연구에서 게임까지 : 3D공간 인터랙션

(From Research To Games: Interacting With 3D Space)

조셉 라비올라 주니어(Joseph LaViola Jr)

가마수트라 등록일(2010. 04. 22)

http://www.gamasutra.com/view/feature/6469/postmortem_zen_studios_pinball_.php

[3D 인터페이스가 소비자에게 소개되기 시작하자마자, 수년 간의 연구가 쏟아져 나왔다. University of Central Florida 의 Joe LaViola 박사는 게임과 상호작용하는 유용하고도 재미있는 방법을 고안하는 것에 대한 학술적인 결과를 소개한다.]

지난 몇 년 간, 우리는 모션 컨트롤러 기술이 비디오게임 산업에 영향을 미치고 있음을 목도했다. Sony EyeToy 와 Nintendo Wii 는 게임 인터페이스가 전통적인 키보드, 마우스를 넘어설 수 있고 게임 컨트롤러가 좀더 현실적이고, 몰입적이며 자연스러움 게임 플레이 메캐닉을 구현하고 있음을 보여주었다.

사실, 향후 몇 년 안에, Sony Move, Microsoft 의 Natal, Sixense 의 TrueMotion 을 비롯한 모든 메이저 콘솔 및 PC 는 3D 공간 인터랙션을 지원하는 인풋 영역을 개발할 것이다. 이 인터랙션 기술 분야에 있어서의 패러다임 이동은 게임 개발자가 게임 인터페이스를 만드는데 거대한 잠재력을 발휘하며, 이전에는 한번도 사용하지 않은 전략을 만들도록 해 주고 있다. 이러한 인터페이스는 플레이어가 좀더 몰입적인 경험을 할 수 있도록 해 주며, 행동을 더 근접하게 할 수 있도록 해 준다.

이러한 디바이스와 인터랙션의 스타일이 새로워 보이겠지만, 이것들은 실제로는 이미 연구 분야 내에서는 존재해 왔던 것이다. 사실, 가상 현실 분야와 3D 유저 인터페이스 분야는 지난 20 년 동안 이러한 인터랙션 스타일에 대해서 연구해 왔다.

90 년 대 이래, 이 분야에서의 연구 성과를 살펴 보면, 지금 수백만의 사람들에게 전해지고 있는 응용 영역에 활용되고 있는 혁신적인 3D 공간 인터페이스 기술은 이제 중요성을 드러내고 있는 것으로 보인다.

이 분야의 연구는 수년 동안 핵심 기술로 활용될 수 있는 콘텐츠를 찾아 왔고, 비디오 게임에 적용될 수 있는 절호의 기회를 맞이하였다.

나는 게임 산업이 학계에서 이루어 온 것을 잘 활용하고 있다는 생각이 든다. 가상 현실 및 3D 유저 인터페이스 분야가 수년 동안 연구되어 오는 동안 게임 개발에 직접적으로 응용되기까지 지식의 과잉 현상이 있었다.

게임 개발자가 최근 동작 센싱 인풋 디바이스 개발을 위한 혁신적인 인터랙션 기술 개발을 지속하고 있다는 것은 중요한 사실이겠지만, 또한 개발자가 불필요한 일을 할 필요가 없었다는 것도 중요하다.

즉, 이 글의 목적은 지난 10-15 년 간 학계에서 이루어진 3D 공간 인터랙션 기술을 검토해 보는 것이다. 게임 개발자가 이 글을 이용하여 3D 유저 인터페이스와 가상 현실에 적용할 수 있는 훌륭한 기저가 될 수 있기를 희망한다.

비디오 게임 에서의 3D 공간 인터페이스 활용뿐만 아니라 이 분야의 연구 정보를 보다 자세히 밝히기 위하여 글의 마지막 부분에 짧은 읽기 목록을 제공하고자 한다.

3D 공간 인터랙션

도대체 3D 공간 인터랙션이 무엇인가? 대략적으로, 3D 유저 인터페이스(3D 공간 인터랙션)은 이용자의 행위가 3D 인풋 디바이스로 3D 공간 콘텍스트에서 이루어지거나 2D 인풋 디바이스를 3D 로 맵핑하게 되는 곳에서 이루어지는 인간과 컴퓨터 간의 상호작용을 수반하는 UI 이다. 즉, 3D UI 는 입력 도구와 인터랙션 기술을 요하는데, 3D 컴퓨터 생성 콘텐츠를 효율적으로 컨트롤하기 위해서이다. 비디오 게임에 적용될 때도 예외는 없다.

