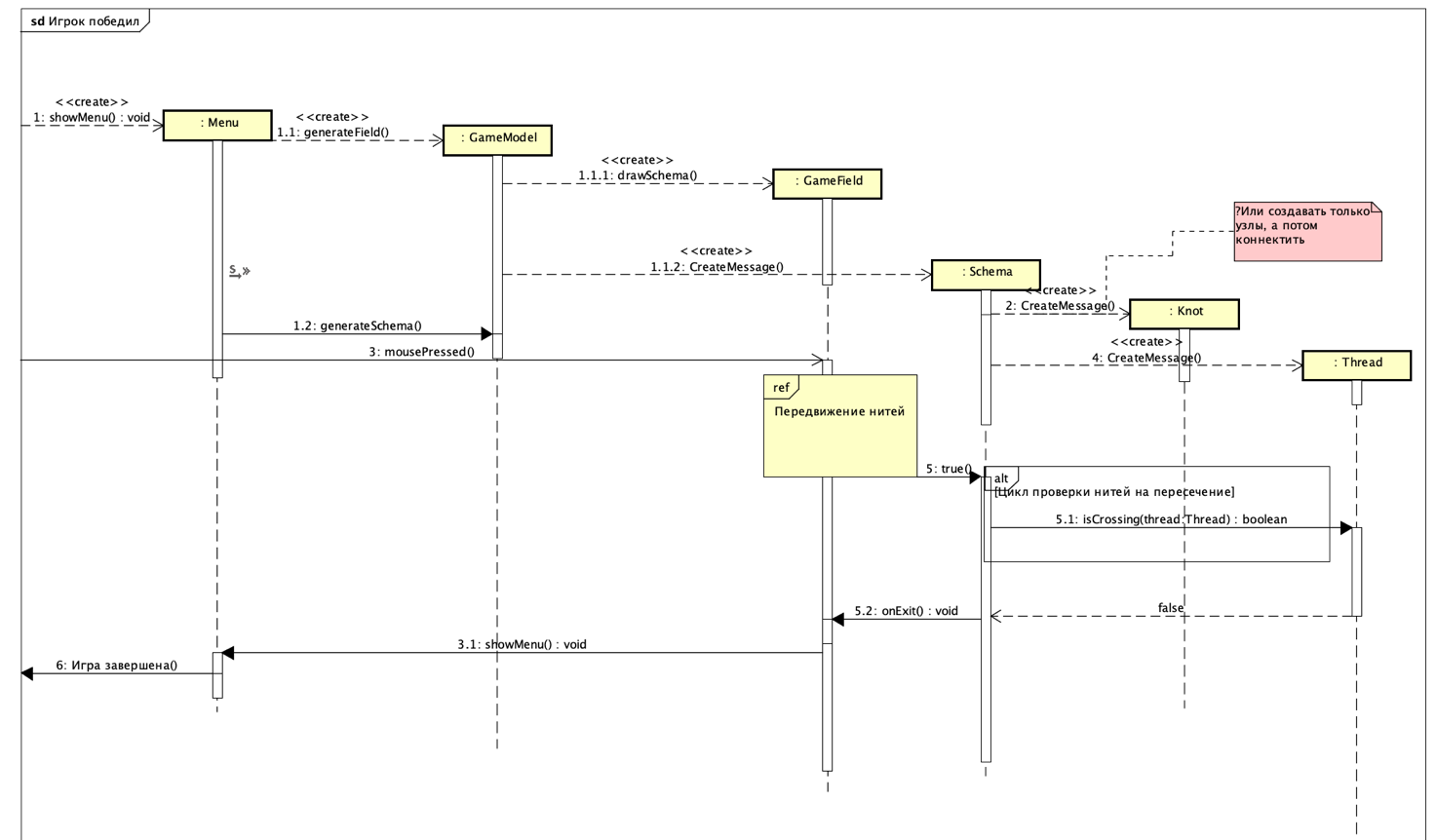
**3.3 Типовые процессы предметной области**

Диаграмма 2 «Игрок победил»

