

Разработка пошаговой стратегической игры с использованием интеллектуальных методов

Дипломная работа

Выполнил студент гр. 63502/1

А. А. Лесников

Научный руководитель

Л. А. Станкевич

2015 г.

Поставленные задачи

- Разработать прототип пошаговой стратегической игры
- Разработать гибкий и эффективный игровой ИИ для одиночного режима игры
- Предусмотреть кроссплатформенность клиентской части
- Предусмотреть возможность сетевой и локальной многопользовательской игры

Использованные компоненты

Для проекта необходимо разработать собственный графический движок, потому что свободные графические движки не удовлетворяют необходимым требованиям, а основывать проект на закрытых продуктах опасно в долгосрочной перспективе.

Внешними зависимостями являются:

gl-rs	https://github.com/bjz/gl-rs
glutin	https://github.com/tomaka/glutin
cgmath-rs	https://github.com/bjz/cgmath-rs
image	https://github.com/PistonDevelopers/image
stb_truetype	https://github.com/nothings/stb

Язык программирования

Рассмотренные языки: C, C++, D, Nim, Rust

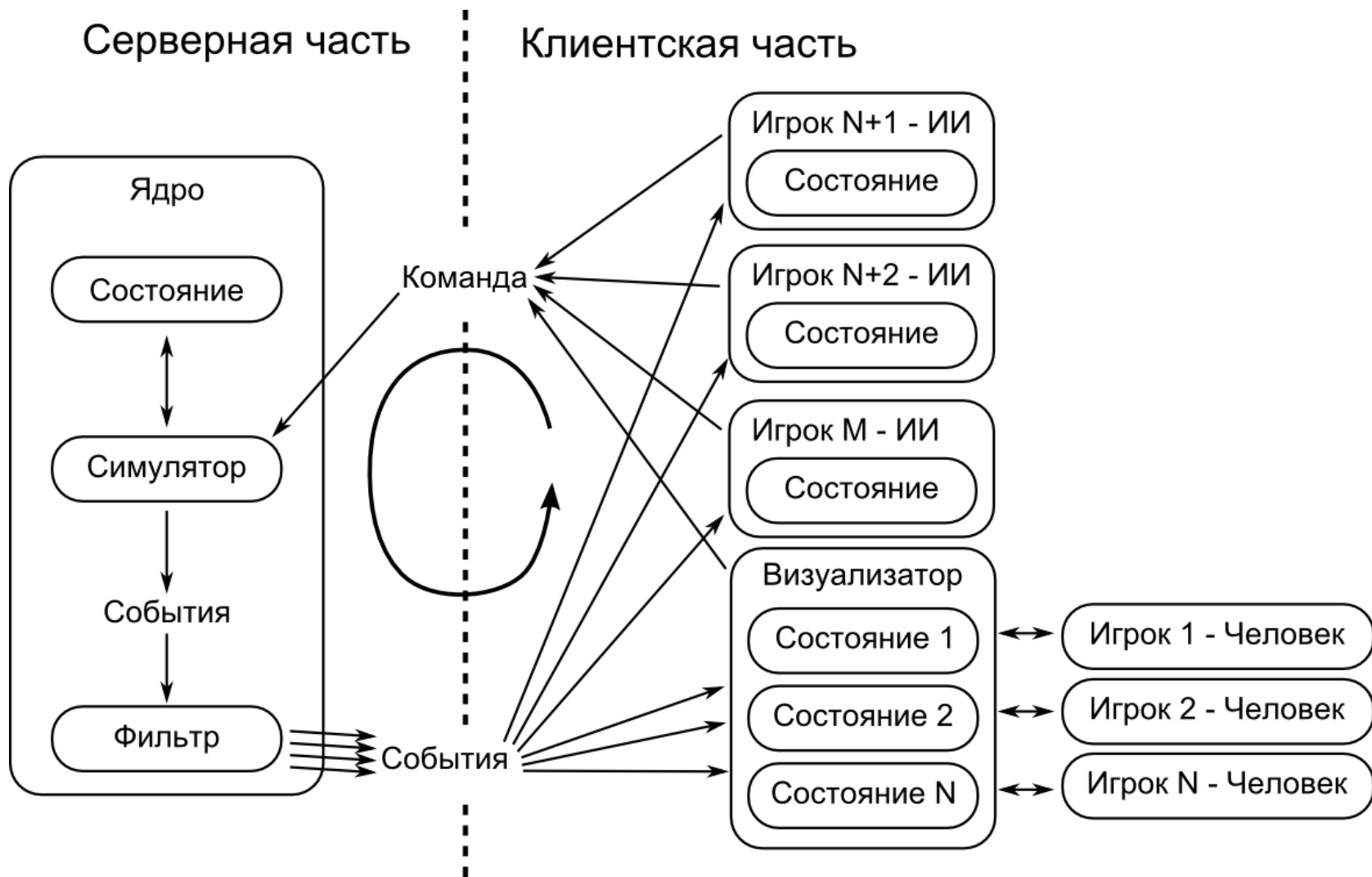
Особенности языка Rust:

