

PS2 기반의 게임을 X-box에서 실행시켜라!

1

SOE 사례 연구 ... 특수 X-box 렌더링 엔진 개발

PS2 기반 게임 SOE(State Of Emergency)가 성공을 거두자 이 게임의 개발업체인 미국 VIS엔터테인먼트사는 이를 X-box에서도 게임이 가능하도록 변환시키기로 결정했다. 이는 SOE의 성공을 보다 오랫동안 누리기 위한 방안이었다. 시간은 충분했다. PS2와의 계약기간이 8개월이나 남아 있었기에 그 이전에 SOE의 X-box버전을 개발해야할 필요성은 없었다. 문제는 수작업으로 조정된 모든 코드를 수정해야 한다는 점이었다. SOE의 사례를 바탕으로 새로운 미션에 대한 해결법을 찾아보자.

「편집자 주」

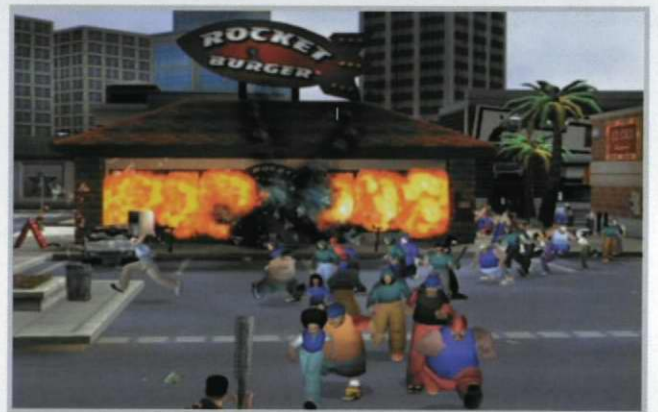
플레이스테이션2 기반 게임인 State of Emergency(SOE)를 X-box로 옮길 때 수반되는 문제점들이 있다. SOE는 PS2 게임으로 개발됐으며, 기반 기술인 렌더링 및 애니메이션 기술은 5000라인 이상의 수기(hand written) 벡터 유닛 어셈블리 코드로 구성돼 있다. 문제는 게임 전체가 PS2 아키텍처에서 최대의 성능을 발휘할 수 있도록 수작업으로 조정된 코드를 기반으로 구축됐다는 점이다.

여기서 설명하고자 하는 접근 방법은 특수한 고성능 PS2 하드웨어를 사용하는 엔진에서 성능이 극대화되도록 최적화된 특수 X-box 렌더링 엔진을 개발하는 것이다. 특히 이 코드는 이전에 벡터 유닛 프로세서를 사용했다. 그 결과 X-box 엔진은 PS2 버전과 동일한 수의 캐릭터를 구현할 수 있었지만, 초당 30프레임이 아니라 60프레임을 사용했다. 또한 SOE가 X-box와 대조적인 아키텍처에서 실행되면서 고성능을 얻기 위해 최적화 작업이 필요했다.

최초의 착상

SOE에 대한 처음의 착상은 폭동 시뮬레이터였다. 플레이어는 수백 명의 캐릭터를 통제해 수백 명의 다른 캐릭터들과 싸움하게 된다. 이 게임은 전략게임과 액션 장르가 혼합된 것이다. SOE 개발은 드림캐스트(Dreamcast)에서 시작했으며, SOE를 차세대 게임으로 제작할 방침이었다. 이 게임의 핵심

사양은 캐릭터의 수와 복잡성인데 이전 세대의 콘솔에서는 불가능했다. SOE가 목표로 한 고객은 드림캐스트가 본래 계획했던 것보다도 PS2의 목표 고객 집단에 근접했다.