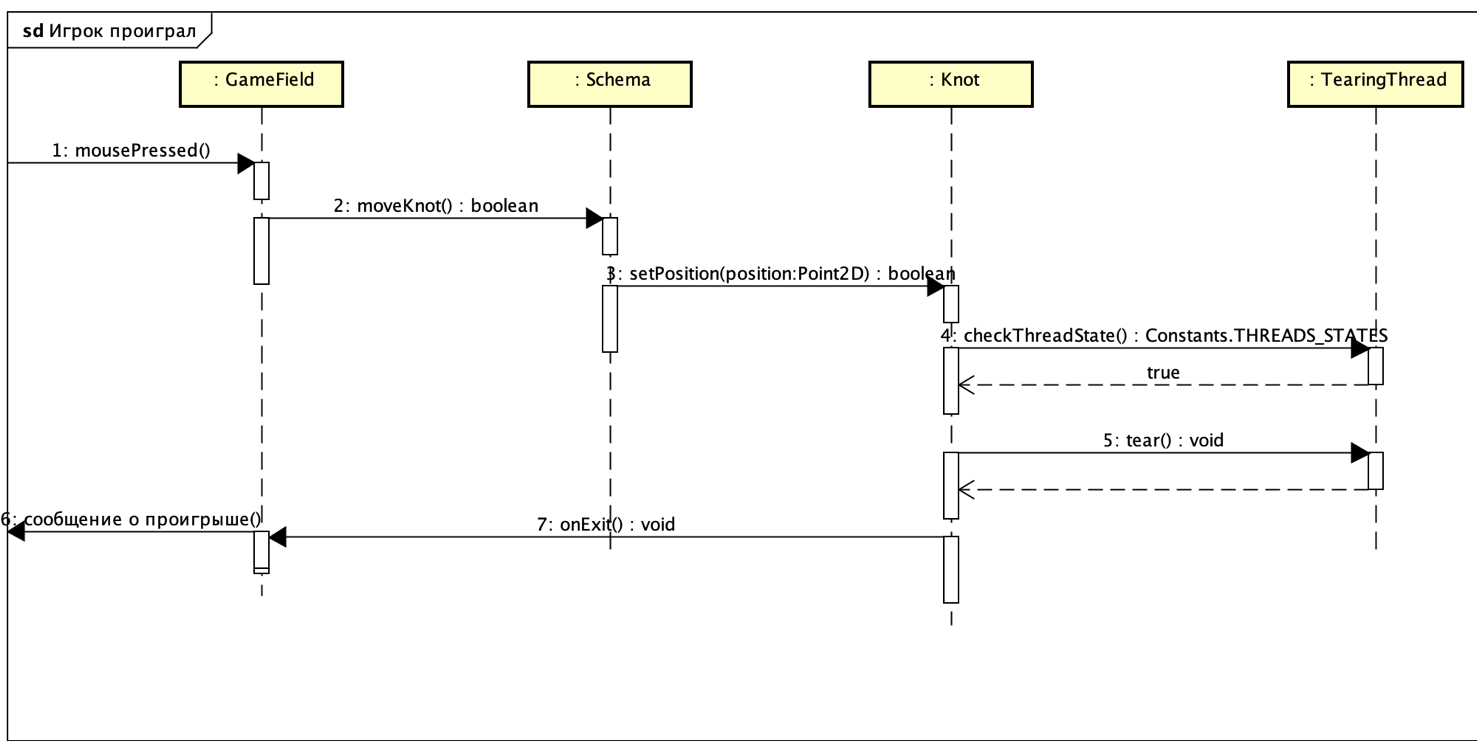
****

Диаграмма 3 «Игрок проиграл»

