▍ Вступление

Джон Кармак вместе с тёзкой Джоном Ромеро стали родоначальниками игрового жанра FPS. На слабых микропроцессорах Intel486SX, без видеокарт они выдали настоящий 3D-шутер без лагов с эффектом присутствия, что стало бомбой для 1993 года — в эпоху пошаговых стратегий и отсутствия скроллинга.

В разработке видеоигр Wolfenstein 3D и Doom Джон Кармак действительно использовал кучу нестандартных приёмов вроде конверсии указателей или выхода за границы объектов.

▍ Проекты

Чтобы понять уровень легендарности Джона Кармака, достаточно взглянуть на список его разработок, который включает в себя 41 игру, изданную с 1990 по 2012 годы. Некоторые экземпляры:

* Slordax, 1990 год, первая реализация сайд-скроллинга в играх на PC,
* Wolfenstein 3D, 1992, наложение текстур,
* Doom, 1993, BSP-деревья, отказ от музыки в пользу окружающих шумов (идея Кармака),
* Doom II, 1994,
* Heretic, 1994,
* Hexen: Beyond Heretic, 1995,
* Quake, 1996,
* Quake II, 1997,
* Quake III Arena, 1999,

В последующих играх (после 1999 года) он выступал уже не как ведущий программист, а как технический директор.

Игровые движки он разрабатывал практически в одиночку, тогда как другие сотрудники концентрировались на дизайне, уровнях и прочих игровых механиках.

▍ Детство за компьютером

Джон Кармак родился 20 августа 1970 года и увлёкся компьютерными играми в юном возрасте. В Тогда ещё не было интернета, поначалу маленький Джонди (Джон Д. Кармак Второй) ходил в библиотеку и искал какие-то книги по программированию, учился по ним.

В следующие годы он активно изучал программирование, играл в Ultima, но когда в возрасте 15 лет прочитал книгу Стивена Леви «Хакеры. Герои компьютерной революции», судьба подростка была решена. В книге рассказывалось о вундеркиндах, изменивших мир, и их культуре: экспериментах с мейнфреймами в MIT, самодельных программах, независимых компаниях по разработке игр.

Больше всего ему нравилось программировать графику, в том числе 3D-модели. Результат сразу был виден на экране. Первая игра Кармака называлась Shadowforge и очень походила на Ultima, хотя содержала несколько дополнительных фишек: например, персонаж мог атаковать не в четырёх направлениях, а в восьми. Джон заработал тысячу долларов, продав игру частной компании. Затем продолжал работать как фрилансер, продавая свои игры разным издателям, а в конце концов согласился принять предложение на работу от Softdisk. Так началась его игровая карьера как профессионального разработчика.

Работая в Softdisk, 19-летний Кармак придумал способ реализовать скроллинг на PC. Это было настолько прорывной и впечатляющей идеей, что коллега по имени Джон Ромеро уговорил вундеркинда забрать из старой фирмы рабочие компьютеры и основать свою компанию, которую они назвали id Software.



id Software, начало 90-х

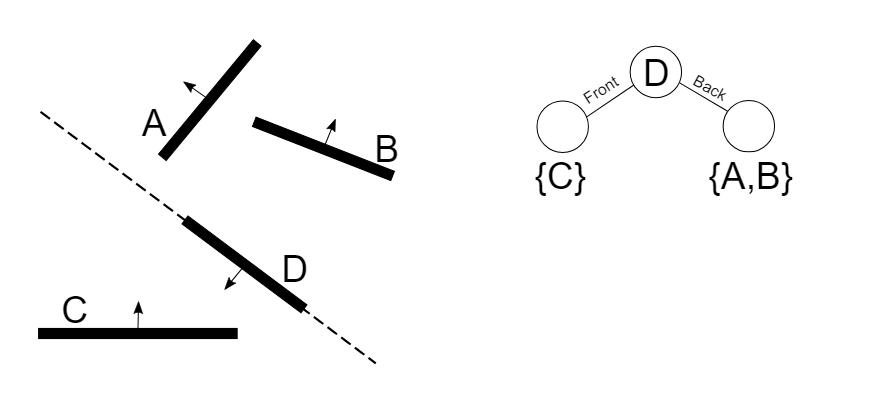
▍ Профессиональные хаки

Одним из подобных хаков является техника двоичного разбиения пространства (или BSP-дерева) — решение задачи отрисовки 3D-сцены. Её иногда называют определением видимой поверхности (visible surface detection, VSD).