> "State of Emergency" 화면 (PS2)

일단 2D 원형 게임이 제작됐으며, 게임플레이로 시연됐지만 그래픽 형태는 아니었다. 게임 플레이가 충분하지 않았기 때문에 게임의 스타일은 두 가지의 전혀 다른 장르를 혼합하는 대신 제3자의 액션게임으로 바뀌었다. 리드 플랫폼 역시 바꾸는 것으로 결정됐다. SOE 제작 전에 소니는 플레이스테이션 2(PS2)의 개발을 발표한 바 있다. SOE 개발을 담당했던 VIS 엔터테인먼트와 퍼블리셔인 록스타 게임즈는 PS2의 목표 인구

집단을 SOE의 목표 사용자 집단과 접목시키기로 결정했다. SOE는 PS2 발표가 다가오면서 고성능 PS2 엔진을 개발할 수 있는 기회였고, PS2가 사용되는 기간 동안 많은 타이틀을 기반으로 사용될 수 있었다.

PS2 버전의 필요 조건

PS2를 사용하는 게임의 주요 조건은 가능한 한 많은 수의 캐릭터를 디스플레이 할 수 있어야 한다는 것이었다. 이를 위해 고도의 전문 렌더링이 필요했다. 스크린에 많은 수의 캐릭터가 등장할 뿐 아니라 캐릭터들이 복잡한 인공지능(AI)을 갖추고 그들의 행동을 통제할 코드와 연결돼야 했다.

통합된 캐릭터 애니메이션과 제작 서브시스템을 갖춘 PS2용 렌더링을 개발하는 접근방식을 택했다. 이것은 PS2의 메인 CPU(감정 엔진 ; Emotion Engine ; EE)와 그래픽 프로세서 유닛(Graphics Processor Unit ; GPU-두 개의 벡터 유닛 ; VU를 사용해 제작됨) 사이의 관계를 이용하도록 설계됐다. PS2 렌더링 기기는 VU 상에서 가능한 최대로 캐릭터 애니메이션 기능을 담당했다. 이렇게 해 이용가능한 EE 시간을 극대화했으며 AI 전산화가 렌더링과 동시에 진행될 수 있었다.

이 렌더링 기기는 화면 제작에 필요한 온라인 프로세싱 분량을 최소화하기 위해 개발됐다. 이를 위해 마야(Maya)로부터 외형 데이터를 최종 형태로 변환하기 위한 오프라인 익스포터(offline exporter)가 만들어졌다. 최종적으로 변환된 데이터는 다이렉트 메모리 액세스(Direct Memory Access ; DMA)를 통해 PS2의 VU로 전송된다. 이 과정은 데이터가 저장되어 있지 않더라도 메모리 사용이 비능률적이다. 또한 지오메트리 데이터는 게임 엔진이 쉽게 접근할 수 있는 형태는 아니기 때문에 충돌 정보와 같은 게임 데이터는 디스플레이 데이터와 함께 별도로 저장돼야 한다.

PS2 렌더링 기기는 캐릭터와 AI뿐 아니라 병렬화를 극대화하고 렌더링 기기 별로 특수한 데이터를 오프라인으로 생성해야 한다. 이를 통해 SOE는 많은 캐릭터들을 동시에 시현하고 처리할 수 있게 된다. SOE의 전형적인 장면에 등장할 수 있는 캐릭터의 수는 100 이상이며, 하나의 프레임 내에서 250캐릭터의 행동까지 처리할 수 있다.

변환 일정

코드를 다른 플랫폼, 특히 X-box로 변환하는 것은 최초의 PS2 게임 개발 당시에는 고려되지 않았다. 이런 개발 방법론을 채택한 이유는 이 게임의 엔진과 제작기가 구체적으로 PS2의

하드웨어에 맞추어져 있었기 때문이다. 필요한 수의 캐릭터를 보여주기 위해서는 5000라인 이상의 VU 어셈블리 언어가 만들어진다. 이 단계에서 교차 플랫폼 조건을 고려했다면 PS2 버전의 성능은 확실히 저하됐을 것이다.

