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.

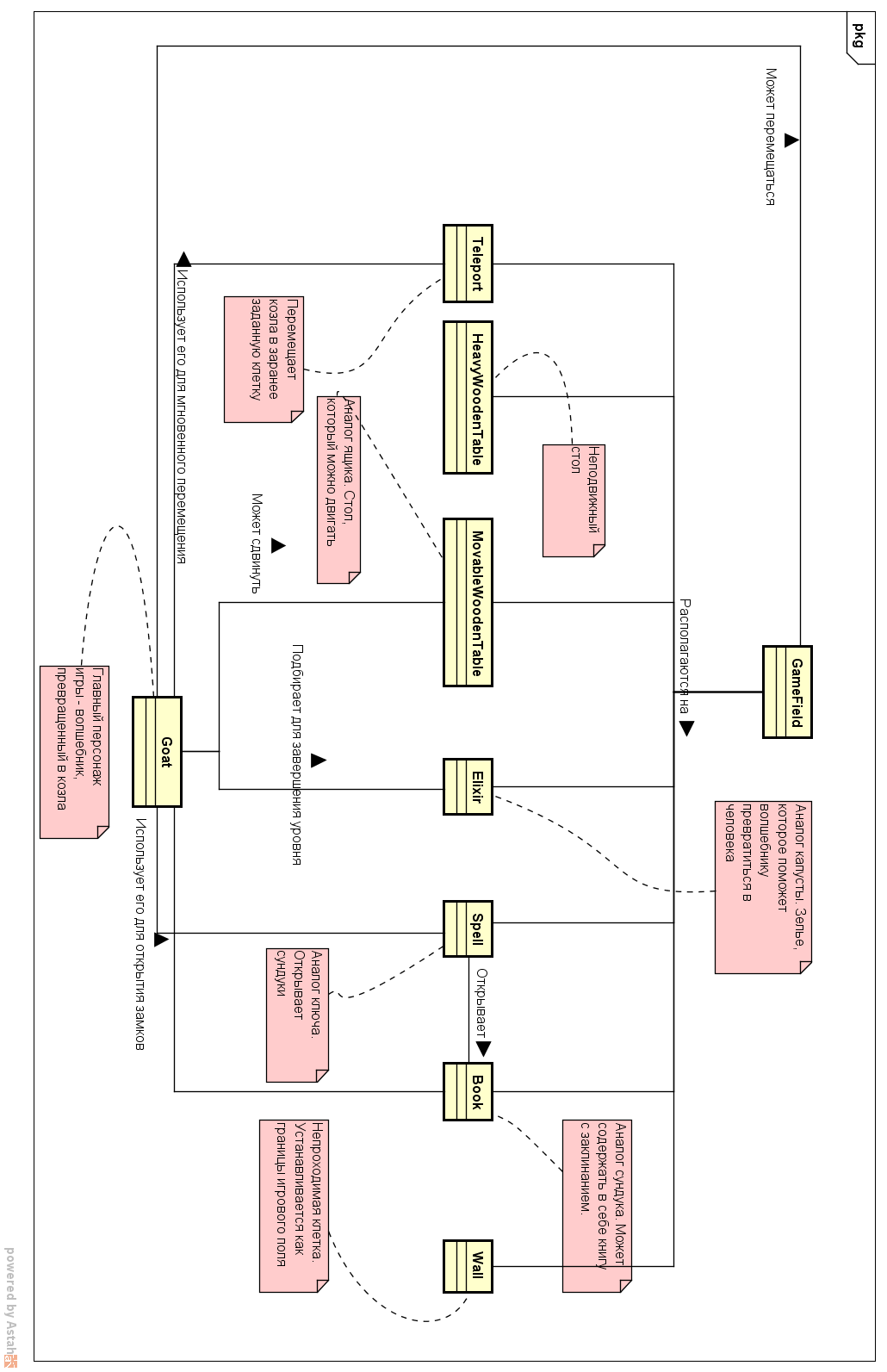
Козел – объект, который может перемещаться между соседними (четырьмя) ячейками. У козла есть определенное число ходов, по истечению которого, игра прекращается. Козел может подобрать эликсир с поля (игра завершится) или заклинание (положится в инвентарь), при этом поднятый предмет убирается с поля. Козел 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.