대부분 복잡한 3D 어플리케이션에서 발견할 수 있는 기본적인 3D 인터랙션 태스크가 네 가지 있다. 실제로는 상징적 인풋- 알파벳으로 3D 환경에 접속할 수 있는 능력-이라 불리는 다섯 번째 태스크가 있지만, 우리는 여기서 이것을 논의하지 않을 것이다. 분명히, 어플리케이션 영역에 국한된 다른 태스크가 있을 것이다. 그러나 이 기본적인 빌딩 블록은 이용자로 하여금 좀더 복잡한 태스크를 할 수 있도록 해 주기 위해 자주 결합될 수 있다. 이 태스크는 이동, 선택, 조작 및 시스템 콘트롤을 포함한다.

Navigation 은 가장 일반적인 VE 태스크이고, 두 가지 요소로 구성된다. *Travel* 은 navigation 의 움직임 요소이고, 여기서 저기로 물리적인 움직임을 의미한다. *Wayfinding* 은 navigation 의 인지적이거나 의사 결정 요소인데, “내가 어디에 있는가?”, “어디고 가고 싶은 거지?”, “어떻게 거기에 가야 하지” 등등의 질문을 요구하게 된다.

Selection 은 간단히 말해서, 어떤 목적을 위한 일련의 대상 또는 대상의 상세 설명서이다.

Manipulation 은 대상의 특징에 대한 상세 설명서(대부분 위치, 방향 등의 특징)을 의미한다. Selection 과 manipulation 은 종종 함께 이용되지만 selection 은 독립형의 대상이다. 가령, 이용자는 “삭제”와 같은 명령을 수행하기 위하여 이 목적에 해당하는 대상을 선택하게 된다.

System control 은 인터랙션 모드나 시스템 상태를 바꾸는 태스크이다. 이것은 대개 시스템에서 명령 형태로 이루어진다. 2D 시스템에서의 예는 메뉴와 명령 수행 라인 인터페이스이다. 대개 시스템 컨트롤 기술은 세 가지의 태스크고 구성되지만(예, 메뉴 커맨드는 selection 을 포괄한다) 또한 특수 기술이 개발되고 일반적이 되면서, 별도로 간주되기 시작하였다.

3D 공간 인터페이스에 대해 생각할 때 일반적으로 대립하고 있는 두 가지가 있다: 현실과 마술. 현실적인 주제나 스타일은 실제 현실 인터랙션을 3D 환경에 가져오고자 한다. 따라서, 목적은 물리적인 현실 인터랙션을 가상 현실로 흉내 내도록 하는 데 있다.

골프 채나 야구 배트를 휘두르거나 가상의 물체를 집어 들기 위해 손을 이용하는 것과 같은 직접적인 조작 인터페이스가 예가 된다. 마술 주제나 스타일은 현실 세계를 공상과학이나 판타지의 영역으로 넘어서게 한다. 마술 기술은 주문을 외우거나, 날아 다니거나 공중부양으로 가상 물체를 옮기는 것과 같은 상상력에 의존 할 수 밖에 없다.

현실과 마술 3D 공간 인터랙션 기술에 모두 사용되는 두 가지 기술적인 접근법은 형태 유사성과 형태 비 유사성이다. 형태 유사성은 모션 컨트롤러와 가상 현실에서의 이에 대응하는 대상물 사이의 일대일 맵핑을 의미한다.

예를 들어, 모션 컨트롤러가 1.5 피트를 x 축을 따라 움직이면, 가상 대상은 같은 거기를 가상 공간에서 움직인다. 반면에, 형태 비 유사성은 입력을 조정하여 컨트롤에서 디스플레이의 비율이 똑같지 않게 만든다. 예를 들어, 모션 컨트롤러가 y 축에 대하여 30 도 각도로 회전한다면, 가상 대상물은 y 축에 대하여 60 도 회전할 수도 있다.

형태 비 유사성은 3D 공간 인터랙션에서는 매우 강력한 접근법인데, 그 자체로 마법과 같은 인터페이스가 될 수 있으며 이용자에게 가상 현실에서의 좀도 유연한 컨트롤을 제공할 수 있기 때문이다.