PS2에서 SOE가 성공을 거두자 VIS엔터테인먼트는 이 게임을 X-box에서 이용할 수 있도록 제작하기로 결정했다. 이런 결정은 게임 발표 직후 PS2에서 구현된 다중 플레이어 분리 스크린 모드를 연구하면서 가속화됐다.

게임의 이용가능성 테스트에서 최선의 방법은 이를 일반에게 공개하는 것이다. X-box 버전에서는 실제로 이 게임을 구매하는 사람들의 피드백에 기초해 SOE의 게임플레이가 발전할 수 있는 기회를 찾았다.

X-box 버전의 콘텐츠 및 수반되는 작업의 범위는 이 게임이 12개월간 PS2 전용으로 계약돼 있다는 사실에 의해서도 영향을 받고 있었다. 처음부터 이 게임은 빨라야 2003년 2월에 발표될 것으로 알려져 있었기 때문에 평균적인 경우보다 사용할 수 있는 시간여유가 많았다.

SOE X-box는 2002년 5월부터 12월까지 8개월의 프로젝트로 진행됐고, 14명의 핵심 구성원이 참가했으며, 이들 중 상당수는 파트타임으로 참여했다. 정확한 팀의 규모는 프로젝트의 필요에 따라 개발 도중에 계속 변화됐다. 평균적으로는 이 프로젝트에 8명의 전임 개발자가 참가하고 있었다. 주요 작업 부분은 소프트웨어 개발이었으며, 3~4명의 프로그래머가 렌더링 엔진과 게임코드 자체의 변환 작업에 참여했다. 또한 게임 디자이너 1명이 전임으로 참여하고 있었는데 미션 스크립트 제작을 담당했고, 감독 및 연출가 한 명 및 7명의 아티스트가 해상도를 증가시키는 작업을 담당하고 있었다. 8개월을 다시 나누어보면 제작기간이 5개월이었으며, 2개월은 알파-투-베타 시험에 사용됐고, 나머지 1개월은 베타-투-골드-마스터에 사용됐다.

초기 변환

개발과정에서 처음 3개월은 3 종류의 상이한 사전 처리 작업에 관여했다. 첫째로 연구개발 팀은 X-box 렌더링 개발에 착수했고, 둘째로 게임 디자이너 및 연출가는 PS2에서의 기능 검토 및 수정을 시작했다. 세 번째로 게임 프로그래머가 게임 엔진을 변환하기 시작해 X-box에서 실행되게 했다.

게임 엔진이 직접적으로 PS2 버전용 제작 데이터에 액세스하지 않아서 단순히 렌더링 기능을 불러오는 것으로 게임을 컴파일하고 작동시키는 것이 가능했다. 또한 X-box의 가상 메모

리 관리 시스템을 사용해 매우 신속한 게임 코드 변환이 가능했다. 게임 엔진 데이터용 메모리를 PS2에서 사용되던 어드레스와 동일하게 배치할 수 있었기 때문에 바이너리 맵 데이터가 절대 메모리 어드레스를 포함하고 있는 경우에도 변경되지 않고 온전하게 로딩될 수 있었다. 게임은 단 몇 시간만에 X-box에 컴파일 및 운용됐지만 TV 스크린에는 아무것도 나타나지 않은 상태였다.

X-box 개발 키트가 DirectX에 기초하고 있었기 때문에, 기능 렌더링은 단시간에 제작됐다. PS2에서 SOE가 사용했던 렌더링 기능은 게임 코드 변환 중에 불러내어 DirectX에 기초한 렌더링을 변환시킨 래퍼 라이브러리(wrapper library)에서 다시 구현됐으며 PS2에서 사용됐던 것과 동일한 콜 기능을 담당했다. 이런 초기의 변환은 약 3개월이 소요됐는데 상당한 작업이 DirectX 렌더링 및 래퍼 라이브러리 개발에 집중돼 있었다. 특히 이 래퍼 라이브러리를 사용해 SOE는 게임 코드를 많이 변화시키지 않고 렌더링을 사용할 수 있게 됐다.