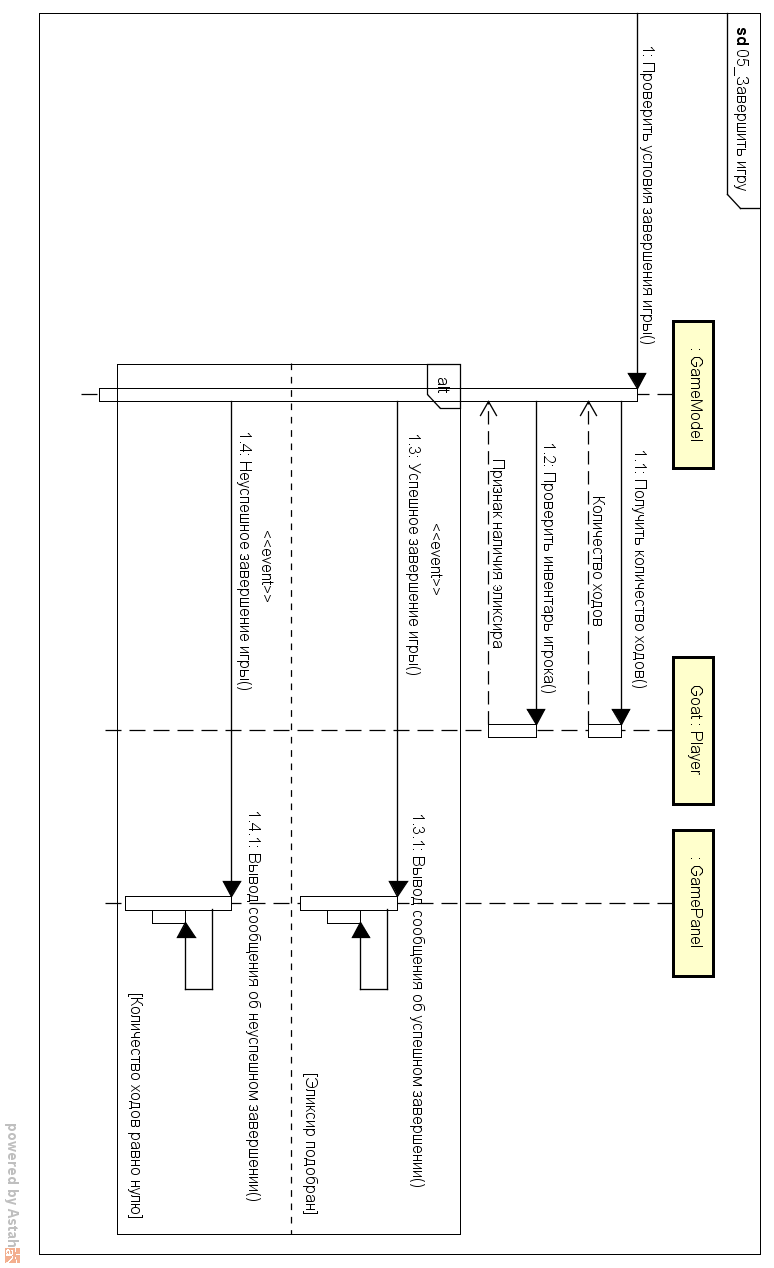
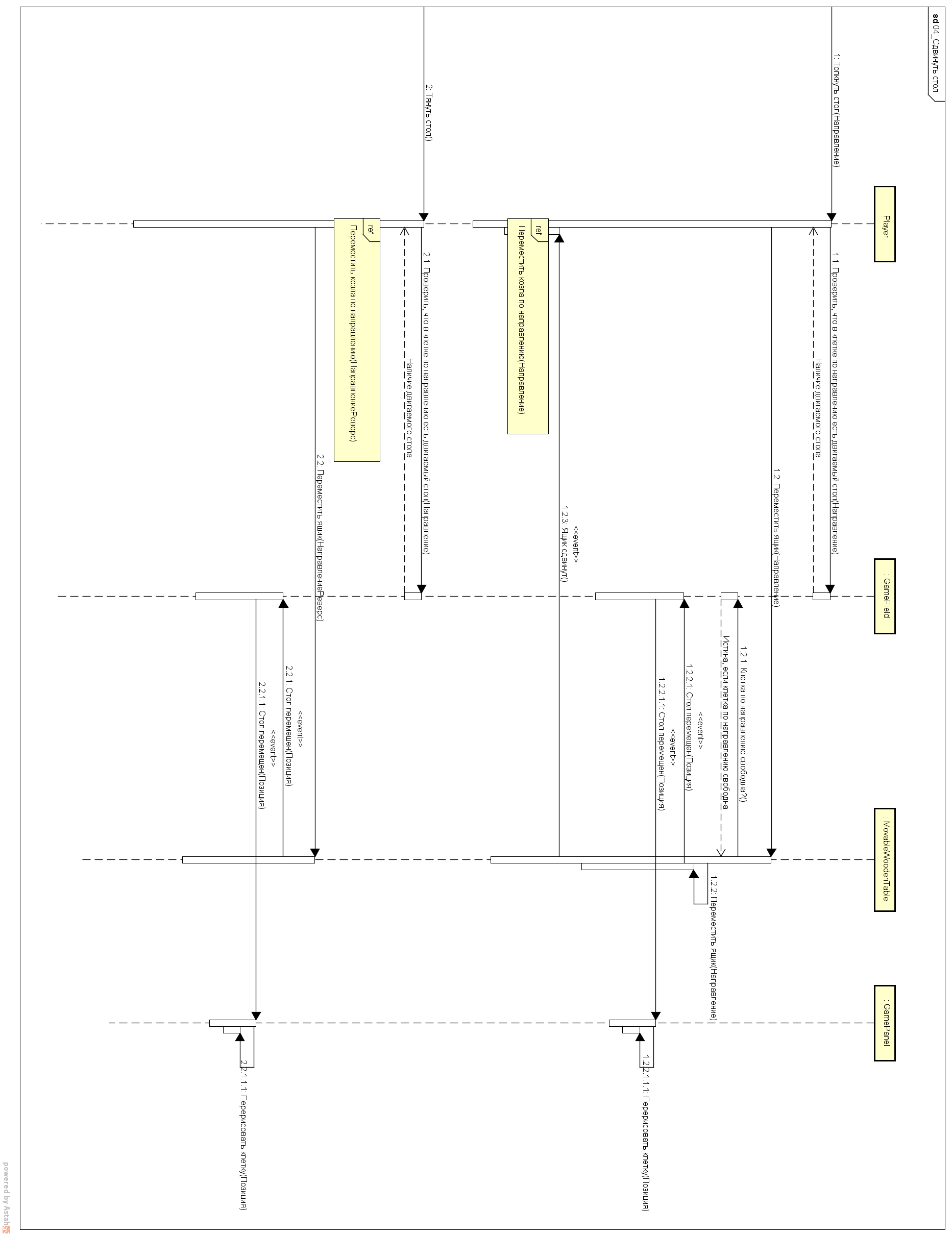
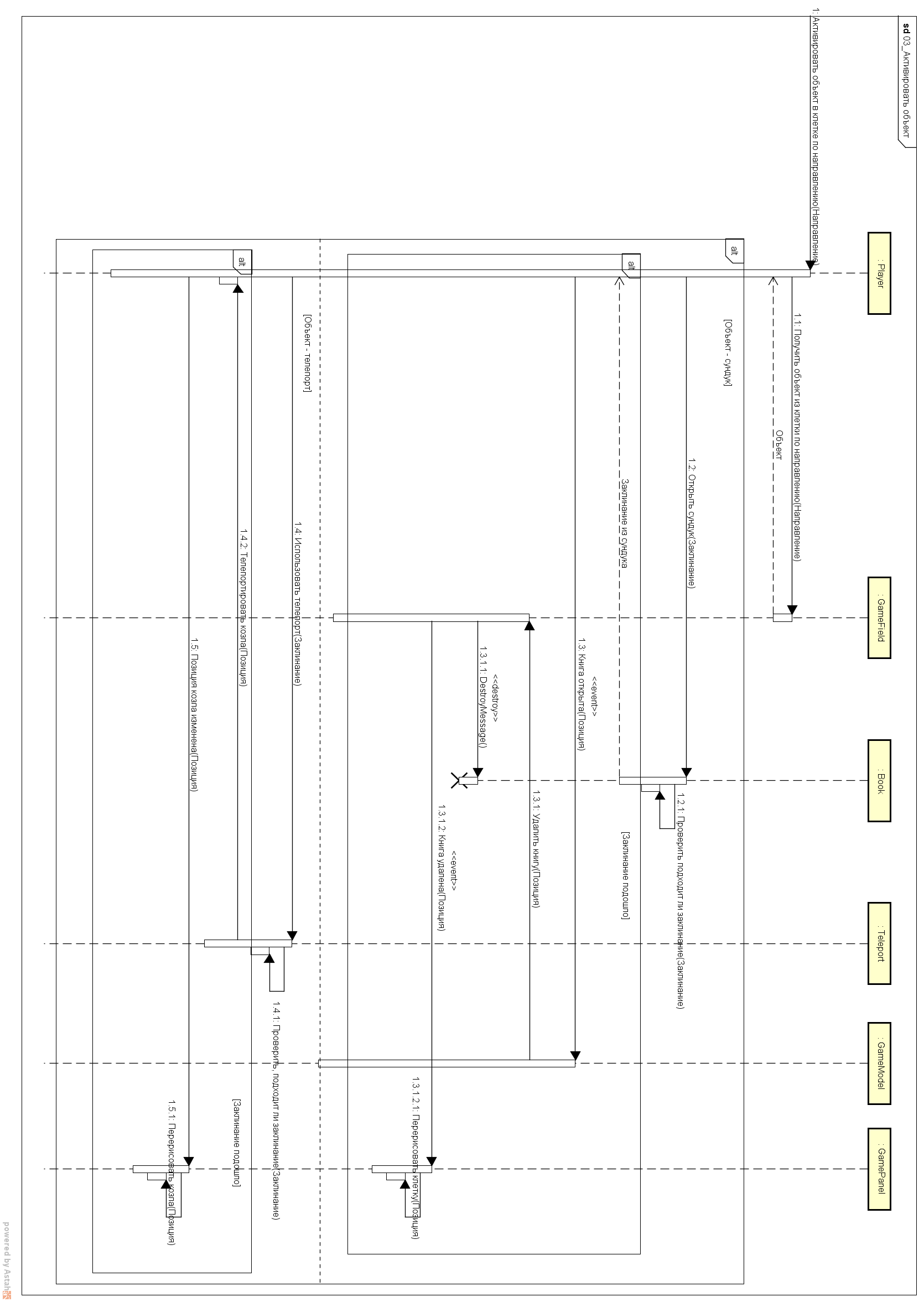