일반적인 3D 공간 인터랙션 기술

가상 현실이나 3D UI 문헌에서 찾을 수 있는 3D 유저 인터페이스 기술에 활용되는 핵심은 6 DOF 를 지원하는 입력 디바이스를 가지는 것이다. 이 경우에, 6 DOF 는 위치(x, y, z)와 방향(roll, pitch, and yaw)을 모두 제공하는 입력 디바이스를 의미한다.

일부의 경우에, 이용자의 머리 위치나 방향을 쫓을 수 있게 하는 것도 중요하다. 이것은 다소 현재 모션 컨트롤러 하드웨어에는 문제가 될 것이지만, 이 기술은 헤드 트랙킹이 가능하지 않을 때 이용할 수 있도록 수정될 수 있다.

6 DOF 는 3D 공간 인터페이스에서는 성전과 같은 원칙이며, 비디오 게임에서도 똑같이 활용된다.

Nintendo Wii Remote 와 같은 입력 도구를 가지고 3D 공간 인터페이스를 여전히 설명할 수 있지만, 입력 도구가 어떠한 상황에서든 DOF 를 제공하면서, 좀더 다른 방식의 입력 도구에 대한 수요가 발생했다. 이 글 끝의 읽기 목록에, Nintendo Wii remote 와 3D 공간 인터랙션을 둘러싸고 있는 문제를 어떻게 다룰 것인가에 대해 논의하고 있는 글을 소개하였다.

Navigation: Travel

Navigation 의 움직임 요소는 travel(시점 이동 같은 것)로 알려져 있다. 3D UI 에서 이동을 다룰 때 고려되는 몇 가지의 이슈가 있다.

이 중 하나는 속도와 가속도의 컨트롤이다. 이것을 하기 위한 여러 가지 방법이 있는데, 몸짓, 스피치 컨트롤, 슬라이더 등이 있다. 그 다음 이슈는 지형에 따르거나 일정한 높이를 유지하는 것과 같은 방식으로 동작이 어떤 방식으로 제약 되어야 할지 아닐지를 고민해야 하는 것이다.

가장 낮은 단계로, 입력 상태는 반드시 고려되어야만 한다. 즉, 언제, 어떻게 동작이 시작되고 끝나는지 (start/stop 클릭, start 는 누르고, stop 은 떼기, stop 은 자동적으로 특정 장소에서 되기, 등)? 좀더 일반적인 3D travel 기술은 이용자의 머리 움직임 측정, 포인팅, 지도 기반의 이동 및 “입체적인 움직임” 등의 네 가지로 정리한다.

Gaze-Directed Steering

Gaze-directed steering 은 가장 일반적인 3D 이동 기술로, 1995 년에 처음 논의되었다. 그럼에도 불구하고, “gaze”라는 용어는 오용된 것이다. 대개 눈동자 추적은 수행되지 않는다. 그래서 시선의 방향은 이용자의 머리 방향을 추적하여 이용된다.

이것은 실행, 이용 모두 간단한 기술이지만, 움직이는 동안에 주변을 볼 수 없다는 다소 제한점을 가지고 있다. 비디오 게임에서의 시점 이동에 대한 예시는 자동차를 움직이게 하거나 실시간 전략 게임에서 월드 주변을 돌아다니는 것 등이 있다.

시점 이동을 수행하기 위해서, 전형적으로 콜백 기능이 마련되고 렌더링 된다. 이 콜백이내에, 헤드 트랙킹 정보를 얻는다(대개 4x4 매트릭스 형태로). 이 매트릭스는 여러분에게 기본 트랙커 조정 시스템과 헤드 트랙커 조정 시스템 사이의 변형 값을 준다.

월드 조정 시스템과 기본 트랙커 조정(만약 있다면) 사이의 변형 값을 고려해 보라. 여러분은 전체 변형 값을 구할 수 있다. 이제, 헤드 트랙커 공간(마이너스 z 축)에서 벡터(0, 0, 1)를 고려해 보자.

월드 조정에서 표현된 이 벡터는 당신이 움직이고자 하는 방향이다. 이 벡터를 일반화 하고, 속도를 덧입혀서, 월드 조종에 대한 시점으로 변형한다. 주의할 점은 현재의 "속도"가 프레임으로 반영된다는 것이다. 여러분이 진정한 속도(units/second)를 원한다면, 여러분은 프레임 사이의 시간 트랙을 유지해야 하고, 이 시간에 맞추어 시점을 바꾸어야 한다.