X-box에서 게임할 수 있는 첫 번째 버전의 결과는 실망스러웠다. X-box의 사양 때문에 구현 속도가 PS2 버전보다 빠르지 않았다. X-box상의 SOE는 PS2에서보다 느렸다. PS2가 거의 모든 장면에서 초당 30프레임을 유지했고 이론적으로는 약 50%의 시간에서 초당 60프레임을 묘사할 수 있었지만, X-box SOE는 절반의 시간에서 초당 30프레임을 유지했으며, 초당 15프레임, 10프레임까지 떨어지는 경우가 빈번했다.

아키텍처

SOE를 X-box로 변환한 초기에 PS2와 X-box의 서로 다른 아키텍처가 어떠한 방식으로 전혀 다른 성능을 시현하는지 보여주었다. PS2에서 게임 엔진 CPU의 필요조건과 렌더링 성능간의 관계는 정교한 균형을 유지하고 있었다.

앞서 설명한 바와 같이 SOE가 PS2에서 성공할 수 있었던 이유는 EE와 VU의 유사성이 극대화됐기 때문이었다. 게임의 모든 면에서 제작을 위한 EE 프로세서 사이클의 조건은 미미했다. 이로 인해 많은 부분의 CPU 시간이 게임 엔진용, 특히 AI 행동용으로 사용될 수 있었다.

이것이 중요한 이유는 각 프레임의 많은 비 활동 캐릭터(Non-Player Characters; NPC)에 AI 프로세싱 및 경로 찾기의 성능이 필요하기 때문이다. AI 엔진의 특성과 PS2 디자인은 이러한 행동 처리에서 많은 프로세서 시간을 사용해야 한다. AI 엔진은 데이터 및 명령어 캐시 사용에서 효율적인 성능을 발휘하지 못한다. 기초적인 수준에서 AI 엔진은 루프로 작

동하며 많은 양의 복잡한 코드가 반복할 때마다 상이한 데이터 세트에서 기능을 실행한다. EE코드는 이러한 종류의 코드 성능 개선에 적극적으로 개입하지 않고, 몇 가지 다른 프로세서가 동일한 방식으로 기능을 수행한다. 따라서 PS2에서 병목현상이 생긴다면 이것은 AI 프로세싱에 의한 것이다.

X-box에서는 정 반대의 경우다. 애플리케이션의 제작 측면은 DirectX에 기록되며 이것은 하드웨어에서 분리된 추가 레이어이다. 이것은 지오메트리 렌더링이 CPU 사용을 많이 사용하기 때문에 동일한 장면의 제작에 있어서 PS2보다는 실질적으로 더 많은 시간이 필요하다는 것을 의미한다. 이에 따라 게임 엔진과 AI 코드는 특별한 변화 없이 빠른 속도로 작동할 수 있다. X-box의 CPU가 PS2의 EE보다는 상위의 성능을 가지고 있기 때문에 AI 코드는 X-box 프로세서에 더 적합할 것이라고 기대했었다. 하지만 PS2에서 전반적인 성능은 CPU와 GPU의 균형을 맞추어 유사성을 극대화함으로써 가능했다. 이런 유사성은 X-box에서 즉각적으로 명확하게 나타나지 않았고 초기에는 동일한 성능 수준에 이르지 못했다.

X-box에 대한 최적화 작업

PS2 게임과 동일한 수준의 성능을 구현하기 위해서는 특별한 최적화 작업이 필요했다. 이런 최적화 작업을 통해 게임의 성능이 향상되고 단일 플레이어 모드에서 초당 60 프레임, 화면분할 다중 플레이어 모드에서 초당 30 프레임을 유지할 수 있을 것으로 생각했다.



> "State of Emergency" 화면 (PS2)

구현돼야 할 주요 최적화 과제는 푸시버퍼, 애니메이션 캐싱 및 유사성을 극대화하기 위한 프레임 프로세싱의 재조직 등 3가지였다. 최적화를 시작하기 전에 주요 병목현상은 CPU의 사

용이었다.