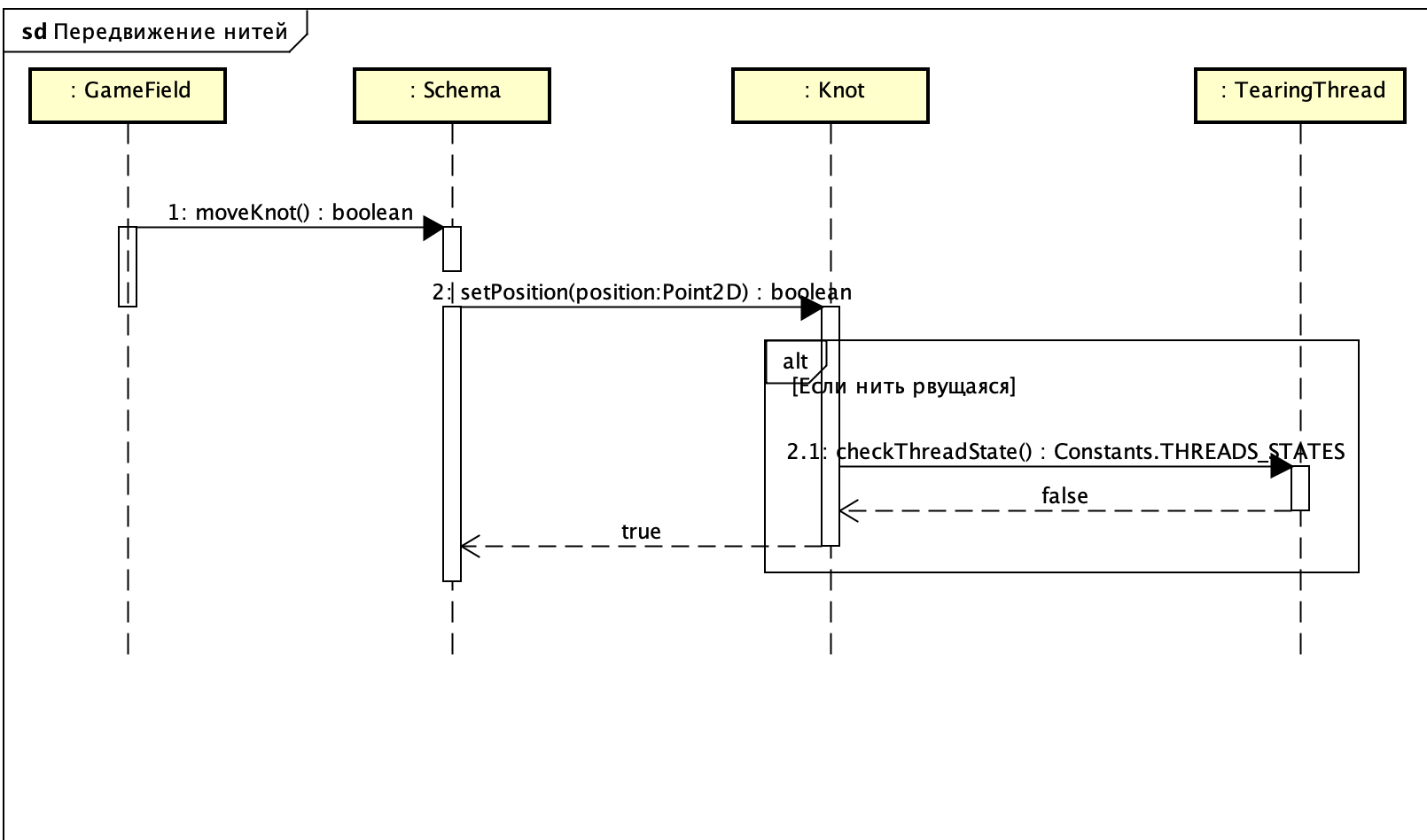


Диаграмма 4 «Передвижение нитей»