Pointing

포인팅은 또한 1990 년 중반에 개발되었던 이동 기술이다. 이 경우에, 손의 방향이 방향을 결정하는데 이용된다.

이 기술은 다소 이용자가 배우기 어렵지만, 시선 이동 보다는 유연하다. 포인팅은 시선 처리와 정확히 똑 같은 방식으로 수행된다. 예외라면, 머리 대신 손을 추적한다는 것이다. 포인팅은 1 인칭 및 3 인칭 슈팅 게임에서 모션의 방향과 시야를 구분하는데 이용될 수 있다.

Map-Based Travel

지도 기반의 이동 기술은 목표 기반의 기술이다. 이용자는 2D 지도에 icon 을로 보여진다. 이동하기 위해서, 이용자는 이 아이콘을 새로운 지도상의 위치로 드래그한다(그림 1 참조). 아이콘이 떨어 질 때, 시스템은 부드럽게 이용자를 현재의 위치에서 새로운 위치로 이동시킨다. 지도 기반의 이동은 많은 게임 장르에서 발견되는 2D 게임 지도를 증가시켰다.

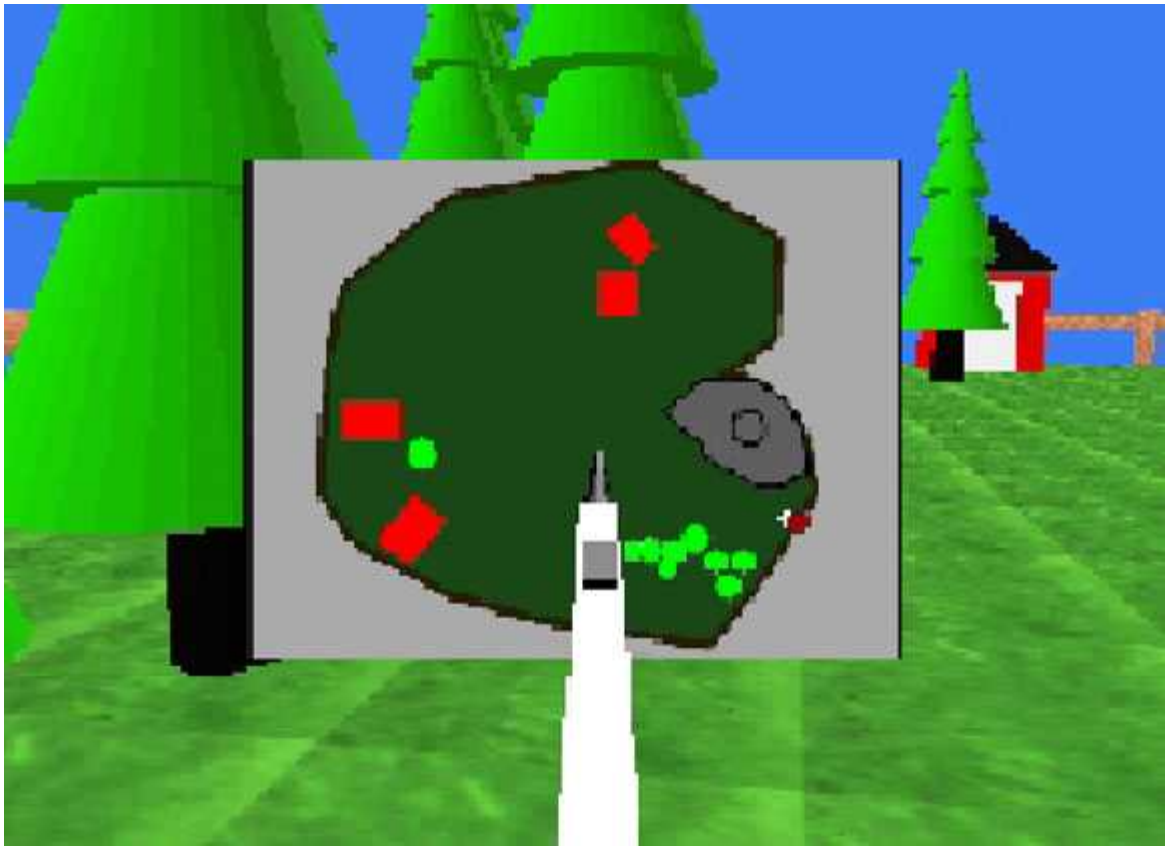

그림 1. 이용자 아이콘을 새로운 월드 내 장소로 움직이기 위하여 드래그 하기. 이 이미지는 1998 년에 그려졌다.