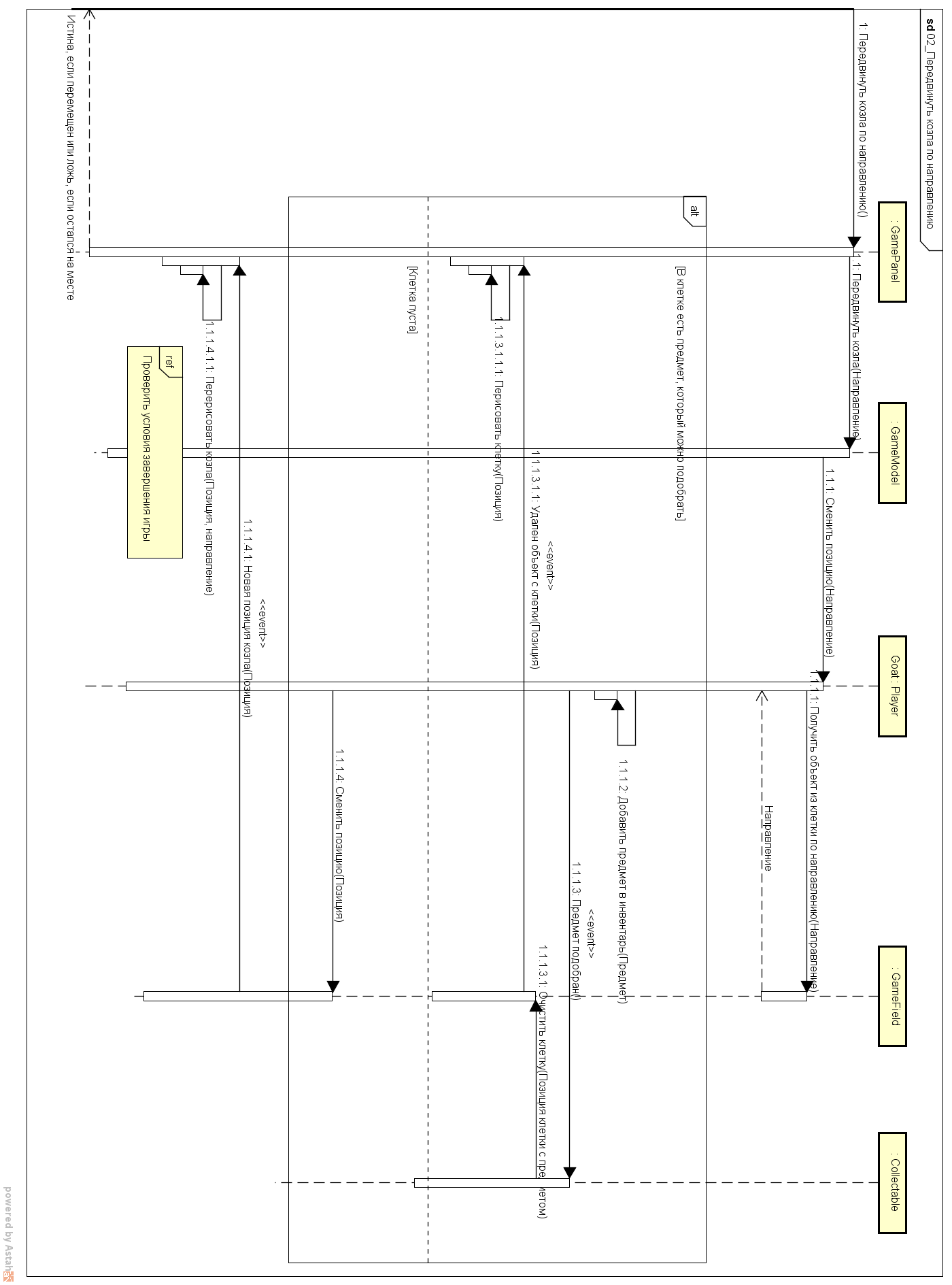
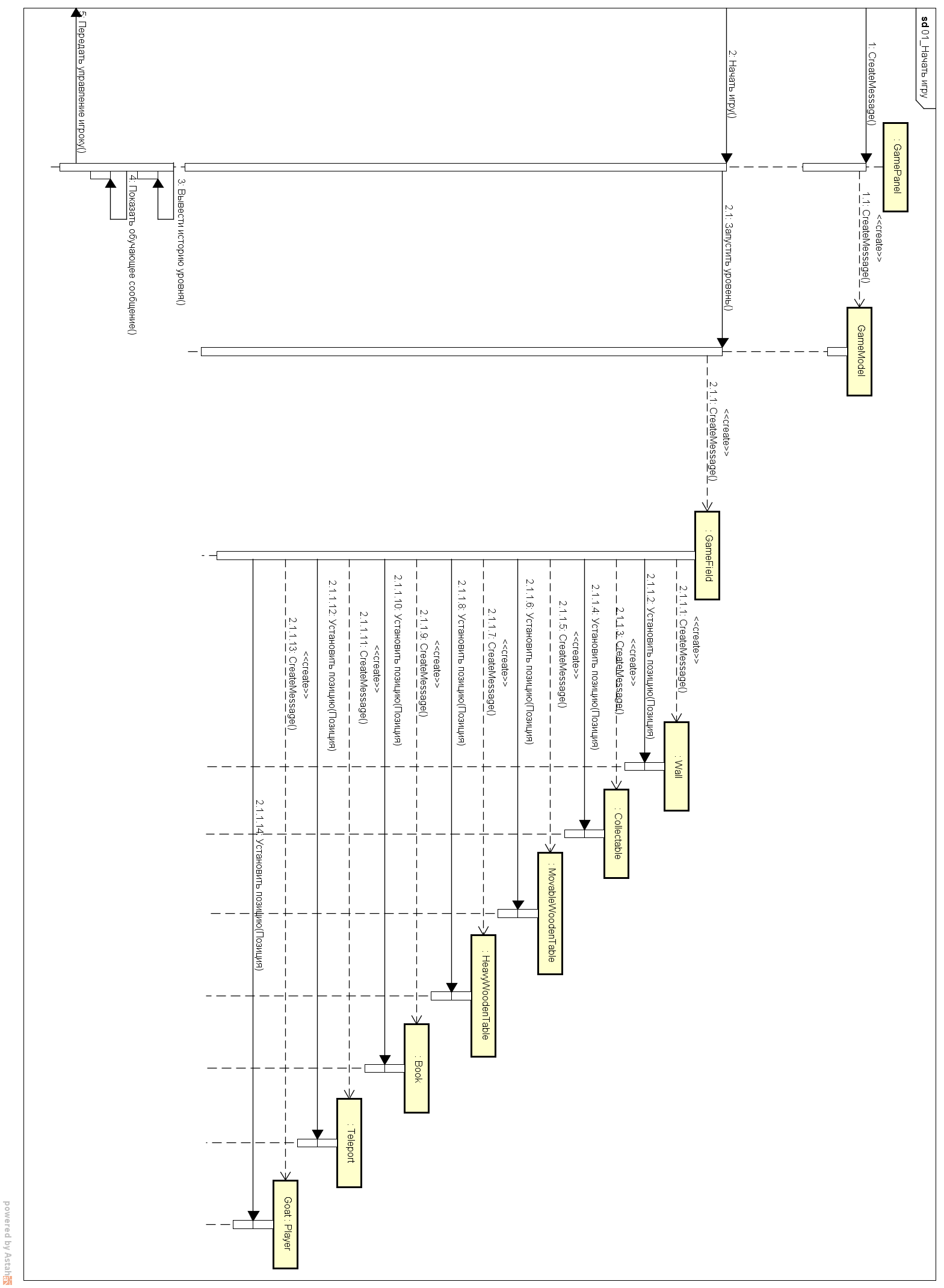
Книга – игровой объект, который может располагаться в ячейке. При активации книги, заклинание, которое она содержит, автоматически добавляется в инвентарь козла, а сама книга уничтожается. В ячейке может быть только одна книга, а сама книга может содержать только одно заклинание.

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

# 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