# 4 Проектная модель

**4.1 Структура программы**

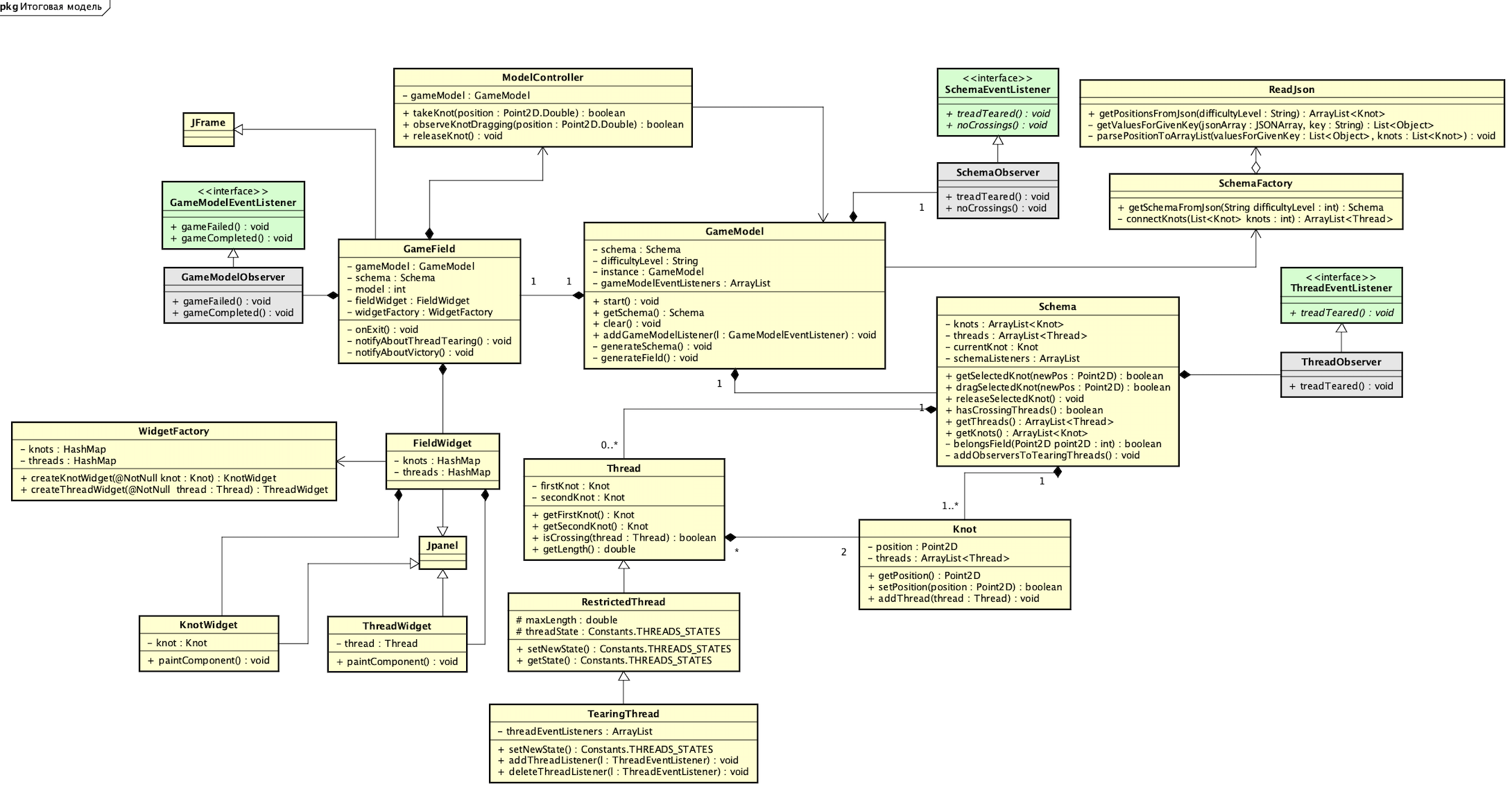
****

Диаграмма 5 «Структура программы»