▲ 푸시버퍼

푸시버퍼는 X-box GPU의 고유한 렌더링 포맷이다. DirectX가 X-box에서 사용될 때, 렌더링 명령어는 푸시버퍼로 변환돼 GPU에서 실행된다. 메인 렌더링 루프 내의 명령어가 역으로 푸시버퍼 코드를 결정하며, 대규모 푸시버퍼를 구축하는 오프라인 컨버터 애플리케이션 개발에 사용된다. 이 명령어에는 draw primitive, z-writes, set vertexbuffer, select vertexshader, set pixelshader 등이 있다. 또한 푸시버퍼 내의 각 객체는 비작동(null operation: NOP)에 앞서며 객체가 가시적이라면 NOP로 설정되고 가시적이지 않으면 건너뛰게 된다. 이 개념은 DirectX 3.0의 버퍼 렌더링 방법의 실행과 유사하다. 이 푸시버퍼 기법은 제작에 사용되는 CPU 셋업 시간을 줄여주며 프레임 타임을 절반으로 줄인다.

▲ 애니메이션 캐싱 작업

PS2의 애니메이션 시스템은 VU 어셈블리 코드를 이용해 구현되며, CPU를 거의 이용하지 않는다. 하지만 X-box에서 애니메이션 코드는 CPU상에서 구현돼 있기 때문에 CPU 병목현상의 원인이 된다.

애니메이션 코드를 최적화하기 위해서는 많은 캐릭터들이 동일한 애니메이션을 이용해야 할 것으로 생각된다. 대부분의 가시적 NPC가 공포 속에서 달리는 데 시간을 사용하기 때문이다. 또한 SOE에는 단 3 종류의 골격 유형만이 있어서 대부분의 애니메이션 매트릭스가 매우 제한적인 저장장치를 가지고 도 사전에 계산될 수 있다.

대용량 메모리는 최종 애니메이션 매트릭스 저장용으로 설정되며 그 저장용량의 절반이 캐시로 사용된다. 캐시로 설정된 각 매트릭스는 골격, 애니메이션, 프레임이 저장된 헤더, 점수 및 레퍼런스 카운터 등을 가지고 있다. 이 점수는 각 프레임마다 점차적으로 감소하며 조회 수가 0이 되고 점수가 낮으면 슬롯은 풀리는 것으로 간주된다. 검색 속도를 향상시키기 위해 첫 번째 x 슬롯은 프레임 0, 10, 20에 할당되고 두 번째 x 슬롯은 프레임 1, 11, 21 등에 할당된다. 만일 캐시가 모두 소진되거나 매트릭스의 변경이 필요하게 되면 메모리의 끝에 비 캐시 공간이 생긴다. 이러한 엔트리는 공동 매트릭스 세트를 포함하지만 조회 수를 제외한 추가정보는 없다.

▲ 렌더루프 재구성

두 솔루션은 CPU 사용을 크게 줄이지만 프로젝트가 끝날 무렵에는 게임의 GPU 사용이 증가하기 시작했다. 이것은 부분적으로 DirectX 프리미티브에서 푸시버퍼로 변경됐기 때문이지만 엔진의 기능이 증가하면서 제작되는 데이터의 양을 증가시켜서 발생한다. 또 다른 문제가 있었는데 첫 번째와 세 번째 프레임에서 CPU가 대량 사용된 데 이어 GPU가 대규모로 이용됐다. CPU 코드의 양은 적었지만 차단된 채로 GPU가 풀리기를 기다려야 했다.

최종 최적화 작업은 프로세싱을 2프레임 위에 겹치는 것이었는데 이렇게 함으로써 프레임의 초기에 푸시버퍼가 실행됐다. 이것은 많은 GPU를 필요로 했지만 시작된 이후로는 CPU를 필요로 하지 않았다. 이것이 작동하는 동안 모든 CPU 중심의 게임 엔진 코드가 작동했으며 렌더링 목록은 다음 프레임용으로 저장됐다. 이렇게 해 PS2와 유사한 방식으로 CPU와 GPU 사용의 균형이 이루어졌고, 유사성이 극대화됐으며 X-box는 대부분의 게임 플레이상황에서 초당 60프레임을 유지할 수 있었다.