이 기술을 실행하기 위해서, 월드와 관련되어 있는 지도에 관하여 알아야 할 두 가지가 있다. 첫째, 우리는 스케일 요소를 알 필요가 있다. 지도와 가상 현실 사이의 비율 말이다. 둘째, 월드 조정 시스템의 시작점이 지도상에 어느 지점으로 나타나는지를 알아야 한다. 우리는 여기서 지도 모델이 월드와 같다고 가정한다. (즉, 지도 상의 x 방향은 월드 조종 시스템에서의 x 방향을 나타낸다.)

이용자가 버튼을 누를 때, 아이콘은 각 프레임에 맞추어 이동되어야 한다. 아이콘을 스타일러스에 단순하게 붙일 수 없다. 우리는 스타일러스가 그렇게 하지 못하더라도 지도 상에 아이콘이 남아 있기를 원하기 때문이다.

이것을 위해서, 우리는 먼저 지도 조종 시스템에서 스타일러스의 위치를 찾아 둔다. 조종 시스템 간의 변형 값을 필요로 하는데, 스타일러스가 지도의 아이가 아니기 때문이다. 스타일러스 위치의 x 와 z 조종은 아이콘이 움직여야만 하는 지점이다. 스타일러스가 지도 밖으로 드래그 될 때에 대해서는 여기서 논의하지 않는다, 그러나 이용자 아이콘은

스타일러스가 지도 영역 내에서 움직일 때까지 지도에 “고정” 되어야만 한다. 우리는 이용자가 월드 밖으로 나가기를 원하지 않기 때문이다.

버튼이 눌려지면, 우리는 월드에서 이상적인 시점의 위치를 계산한다. 이 위치는 지도 조종 시스템에서 월드 조정 시스템까지 변형 값을 이용하여 계산되며, 아래에 자세하게 설명하겠다.

첫째, 월드에 상응하는 지점에서 지도 조종 시스템 내 오프셋을 찾아라. 그리고 나서, 지도 스케일을 나누어라(지도가 1/100 사이즈라면, 이 것은 100 배 곱해야 한다). 이것은 우리에게 x 와 z 조정을 통한 시점 위치를 준다.

지도가 2D 이기 때문에, 우리는 여기서 y 값을 얻을 수 없다. 그러므로, 기술은 새로운 시점에서 이상적인 높이를 계산하는 방법을 필요로 한다. 가장 간단하게, 상수로 처리할 수도 있다. 다른 방법으로는, 장소의 지형적 높이 또는 다른 요소에 기반을 두는 것이 있다.

일단 우리가 이상적인 시점을 알게 되면, 시점의 애니메이션을 설정해야만 한다. 벡터 $\vec{m}$ 이동은 각각의 프레임에서 이루어지는 이동의 양을 나타낸다(선형적인 경로를 가정할 때). $\vec{m}$을 찾기 위해서, 우리는 현재 위치에서 이상적인 위치를 뺀다(전체 이동 양을 구할 수 있음), 그리고 두 지점 간의 거리로 이것을 나눈다(거리 공식을 이용하여 계산함). 그리고 이상적인 속도를 곱한다. 그 결과 $\vec{m}$은 우리에게 프레임마다 각각의 차원에서 움직일 수 있는 양을 알게 된다.

유일하게 남아 있는 계산은 이 동작이 이루어질 프레임의 수이다 : 거리/속도 프레임. 여기서 속도는 units/second 가 아니라 units/frame 으로 계산된다는 것을 다시 한번 주의해라.

Grabbing the Air

"grabbing the air" 기술은 1995 년에 처음 논의되었다. 여러분 주의의 세상(보통 빈 공간)을 문자 그대로 움켜 쥐려고 하는 은유를 사용하고, 손 동작을 이용하여 관여하도록 한다. 이것은 여러분을 로프로 묶어 당기는 것과 유사한데, 단지 “로프”가 어디에도 존재하지 않으며, 당신을 어느 방향으로든 데려갈 수 있다는 것이다. "grabbing the air" 기술은 건물이나 산을 오르거나, 수영 및 날기 등으로 비디오 게임에서 다양하게 잠재적 이용을 가할 수 있다.