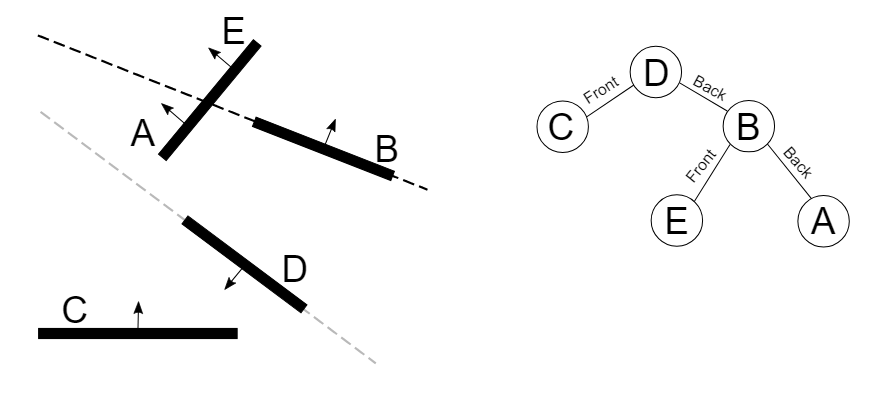
Для игры Doom пришлось написать принципиально новый рендерер, ориентированный на объекты.

Построенное дерево BSP легко использовать для обеспечения приоритетного порядка объектов в сцене. Вот примерная схема построения и обхода дерева BSP, представляющего простую двумерную сцену. В трёхмерной сцене геометрия сложнее, но идея та же самая:

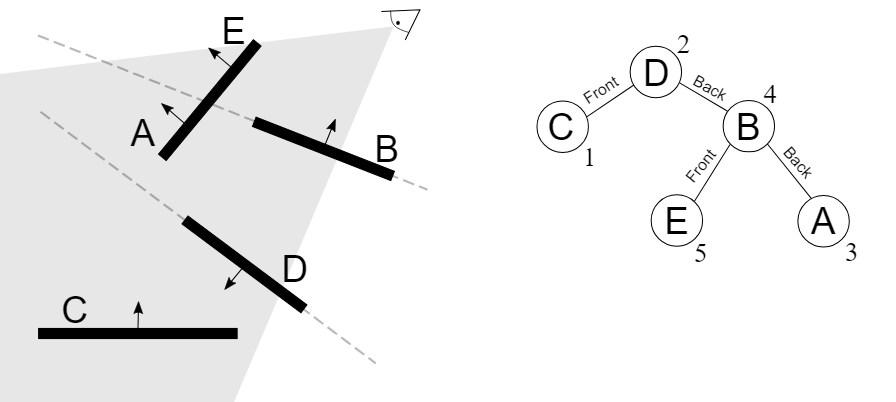
Первый шаг. Основная линия вдоль стены D разбивает сцену на два множества



Второй шаг. Полупространства по обе стороны D снова разделяются. Стена C — единственная в своём полупространстве, поэтому разбиение не требуется. Стена B образует новую разделительную линию в своём полупространстве. Стену A нужно разбить, потому что она пересекает линию раздела



Третий шаг. Упорядочение стен по отношению к точке обзора в правом верхнем углу, необходимое для алгоритма окраски пикселей. Здесь просто последовательный обход дерева



Замечательная особенность дерева BSP состоит в том, что его достаточно построить один раз, а затем одно и то же дерево BSP можно использовать для рендеринга сцены независимо от того, где находится точка зрения камеры. Вот почему дерево BSP так полезно для рендеринга в реальном времени — вся тяжёлая работа по построению дерева выполняется заранее, а не во время рендеринга.

Джон Кармак не только оценил мощь идеи, но и нашёл способ реализовать BSP-деревья в реальном времени на процессорах, которые даже не могли выполнять операции с плавающей точкой.

Некоторые специалисты считают, что именно эпизод с двоичным разбиением пространства — самый лучший наглядный пример того, почему Кармак стал такой легендарной личностью в программировании.

В 2000-е годы Кармак увлёкся конструированием ракет (Armadillo Aerospace, на фото он на ракетном фестивале X-Prize Cup 2005) и шлемов виртуальной реальности (Oculus Rift). А сейчас применяет свои способности для решения более интересной задачи — создания сильного ИИ (Keen Technologies). Джон Кармак считает появление Artificial General Intelligence (AGI, Strong AI) «возможным», а у него лично есть «ненулевой шанс внести свой вклад» в это чрезвычайно ценное изобретение.