X-box 버전에서의 차이점

SOE X-box가 원래의 게임에서 직접 옮겨진 것이 아니기 때문에 이 게임의 PS2 버전 이상의 것을 개발할 수 있었다. 구체적으로 보면 '레벨루션' 모드에 포함돼 있던 미션을 수정했으며, 게임 상의 그림 부분을 향상시켰고, 아케이드 모드의 멀티플레이어, 화면 분할 확장 기능을 추가했다.

SOE의 핵심은 군중과 NPC의 행동이다. 특히 NPC는 사건에 반응하며, 기타의 NPC는 이런 반응들에 반응한다. 본질적으로는 단순화된 빌딩 블록으로부터 상당히 복잡하고 디자인이 되지 않은 행동들이 나타난다. 이렇게 해 SOE의 세계가 생생해진다. 플레이어는 거리의 코너에서 특정 집행인을 따라가며, 두 라이벌 갱의 싸움을 말리기 위해 달리게 된다.

그러나 이런 행동은 불시에 일어나며 처음부터 게임 속에 디자인돼 있는 것이 아니기 때문에 매우 돌발적이며, 미션게임에서는 중요한 역할을 하지 않는다. SOE X-box에 있어서는 미션 게임이 광범위하게 검토됐다. PS2 게임을 마치고 X-box로 전환하기까지 3개월의 기간이 있었기 때문에 개발팀은 품질 및 미션의 즐거움 등에 대해 세밀하게 검토할 수 있었다.

대부분의 개발팀이 즐겁게 생각하지 않았던 미션은 제거되거나 수정돼 게임을 더욱 재미있게 만들었다. 예를 들면 무장하지 않은 파시스트가 많은 수의 무장한 집행인들에 의해 살해되지 않도록 해야 하는 플레이어의 미션은 즐거운 것이 아니었

다. 많은 수의 무장한 연합군의 도움을 받아 그보다 많은 수의 적 NPC를 제거하는 플레이어의 미션이 훨씬 더 흥미 있었다. 이러한 후자의 미션은 그리 많지 않았는데 구체적으로는 이 유형의 게임플레이를 가능하게 하는 것은 돌발적인 AI 구현에서부터 나오는 것이기 때문이다.

미술부문의 개선

X-box로 옮기면서 미술팀은 SOE의 미술부문을 수정할 수 있었다. PS2에서는 메모리의 사용이 매우 타이트했으며 대부분의 레벨에서 이용 가능한 32 메가바이트의 메인 메모리 중에서 일부 킬로바이트만을 사용할 수 있었다. X-box가 64 메가바이트의 RAM을 탑재하고 있었기 때문에 추가의 미술작업을 포함시키는 것이 가능했다.

첫째, 텍스처의 해상도가 증가됐다. 메모리와 텍스처의 캐시 문제에 있어서 PS2는 64x64 텍셀의 압도적인 텍스처를 가지고 있었다. 많은 경우 SOE의 텍셀과 픽셀의 비율은 실제적으로 1:1이지만 텍스처를 심도 있게 검토해 보면 그렇지 못한 경우가 많이 발견된다. 이러한 경우 128x128 텍셀의 수준까지 증가한다. PS2는 또한 텍스처를 사용해 메모리를 줄이고 제작 속도를 증가시킨다. X-box 버전은 본래의 24 비트 소스의 텍스처를 DXTC 압축과 함께 사용한다.

둘째, 추가 메모리를 사용해 게임의 다양성을 높이기 위해 미션 별로 특수한 캐릭터를 만들었다. 예를 들면, SOE PS2에서 모든 미션 제공자는 동일하지만 X-box에서는 6종류가 있다.