**4.2 Поведение программы**

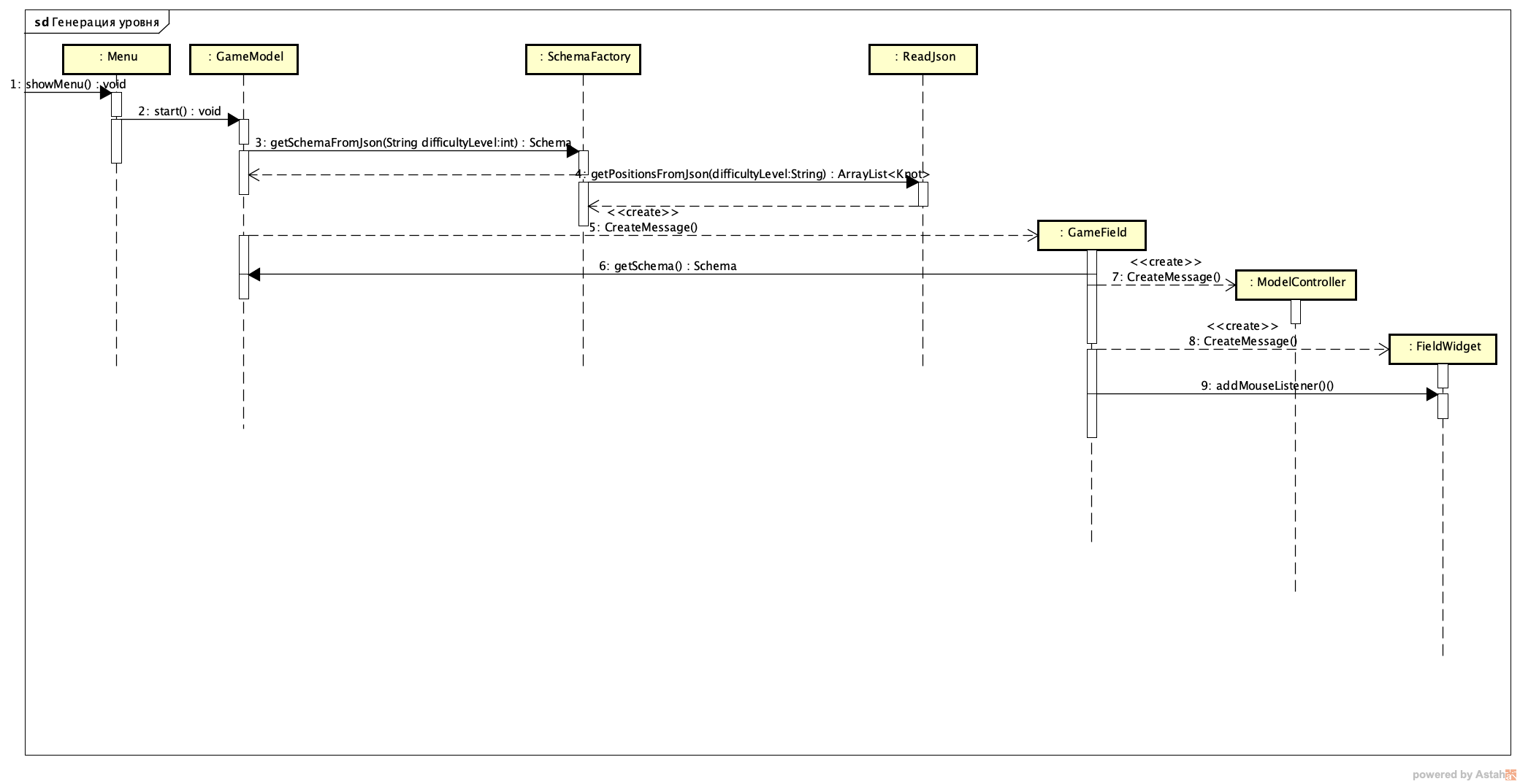
****

Диаграмма 6 «Генерация уровня»

