

Примерная речь для защиты дипломной работы:

Слайд 1: Актуальность

«Современная игровая индустрия сталкивается с необходимостью создания масштабного и разнообразного контента при ограниченных ресурсах. Процедурная генерация становится ключевым инструментом для решения этой задачи, особенно в жанре roguelike, где случайность и реиграбельность — основа игрового процесса. Однако выбор оптимальных алгоритмов остается сложной проблемой из-за отсутствия систематических критериев оценки. Это и определило актуальность моего исследования».

Слайд 2: Цель и задачи

«Цель работы — разработка методики сравнительного анализа трёх алгоритмов процедурной генерации: BSP, Drunkard Walk и клеточных автоматов. Для её достижения были решены задачи:

1. Анализ успешных кейсов применения алгоритмов в индустрии.
2. Теоретический обзор их принципов работы.
3. Формирование системы критериев оценки (скорость, связанность, баланс тупиков).
4. Создание модульного прототипа на Unity для тестирования.
5. Сравнительный анализ результатов».

Слайд 3: Практическая часть — архитектура

«Архитектура прототипа построена на принципах модульности и слабой связности.

- **LevelGenerator** координирует работу алгоритмов через единый интерфейс `ILevelGeneratorAlgorithm`.
- **LevelData** хранит результаты в унифицированном формате.
- Визуализация и анализ изолированы для чистоты эксперимента. На Sequence-диаграмме (рис. 3.2) показано, как пользовательский интерфейс взаимодействует с генератором, алгоритмами и модулем анализа».

На слайде представлена диаграмма взаимодействий, рассказать весь путь взаимодействия пользователя и ПО.

Слайд 4: Реализация алгоритмов

BSP (Binary Space Partitioning):

«Алгоритм рекурсивно делит пространство на узлы, создаёт комнаты и

соединяет их коридорами. Демонстрирует структурную целостность, но ограничен прямоугольными формами».

Drunkard's Walk:

«Агент случайно перемещается по сетке, формируя органичные пути. Уровень имеет высокую вариативность, но риск фрагментации».

Клеточные автоматы:

«Правила рождения/смерти клеток создают пещероподобные структуры
Показать примеры уровней.

Слайд 5: Интерфейс прототипа

«Интерфейс позволяет настраивать параметры алгоритмов в реальном времени. Здесь можно задать минимальный размер комнат для BSP, длину пути для Drunkard Walk или вероятность появления стен для клеточных автоматов. Результаты генерации сразу отображаются в окне Unity, что ускоряет тестирование».

Показать скриншот интерфейса.

Слайд 6: Метрики и анализ

«Для сравнения алгоритмов использовались метрики:

- **Время генерации:** BSP — 45 мс, Drunkard Walk — 12 мс, CA — 80 мс.
- **Связанность:** BSP — 100%, Drunkard Walk — 92%, CA — 88%.
- **Тупики:** BSP — 1%, Drunkard Walk — 18%, CA — 25%.

На тепловой карте видно, как Drunkard Walk создаёт более ветвистые структуры, а BSP — предсказуемые комнаты. Графики показывают зависимость времени генерации от размера уровня».

Слайд 7: Выводы

«1. **BSP** идеален для структурно сложных уровней с гарантированной связностью, но ограничен геометрией.

2. **Drunkard Walk** обеспечивает скорость и органичность, но требует контроля фрагментации.

3. **Клеточные автоматы** подходят для пещерных ландшафтов, но критичны к настройкам.

Для rogue-like игр оптимальна комбинация BSP и Drunkard Walk: первое задаёт структуру, второе — добавляет вариативность. Разработанный прототип и методика анализа могут быть интегрированы в игровые движки для ускорения разработки».

Заключение

«Спасибо за внимание! Готов ответить на вопросы».