- Явное разделение на опасный и безопасный код – «unsafe {...}»
- Безопасная работа с памятью без сборщика мусора
 - Интеграция «владеющих» указателей, концепции одалживания и явного времени жизни в язык
 - Отсутствие нулевых указателей
- Алгебраические типы данных и сопоставление с образцом
- Поощрение функционального стиля программирования
- Официальный компилятор использует LLVM для кодогенерации
- Мощное метапрограммирование: обобщенные функции, синтаксические макросы и плагины компилятора
- В активной разработке, не достиг версии 1.0
- Разрабатывается Mozilla для экспериментального браузерного движка Servo

Игровая механика

- Варгейм тактического уровня
- Гексагональная сетка
- Активный и реакционный огонь
- Система боевого духа / подавления
- Влияние местности на передвижение отрядов, эффективность огня и видимость

Архитектура проекта

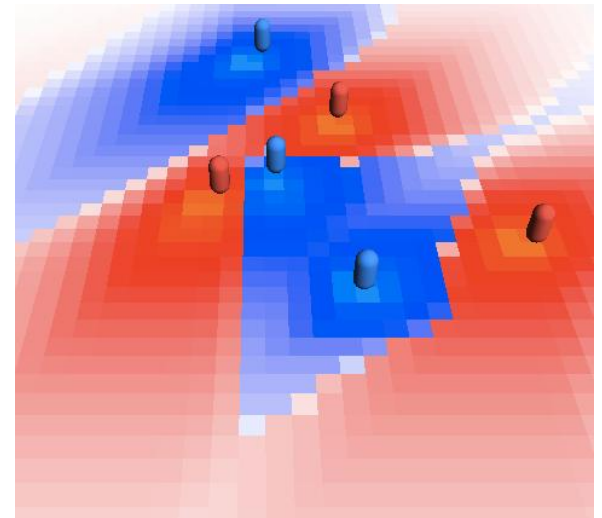


Подходы к реализации игрового ИИ

Архитектура	Достоинства	Недостатки
Конечные автоматы	Просты	Очень плохо масштабируются
Иерархические конечные автоматы	<ul style="list-style-type: none">•Просты;•Группируют поведения	Плохо масштабируются
Деревья поведения	<ul style="list-style-type: none">•Отделяют логику принятия решений от кода состояний;•Относительно просты	Жестко заданные приоритеты поведения
Планировщики	<ul style="list-style-type: none">•Хорошо решают уникальные и нестандартные ситуации;•Быстро приспосабливаются к новым действиям	<ul style="list-style-type: none">•Частичная потеря дизайнерского контроля;•Значительная нагрузка на процессор
Практичные ИИ	<ul style="list-style-type: none">•Хорошо решают уникальные и нестандартные ситуации;•Гибкая настройка поведения	<ul style="list-style-type: none">•Частичная потеря дизайнерского контроля;•Требуют представления всей информации в числах
Нейронные сети	<ul style="list-style-type: none">•Самообучаются•Просто подключаются к проекту	Полная потеря дизайнерского контроля