마지막으로 맵의 특정 요소들이 이국적인 셰이더와 함께 증가됐다. 선택된 자동차, 창문, 거리의 간판들이 배경을 갖추었으며 반사 맵이 추가됐고, 어떤 경우에는 광택이 있는 맵 요소가 추가됐다. 특히, 상점의 마루에는 반사되는 환경 맵 표면이 추가됐다. 이러한 표면 외관을 개선하기 위해 자동차와 같이 이들을 사용하는 특정 물체들의 다각형 해상도가 증가됐다.

멀티 플레이어 모드

SOE X-box에 추가된 가장 큰 요소는 아케이드 모드의 멀티플레이어 버전이다. 처음부터 렌더링은 PS2에서 화면 분할 성능을 갖추도록 디자인됐다. 사실 게임모드는 X-box 렌더링 기능을 발휘하기 전에 PS2에서 처음 원형으로 제작됐다.

SOE의 특정 요소들이 멀티플레이어 게임에 적합하지 않다는 것이 곧 발견됐다. 예를 들면, 손을 사용한 전투는 적이 주요 플레이어로서 싸우기 위해 일어설 때 싱글 플레이어 게임에

서 작동한다. 그러나 싱글플레이어 SOE에서도 공격받을 때 맞서다 도망치는 시민을 상대로 한 싸움은 매우 어렵다. 이 싸움을 재 디자인할 수 있어 멀티플레이어 싸움에 적용할 수 있었지만 싱글 플레이어 게임과는 다르다. 인간 적과 함께 손으로 싸우는 것에 집중하는 것부터 멀티플레이어 모드를 없애기로 결정했다.

SOE X-box의 장면 제작은 게임 엔진이 동일한 장면을 처리함에 있어서 좀더 긴 CPU 시간을 필요로 했다. 이 시간의 대부분은 기하학적 외형 설정 작업에 소요됐기 때문에 멀티플레이어 모드는 초당 60 프레임이 아니라 30 프레임으로 작동했다.

예상하지 않았던 문제점들

프로젝트의 기획단계에서 전혀 예상하지 못했지만 SOE X-box 개발 과정에서 발생한 많은 문제점들을 설명하고자 한다.

▲ 멀티플레이어 기능의 채택

SOE는 처음부터 싱글 플레이어 게임으로 디자인됐다. 따라서 멀티플레이어 요소를 변형해서 기존의 게임 엔진에 맞추어야 했다. 플레이어를 확대하기 위한 초기의 확장은 2일 만에 구현됐다. 이것은 렌더링의 화면 분할 기능을 포함해 도움이 됐다. 또한 메인 플레이어 캐릭터와 NPC는 엔진의 동일한 데이터 형태로 저장됐다. AI에 의해 통제됐던 NPC를 플레이어에 의해 제어하는 캐릭터로 바꾸는 것은 상대적으로 쉬운 작업이었다. 초기의 멀티플레이어 코드를 신속하게 추가하는 것이 가능했으며 많은 게임 코드는 사용자에게 의해 통제되는 플레이어를 캐릭터 제로로 간주했고 기타 모든 캐릭터는 AI 통제 하에 있는 것으로 간주했다. 게임 로직이 인간에 의해 제어되는 모든 캐릭터에 대해 일관되게 행동하지 않는 경우를 파악하고 수정하는데 상당한 QA 및 프로그래머 시간이 사용됐다.

▲ 하드웨어 차이로 인한 버그

서로 다른 플랫폼으로 변환하는 경우에 본래 있었던 버그를 발견하는 것은 특별한 일은 아니다. 하지만 SOE를 X-box로 변환하는 초기과정에서 이 버그의 효과는 예상하지 않았다. 특히 상당수의 코드 에러가 발견됐는데 X-box에서 바람직하지 않은 효과 (충돌 등)를 야기했다.

