

Полный сценарий вычислительной модели игры "Следы" с вариативными элементами

1. Основные сценарии

1.1 Сценарий "Старт игры и инициализация уровня"

1. Пользователь запускает игру
2. Игра создает экземпляр GameManager
3. GameManager загружает конфигурацию уровня (JSON/XML)
4. GameManager создает Level с параметрами:
 - rows, cols (размер поля)
 - wallPositions (позиции стен)
 - keyPositions (позиции ключей)
 - startPosition (стартовая позиция)
 - exitPosition (позиция выхода)
 - teleportPosition (позиция телепорта, опционально)
5. Level создает:
 - LevelModel (логика поля)
 - LevelView (отображение)
 - LevelController (управление)
6. LevelModel инициализирует поле:
 - Создает обычные клетки (Cell)
 - Заменяет клетки на стены (Wall) по wallPositions
 - Размещает ключи (Key) по keyPositions
 - Устанавливает стартовую позицию (Start)
 - Устанавливает выход (ExitCell)
 - Устанавливает телепорт (TeleportCell, если есть)
7. LevelView отображает поле
8. GameManager переходит в состояние ожидания хода игрока

1.2 Сценарий "Успешное завершение уровня"

1. Игрок перемещается на клетку с выходом (ExitCell)

2. ExitCell проверяет:
 - Все ли ключи собраны (player.hasAllKeys())
3. Если ключи собраны:
 - ExitCell уведомляет GameManager через IExitCellActionListener
 - GameManager вызывает endCurrentLevel()
 - GameManager разблокирует следующий уровень
 - GameManager показывает окно победы (ResultWindow)
4. Если ключи не собраны:
 - ExitCell остается активной
 - ExitCell телепортируется на соседнюю незанятую клетку (меняется с ней местами)
 - Игрок остается на прошлой позиции ExitCell

1.3 Сценарий "Сбор ключа"

1. Игрок перемещается на клетку с ключом
2. Cell проверяет наличие ключа
3. Если ключ есть:
 - Ключ добавляется в инвентарь игрока (player.addKey())
 - Ключ удаляется из клетки (cell.removeKey())
 - LevelView обновляет отображение клетки
4. Игрок продолжает игру

2. Сценарии вариативных элементов

2.1 Сценарий "Подвижная клетка со следом"

1. Игрок перемещается на TeleportCell
2. TeleportCell:
 - Вызывает стандартное поведение (оставляет след)
 - Получает доступного для перемещения соседа (getFirstEnableNeighbour())
 - Если такой находится
 - * Меняется местами с выбранным соседом (swapWith(neighbour))
3. LevelView обновляет отображение обеих клеток
4. Игрок остается на новой позиции клетки

2.2 Сценарий "Подвижная клетка с выходом"

1. Игрок перемещается на ExitCell
2. ExitCell проверяет:
 - Если все ключи собраны:
 - * Уведомляет GameManager о победе
 - Если ключи не собраны:
 - * ExitCell получает первого доступного для перемещения соседа
 - * ExitCell меняется местами с выбранным соседом
 - * Игрок остается на старой позиции ExitCell
3. LevelView обновляет отображение

3. Словарь предметной области (расширенный)

Термин	Описание
GameManager	Управляет игровым процессом, переходами между уровнями и окнами
Level	Контейнер для конкретного уровня (модель, представление, контроллер)
LevelModel	Содержит логику игрового поля и состояние
LevelView	Отвечает за отображение игрового поля и UI
LevelController	Обрабатывает пользовательский ввод
AbstractCell	Абстрактный базовый класс для всех типов клеток
Cell	Стандартная проходимая клетка
Wall	Непроходимая клетка-препятствие
ExitCell	Клетка выхода. Вариация: при невыполнении правил игры меняется местами с соседней клеткой
TeleportCell	Вариация: телепортируется вместе с игроком в случайную позицию
Player	Игровой персонаж, перемещающийся по полю
Key	Собираемый объект, необходимый для завершения уровня
IExitCellAction Listener	Интерфейс для реакции на достижение выхода