이 기술의 한 손 버전을 실행하기 위해서(회전과 월드 스케일이 지원되면, 두 손 버전은 복잡해질 수 있다), 최초 버튼 입력이 감지될 때, 우리는 단순히 월드 조종 시스템에서 손의 위치를 획득한다.

그리고 나서, 버튼이 정지될 때까지 모든 프레임은 새로운 손 위치를 추적하고, 최초의 위치에서 이 값을 뺀다. 그런 다음 이 값에 때라 월드 내 대상물을 이동시킨다.

번갈아, 여러분은 반대 벡터로 시점을 이동시키거나 고정한 채로 월드를 떠날 수 있다. 콜백 이전에, "구" 손 위치를 다음 프레임에 이용할 수 있도록 업데이트 하는 것을 잊지 말아라.

Selection

3D 선택은 3D 가상 현실에서 하나 또는 그 이상의 대상에 접근하는 절차이다. 선택과 실행은 밀접하게 관련되어 있고, 여기에 설명될 몇 가지 기술은 조작을 위해 사용될 수 있다는 것을 주의하기 바란다.

선택 기술의 실행에 관하여 몇 가지 일반적인 이슈가 있다. 가장 기본적인 것 중의 하나는 선택 이벤트가 발생되는 것을 보여주는 방법이다 (예를 들어, 여러분이 대상물을 터치하고 있다면, 이제 그것을 들어 올리고 싶을 것이다). 이것은 대개 버튼 입력, 몸짓 및 목소리 명령에 의해 이루어지지만 시스템이 이용자의 의도를 추론할 수 있다면 자동적으로 행해질지도 모른다.

이 기술들의 대상 교차점을 위해서 효율적인 알고리즘을 가져야 한다. 우리는 몇 가지 가능성에 대해 논의할 것이다. 어떤 대상이 선택되는 가에 관해서 이용자에게 주어지는 피드백은 매우 중요하다. 이 기술의 많은 부분이 이용자의 손을 위해 아바타를 필요로 한다.

"선택 가능한" 대상의 리스트를 주의 깊게 보자. 그래서 선택 기술은 월드 내 모든 대상에 테스트 할 필요는 없다. 4 가지 일반적인 선택 기술은 가상 손, 레이-캐스팅, 폐색 및 팔 확장이다.

The Virtual Hand

가장 일반적인 선택 기술은 간단한 가상 손으로, 가상의 대상을 직접 "건드려서" "실제 세상" 선택을 하도록 한다. 이 기술은 또한 1980 년 대 말에 논의된 가장 오래된 방식이다. 햅틱 패드백 없이, 이 조작은 가장 손에 의해 이루어졌다 (물리적인 손과 적어도 같은 위치에 있다).

가상 손은 비디오 게임 장르에서 대단한 잠재성을 가진다. 1 인칭 슈팅 게임에서는 스포츠 장비, 직접적인 총 선택, 무기류 및 헬스 팩이 될 수 있으며, 실시간 전략 게임에서는 "신"의 손으로, 액션/어드벤처 게임에서는 퍼즐 인터페이스가 될 수 있다.

이 기술을 구현하는 것은 간단하다. 당신이 좋은 인터섹션/콜리젼 알고리즘을 가지고 있다면 말이다. 인터섹션은 대상물의 실제 지형 이라기 보다는 축이 나란하게 둘러싼 박스나 구형으로 수행된다.

Ray-Casting

또 다른 일반적인 선택 기술은 레이 캐스팅이다. 이 기술은 1995 년에 처음 논의되었고, 레이저 포인터의 은유를 이용한다. 무한한 빛의 확장. 빛이 가로지르는 첫 번째 대상이 선택하가능하게 된다.

이 기술은 실험 결과를 토대로 볼 때 효과적이다. 그리고 이용자가 2 가지의 자유도를 가질 수 있도록 한다(손목의 pitch 와 yaw). 가상 손이나 다른 장소 기반의 기술에 의해 3 DOFs 가 획득되는 것에 비해서 적다. 이상적으로, 레이 캐스팅은 게임에 특별한 힘이 필요하거나 플레이어가 멀리서 대상을 선택하는 능력을 가지려고 할 때 이용될 수 있다.