▍ Introduction

John Carmack, along with John Romero, became the founders of the FPS game genre. On weak Intel486SX microprocessors, without video cards, they gave out a real 3D shooter without lags with a presence effect, which became a bomb for 1993 - in the era of turn—based strategies and the absence of scrolling.

In the development of the Wolfenstein 3D and Doom video games, John Carmack really used a bunch of non-standard techniques like converting pointers or going beyond the boundaries of objects.

▍ Projects

To understand the level of John Carmack's legend, just look at the list of his developments, which includes 41 games published from 1990 to 2012. Some instances:

• Slordax, 1990, the first implementation of side-scrolling in PC games,

• Wolfenstein 3D, 1992, texture mapping,

• Doom, 1993, BSP-trees, abandoning music in favor of ambient noise (Carmack's idea),

• Doom II, 1994,

• Heretic, 1994,

• Hexen: Beyond Heretic, 1995,

• Quake, 1996,

• Quake II, 1997,

• Quake III Arena, 1999,

In subsequent games (after 1999), he no longer acted as a leading programmer, but as a technical director.

He developed game engines almost alone, while other employees concentrated on design, levels and other game mechanics.

▍ Childhood at the computer

John Carmack was born on August 20, 1970 and became interested in computer games at a young age. At that time there was no Internet, at first little Jondi (John D. Carmack II) went to the library and looked for some books on programming, studied from them.

In the following years, he actively studied programming, played Ultima, but when at the age of 15 he read Stephen Levy's book "Hackers. Heroes of the computer revolution", the fate of the teenager was decided. The book told about the geeks who changed the world and their culture: experiments with mainframes at MIT, homemade programs, independent game development companies.

Most of all, he liked to program graphics, including 3D models. The result was immediately visible on the screen. Carmack's first game was called Shadowforge and was very similar to Ultima, although it contained several additional features: for example, the character could attack not in four directions, but in eight. John earned a thousand dollars by selling the game to a private company. Then he continued to work as a freelancer, selling his games to various publishers, and eventually agreed to accept a job offer from Softdisk. Thus began his gaming career as a professional developer.

Working at Softdisk, 19-year-old Carmack came up with a way to implement scrolling on a PC. It was such a breakthrough and impressive idea that a colleague named John Romero persuaded the prodigy to take working computers from the old firm and found his own company, which they called id Software.

▍ Professional hacks

One of the hacks is the technique of binary space partitioning (or BSP-tree) — solving the problem of rendering a 3D scene. It is sometimes called visible surface detection (VSD).

For the Doom game, we had to write a fundamentally new object-oriented renderer.

The constructed BSP tree is easy to use to ensure the priority order of objects in the scene. Here is an example diagram of building and traversing a BSP tree representing a simple two-dimensional scene. In a three-dimensional scene, the geometry is more complicated, but the idea is the same:

The first step. The main line along wall D splits the scene into two sets

The second step. The half-spaces on both sides of D are separated again. Wall C is the only one in its half—space, so splitting is not required. Wall B forms a new dividing line in its half-space. Wall A needs to be broken because it crosses the dividing line

The third step. The ordering of the walls relative to the viewing point in the upper right corner, necessary for the pixel coloring algorithm. Here is just a sequential traversal of the tree

A remarkable feature of the BSP tree is that it is enough to build it once, and then the same BSP tree can be used to render the scene regardless of where the camera's point of view is. That's why the BSP tree is so useful for real—time rendering - all the hard work of building a tree is done in advance, not during rendering.

John Carmack not only appreciated the power of the idea, but also found a way to implement BSP trees in real time on processors that could not even perform floating-point operations.

Some experts believe that the episode with the binary partitioning of space is the best visual example of why Carmack became such a legendary figure in programming.

▍

In the 2000s, Carmack became interested in designing rockets (Armadillo Aerospace, pictured at the X-Prize Cup 2005 rocket festival) and virtual reality helmets (Oculus Rift). And now he is using his abilities to solve a more interesting task — creating a strong AI (Keen Technologies). John Carmack considers the emergence of Artificial General Intelligence (AGI, Strong AI) "possible", and he personally has a "non-zero chance to contribute" to this extremely valuable invention.