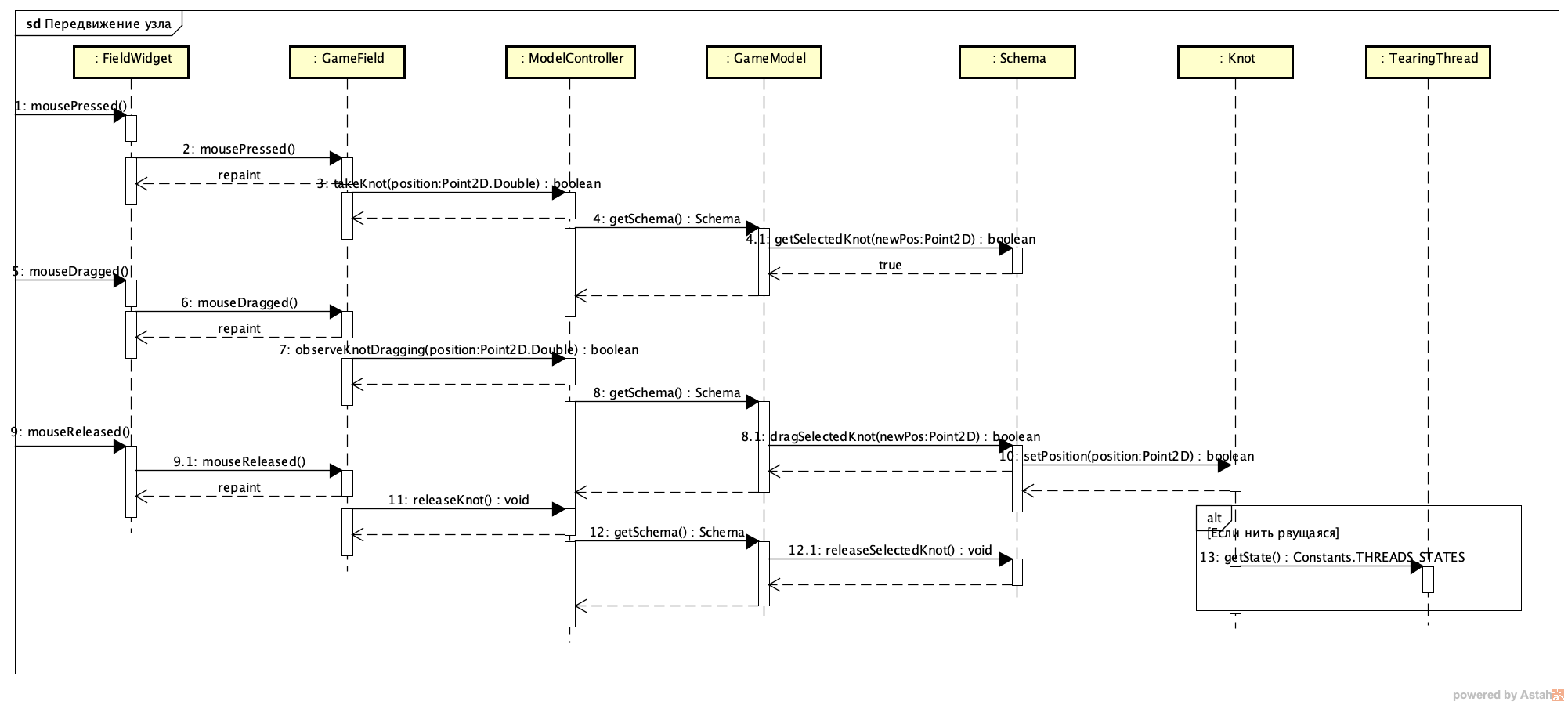


Диаграмма 7 «Передвижение узла»