Практичный ИИ

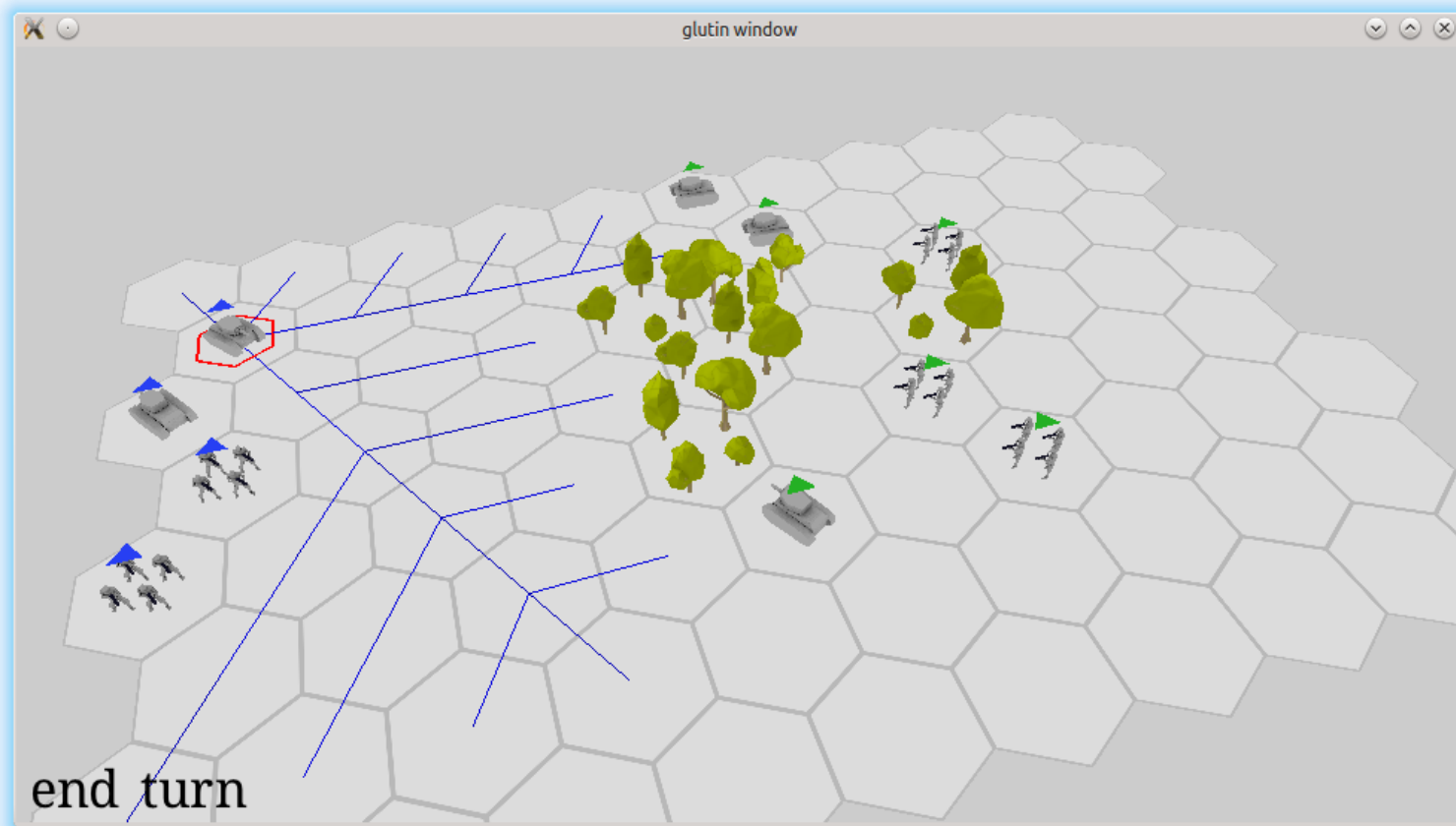
- Основан на теории полезности
- Принимает решение на основе «ожидаемой полезности» действий
- «Предельная полезность» - полезность одного и того же действия зависит от ситуации
- Активно использует «карты влияния» для анализа тактической ситуации и позиционирования отрядов
- Использует алгоритм Дейкстры для путенахождения



Использованные инструменты

- Редактор «vim»
- Компиляторы «rustc» и «gcc»
- Профилировщик «grprof»
- Системы сборки «cargo», «make»
- Система контроля версий «git»
- Веб-сервисы хостинга проектов «github» и «bitbucket»
- Веб-сервисы непрерывной интеграции «travis-ci», «shippable» и «appveyor»
- 3D редактор «Blender»
- 2D редактор «Gimp»
- Отладчик «gdb»
- Графический отладчик «qapitrace»
- Android SDK/NDK: «adb», «ant» и «ndk-build»

Внешний вид приложения



Дальнейшее развитие проекта

- Более глубокая проработка игровой механики
- Наполнение игры
- Разработка сценариев
- Добавление режимов игры
- Улучшение визуальной части
- Оптимизация производительности
- Добавление стратегического режима

Заключение

- Разработан графический движок и прототип игрового приложения
- Приложение кроссплатформенно: протестировано на платформах windows, linux и android
- Разработан эффективный игровой ИИ, основанный на теории полезности и картах влияния
- Проверено использование экспериментального языка Rust для разработки интерактивных систем мягкого реального времени

Спасибо за внимание