

Анализ и разработка алгоритмов (gamedev)

Цель курса: изучить основные методы анализа и синтеза алгоритмов, изучить и реализовать базовые алгоритмы (поиск, сортировка), познакомиться с базовыми и производными структурами данных (массив, список, дерево, куча, хеш-таблица), а также с алгоритмами управления памятью

| | | |
|--|--|--|
| | <p>Тема 1. Анализ алгоритмов</p> <p>Игра как программная система: архитектура, паттерны, реализация, алгоритмы, структуры данных, языки программирования, скрипты, данные, ресурсы вычислительной системы. Анализ алгоритмов, О-нотация, пространство и время.</p> | |
| | <p>Тема 2. Сортировка</p> <p>Задача сортировки, алгоритмы сортировки insertion sort и quicksort. Анализ производительности, $O(N^2)$ и $O(N\log N)$. Выбор опорного элемента. Размер данных и выбор оптимального алгоритма. Вопросы реализации алгоритмов и библиотек на C++. Обобщенные алгоритмы: шаблоны, перегрузка операций, лямбды. Создание стандартной библиотеки для игрового движка.</p> <p>ЛР: создание библиотечной функции сортировки, модульные тесты.</p> | |
| | <p>Тема 3. Контейнеры: массивы и списки</p> <p>Хранение данных и обеспечение доступа к ним. АД. Модель памяти. Хранение данных в массиве, операции вставки, удаления, обхода содержимого. Односвязные и двусвязные списки. Локальность данных, чанки. Концепция итератора. Производные контейнеры: стек и очередь.</p> <p>ЛР: создание библиотечных классов вектора и списка, сортировка и поиск в них, модульные тесты.</p> | |
| | <p>Тема 4. Организация данных для поиска: деревья и хеш-таблицы</p> <p>Задача поиска. Бинарное дерево поиска, операции поиска, вставки, удаления. Рекурсия. Балансировка. Левосклоняющееся красно-черное (LLRB) дерево. Хеш-таблица, операции поиска, вставки, удаления. Хеширующая функция и ее выбор.</p> <p>ЛР: создание библиотечных классов словаря на основе LLRB-дерева и</p> | |

| | | |
|--|---|--|
| | хеш-таблицы, модульные тесты. | |
| | <p>Тема 5. Управление памятью</p> <p>Динамическое размещение объектов, иерархия менеджмента памятью в игровых движках. Алгоритмы управления памятью. Управление памятью на основе битовых карт. Архитектура менеджера памяти. Пулинг объектов и специализированные аллокаторы. Вопросы реализации менеджера памяти на C++: перегрузка operator new.</p> <p>ЛР: разработка менеджера памяти для движка, использование его для контейнеров и объектов, модульные тесты.</p> | |
| | | |