레이 캐스팅을 실행하는 많은 방법이 있다. 공격 접근법은 손의 위치와 방향에 기반을 두고 빛 방정식 파라미터를 계산할 것이다. 먼저 travel 용도 포인팅 기술에서 처럼, 월드 조종 시스템을 벡터 (0, 0, 1) 방정식으로 알아 낸다. 이것은 빛의 방향이다. 손의 위치가 (x_h, y_h, z_h)로 나타난다면, 벡터 방향은 (x_d, y_d, z_d)이고, 파라미터 방정식은 아래와 같이 주어진다.

$$x(t) = x_h + x_d t$$
$$y(t) = y_h + y_d t$$
$$z(t) = z_h + z_d t.$$

$t > 0$인 상황이 주어져야 하는데, 우리는 손의 “뒤”를 고려하지 않기 때문이다. 실제 지형물이 가로질러지는 지 아닌지를 결정하는 것은 중요하다. 따라서 첫 번째 가로지름 테스트는 투영되는 모든 상황에 영향을 미치게 된다.

또 다른 방법은 좀더 효율적일지도 모른다. 이 방법은, 월드 조종 시스템에서 손의 방향을 보는 대신에, 우리는 선택 가능한 대상이 손의 조종 시스템에 있다고 생각하고, 꼭지점이나 박스 자체를 변형시킨다. 이것은 비효율적으로 보일 수도 있는데, 세상에는 수 많은 폴리곤이 있음에도 불구하고 하나의 손만 있기 때문이다. 그러나 우리는 선택 가능한 대상 리스트에 의해 제한된 대상을 가진다고 가정한다. 그래서, 인터섹션 테스트는 훨씬 더 효율적 이라고 할 수 있다.

일단 우리가 꼭지점을 변형하게 되며, 각각의 꼭지점의 z 조종 값을 알 수 있다. 이 것은 3D 폴리곤을 2D(손 조종 시스템의 xy 면)의 평면으로 맵핑할 수 있다.

빛이 이 조종 시스템에서 (0, 0,-1)이기 때문에, 우리는 2D 평면에서 이것을 볼 수 있고, 빛은 오직 (0,0) 포인트가 폴리곤에 있을 때만 폴리곤을 가로지르게 될 것이다. 우리는 쉽게 양의 방향으로 x 축을 가로지르는 2D 폴리곤의 가장자리를 중요하게 생각하는 알고리즘을 가지고 이것을 결정할 수 있다.

Occlusion Techniques

폐쇄 기술(또한 이미지 평면 기술이라고도 불림)은 1997 년에 처음 제안되었고, 이미지 평면 작업에 이용되었다; 가상 손에 의해 "커버링"됨으로써 대상이 선택되고 그 결과 여러분의 시선을 가린다.

기하학적으로, 이것은 빛이 여러분의 눈에서 나와서 손가락을 통해 지나간 다음 대상물에 부딪힌다는 것을 의미한다. 폐색 기술은 장거리에 있는 대상 선택을 위해 사용될 수 있지만, 레이 캐스팅의 은유를 이용한 포인터를 활용하는 대신에, 플레이어는 간단하게 선택할 대상물을 멀리서 "건드리기만"해도 된다.

이 기술은 레이 캐스팅 기술과 같은 방식으로 실행될 수 있다. 빛을 사용하는 방식이기 때문이다. 여러분이 공격 레이 인터섹션 알고리즘을 하고 있다면, 눈의 위치에서 손가락의 위치를 빼서 빛의 방향을 정의하기만 하면 된다.

그러나, 여러분이 선택된 알고리즘을 사용하고 있다면, 빛의 조종 시스템을 정의하기 위해서 대상을 필요로 한다.

이것은 두 단계로 이루어 질 수 있다. 먼저 비어있는 대상물을 만들고, 손의 위치에 그것을 자리하게 한다. 그리고 나서 월드 조종 시스템과 나란히 둔다. 그 다음, 이 대상/조종 시스템을 어떻게 회전시킬 것인지를 결정한다. 빛의 방향을 나란히 하기 위해서이다. 각도는 눈과 손의 위치를 이용하여 결정할 수 있고, 간단한 삼각법을 이용할 수 있다. 3D 에서, 두 개의 회전은 빛으로 새로운 대상물 조정 시스템을 조절하기 위해서 필수적이다.