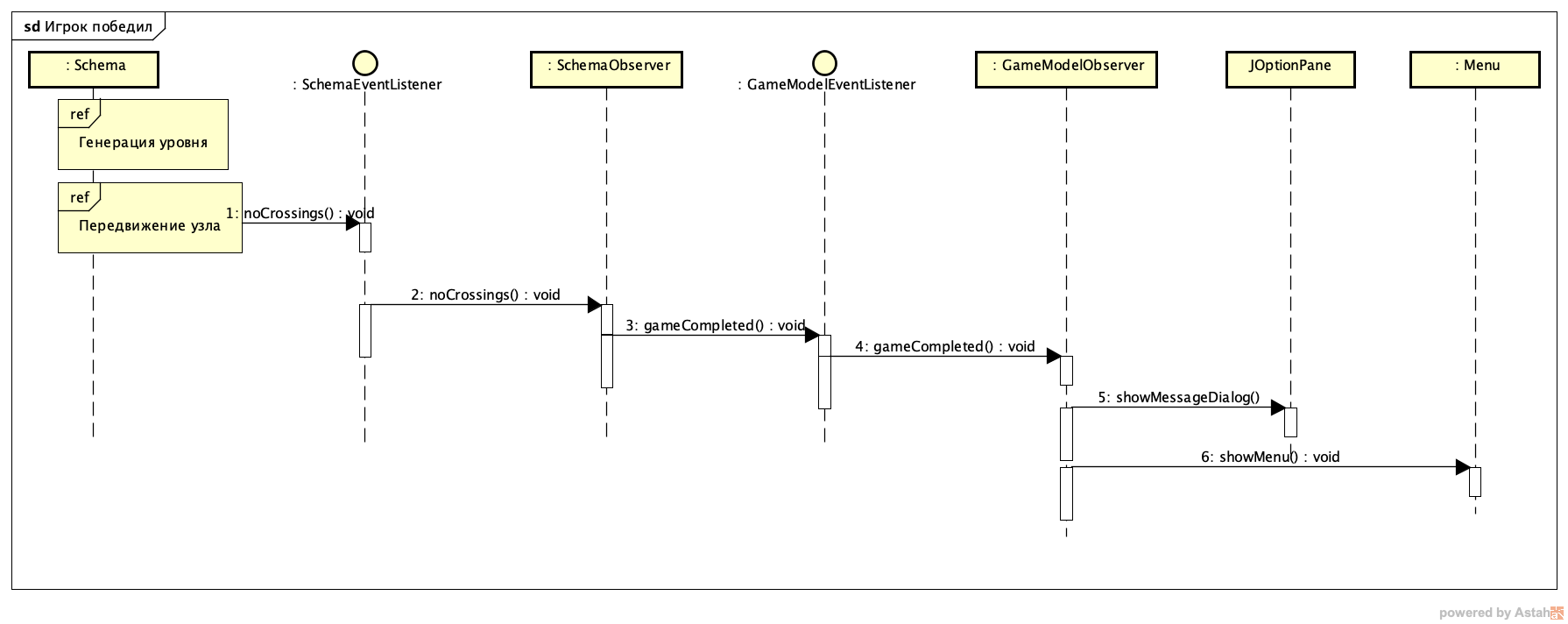


Диаграмма 8 «Игрок победил»

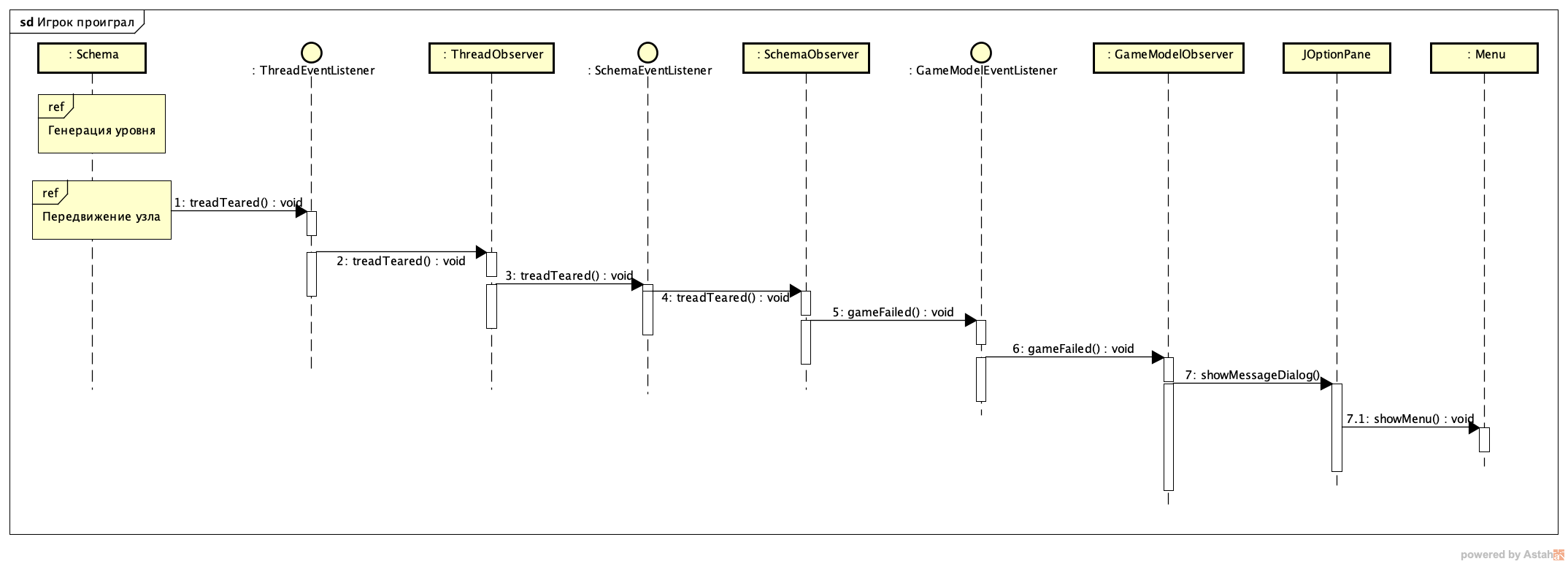


Диаграмма 9 «Игрок проиграл»

# 5 Паспорта классов

# 6 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/.