Biography on John Carmack and various techniques he used in Videogame Engine Design

Introduction

John Carmack is an American software engineer, video game developer and hardware engineer. John Carmack is famous for co-founding ID software. While at ID software, he worked as lead programmer of many of its game series including Wolfenstein 3D, Doom, Quake and Commander Keen. He also worked as CTO for Occulus VR from 2013 to 2019 and founded an aerospace startup company called Armadillo Aerospace. In this biography I will be discussing some of the techniques he used in designing videogame engines as well as his passion for shareware and open source software

Wolfenstein 3D

Wolfenstein was the second franchise to be made by Id Software after the release of the commander Keen games. Wolfenstein was one of the first 3D first-person shooter games ever made and received both immediate and long term critical success. John Carmack worked as the lead programmer and designed the game engine for Wolfenstein 3D.

John Carmack faced several challenges when designing a 3D game engine. One such problem was that computational power at the time was seen as not sufficient enough to render a 3D environment in real time at an acceptable framerate for a fast paced action game. One solution John Carmack used to overcome this was to implement a ray casting system that only rendered surfaces visible to the player rather than the whole 3D environment to reduce the computational load. Another solution was to use a tile grid with texture mapping in order to make simple flat surfaces appear 3D whilst saving on computational power. The enemies were also 2D sprites as these would be easier to compute than £D models

John Carmack and the other members of Id Software were big believers in the shareware distribution model. The first episode of the game was released on shareware websites with users told they could order the other two episodes of the game with mail order. The commercial success of the game proved the viability of the shareware model and would be used by Id Software for future games.

**Doom (1996)**

Doom was a new first-person shooter designed by Id Software. John Carmack was once again tasked with designing the engine and came up with some new innovations from the Wolfenstein 3D engine. John Carmack designed a new file format called WAD an acronym for “Where’s all that Data?”. This file format stored all the data needed by the engine including the level design files and graphics files. This allowed any of this data to be changed without needing to change the engine making the game much easier to modify for enthusiasts. This along with Carmack giving fans access to the level editor allowed fans to create their own levels for the game. John Carmack also designed new features for the engine. Walls and doors could now be placed at any angle or height. He designed a new lighting system that calculated how far a surface was from a light source and made it darker the further it was rather than a more demanding ray tracing system.

One major innovation to the engine was the use of binary space partitioning in the engine instead of ray casting. A major challenge for a 3D engine is determining what is visible and what isn’t and rendering the surfaces that are visible. This is known as visible surface determination. For Wolfenstein 3D, John Carmack had used a ray casting system which worked because all walls were the same height and at a ninety degree angle to one another. As a result the only place a ray can intersect the wall is along a gridline and since every wall is the same height only one ray is needed for every column of pixels. This would not work as quickly for Doom as walls could be different heights and different angles. Binary space partitioning solved this problem by dividing a 3D scene into various parts ahead of time. By doing so the engine can figure out which parts of the 3D environment could obstruct the viewpoint of the player. This saves the need to check if all the various 3D objects were obstructing the viewpoint saving computational power. These parts that formed the entire 3D space was decided by a Binary Space Partitioning (BSP) Tree and had to be generated before the level began.

Doom 3 and the “Carmack Reverse”.

When designing the Engine for Doom 3 John Carmack wanted to add a system to generate shadows for a 3D scene in real time. The technique he used is called shadow volume or more specifically to Doom 3 was the stencil implementation of shadow volumes. The technique works by projecting a ray through objects to a predefined point. Any area between the object and the point the ray reaches is classified as in shadow. Any point which is not is not in shadow. Then for each object, the polygons are either marked as facing towards the light source or away from the light source. The point at which the polygons stop facing towards the light source and start facing away from the light source is marked the edge and creates the silhouette.

A picture containing text

Description automatically generated

A common technique used to make this work in real time is known as depth pass. However this technique would not work correctly is the viewpoint itself was in a shadow volume as it would invert the shadow. John Carmack came up with a solution that worked when the viewpoint was in a shadow volume itself preventing the shadows from being inverted.

**John Carmack’s support for Open source software**

John Carmack has been a vehement supporter of open source software for his entire career. Id Software which he co-founded has released the source code for many of their games including Wolfenstein 3D, Doom, Quake, Quake 2, Quake 3 and Doom 3. He also worked on Linux development helping to the OpenGL drivers for Linux aswell as designing linux ports for many of his games. John Carmack has publically stated his skeptisism for software patents stating:

“Get a dozen sharp programmers together, give them all a hard problem to work on, and a bunch of them will come up with solutions that would probably be patentable, and be similar enough that the first programmer to file the patent could sue the others for patent infringement.  
  
Why should society reward that? What benefit does it bring?”