많은 경우에 있어서 이러한 버그는 PS2에서 치명적 결과를 초래하지 않았다. 대부분의 버그는 부동 소수점 계산의 에러 때문이며 NaN(not-a-number) 값의 결과를 가져왔다. 정밀

한 검사에서 PS2의 부동 소수점 유닛 (FPU)은 산술 동작을 다룰 때 훨씬 더 활발하게 작동해 NaN 값의 결과를 가져오는 것이 밝혀졌다.

FPU 에러 이외에 SOE X-box의 다른 메모리 맵으로 인해 발생한 버그는 메모리 붕괴라는 에러를 초래했는데 PS2에서는 발생할 수 없었던 에러였다. 통상적으로 이것은 추적이 매우 어려웠다.

SOE X-box의 대부분의 장면에서 게임은 싱글 플레이어 모드로 초당 60 프레임의 속도로 작동했고, 멀티플레이어 화면 분할 모드에서는 초당 30 프레임으로 작동했다. 이것은 싱글 플레이어로 초당 30 프레임을 유지했던 PS2 버전과 비교가 된다.

게임에서 프레임 속도가 초당 60 프레임 이하로 떨어지는 경우가 있었지만 PS2 버전에서도 때로 지속적인 프레임 속도유지에 문제가 있었던 곳이었다. 2인 플레이어 게임은 50 퍼센트 이상 초당 60 프레임의 속도로 작동할 수 있었으며 이런 성능을 유지하기 위해 필요한 최적화는 미술 및 게임 플레이의 품



> State of Emergency 화면(X-box)

질 저하를 가져올 수 있다고 결정돼 30 프레임으로 고정됐다. 3인 또는 4인 플레이어 게임 역시 대부분이 게임에서 초당 30 프레임을 유지한다.

성능은 다소 떨어지지만 의미있는 작업

SOE를 PS2에서 X-box로 변환하는 작업을 설명했다. 본래의 게임은 PS2 아키텍처에 최적화돼 있었으며 그 결과로서의 성능은 일차적으로 VU 어셈블리 언어로 작성된 자체 제작에 기인한 것이었다. 이를 통해 엔진은 CPU와 GPU 사이에 유사성을 극대화하는 것이 가능했다. 결과적으로 PS2에서의 성능은 미세하게 조정된 렌더링이 없었다면 이루어지지 않았을 것이다.

게임의 성능은 이런 PS2 아키텍처를 최대한 활용한 결과였고, 초기 X-box로의 변환 결과 전반적으로 PS2 보다는 성능이 저하됐다. 하지만 순수 프로세싱 파워에 기초해 X-box는 PS2 보다는 우수하지는 않더라도 동등한 성능을 구현할 것으로 기대된다. 이를 위해서는 X-box 렌더링에 대한 변경을 필요로 하는데 특히 DirectX 드로잉 프리미티브를 사용하는 코드를 X-box GPU의 고유한 제작 형식인 푸시버퍼로 변환해야 한다. 그 결과로 X-box의 SOE는 싱글 플레이어 모드에서 초당 60 프레임으로 작동했다.

전반적으로 두 플랫폼에서 SOE의 성능은 저급 프로세싱, 즉 캐릭터 애니메이션 전용으로 필요한 분야에서 우수한 성능을 발휘할 수 있는 특수 렌더링의 개발에 의해 이루어질 수 있었다. 또한 X-box 상에서 증가된 프레임 속도는 지오메트리 오프라인을 GPU 상에서 직접 실행할 수 있는 형식으로 프로세싱함으로써 이루어질 수 있었다. 이렇게 해 게임 엔진 코드용으로 이용할 수 있는 CPU 시간이 증가됐고 엔진의 유사성을 극대화시켰다.

“디지털콘텐츠 정보를 한권에”

정기구독을 하시면 직장이나 집에서 편리하게 디지털콘텐츠에 대한 정보와 전문지식을 습득하실 수 있습니다.

> 정기구독 신청 : 월간 <디지털 콘텐츠>

> 문의 : 02-3708-5361