Arm-Extension

팔 확장(예를 들어 Go-Go) 기술은, 1996 년에 처음 논의되었는데 80 년대의 가제트 만화에서 영감을 얻고, 가상 손에 기반을 두었다. 그러나 이것은 물리적 손과 가상 손 사이의 선형적이지 않은 맵핑을 소개하고 있는데, 그래서 이용자의 접근이 대단히 확장될 수 있다.

장거리에서의 대상 선택에 유용할 뿐 만 아니라, Go-Go 는 배트맨이 고리를 움켜지거나 스파이더맨의 거미줄과 같은 환경에서의 이동에도 유용하다. 그림 2 의 그래프는 x 축에 몸으로부터의 손의 물리적 거리와 y 축에 몸으로부터의 가상 손의 거리를 맵핑한 것을 보여주고 있다.

물리적 손이 'D' 지점에 도달할 때 까지는 일대 일 맵핑이 적용된다. D 지점을 지나면, 비선형적인 맵핑이 적용되는데, 이를 통해 이용자가 뻗을 수 있는 구간이 더 넓어지고, 가상 손은 더 빨리 움직이게 된다.

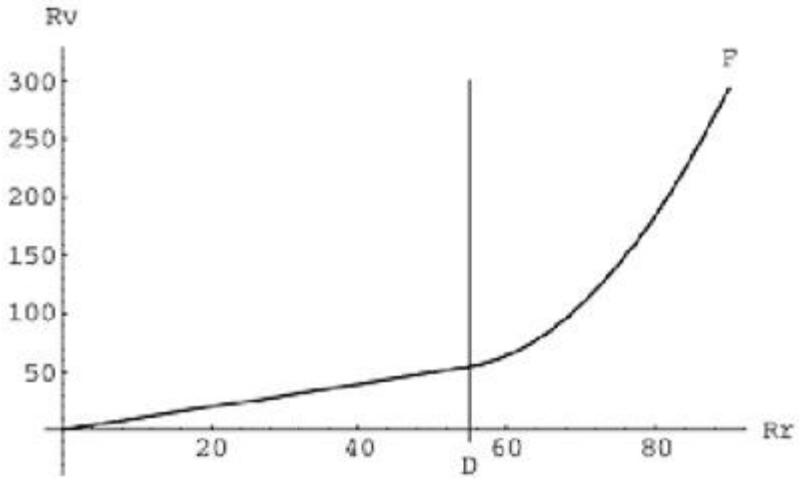

그림 2. Go-Go 선택 기술에 이용되는 비선형적 맵핑 함수

Go-Go 를 실행하기 위해서, 우선적으로 이용자의 신체의 위치에 대한 개념이 필요하다. 머리가 아니라 몸의 중심에서 손을 뻗기 때문이다. 우리는 이것을 토르소 포지션을 이용하여 실행할 수 있는데, 머리에서 음의 방향으로 y 축의 상수 오프셋을 정의할 수 있게 된다. 트랙커는 이용자의 토르소에 위치하도록 할 수 있다.

프레임을 렌더링 하기 전에, 월드 조종 시스템에서 물리적 손의 위치를 구한다. 그리고 나서 거리 공식을 이용하여 토르소 대상으로부터 거리를 계산한다. 가상 손 거리는 그림 2 에 있는 그래프에서 보여지는 함수를 적용하여 구할 수 있다.

$d^{2.3}$(D 에서 시작함)은 유용한 함수이지만, 이용되는 지수는 선택의 정확성과 환경의 크기에 의존한다. 가상 손이 어디에 있는지를 알게 되면, 우리는 이 위치를 결정해야 한다.

가장 일반적인 구현은 가상 손을 토르소로부터 빛이 나아가는 방향에 두고 물리적 손을 지나가도록 하는 것이다. 두 지점 사시의 벡터를 구할 수 있다면, 이것을 일반화 하고, 거리를

곱한 다음 이 벡터를 토르소에 더하여 가상 손의 위치를 구한다. 결국, 우리는 가상 손 기술을 이용할 수 있다.

Manipulation

앞서 언급했듯이, 조작은 선택과 관련 있다. 대상이 조작되기 전에 선택되어야만 하기 때문이다. 그래서, 어떠한 조작 기술이든 간에 중요한 점은 주어진 선택 기술과 얼마나 잘 통합되는 것이냐는 것이다. 앞서 밝힌 많은 기술은 이 둘의 통합을 잘 하고 있다: 가상 손, 레이 캐스팅 및 Go-Go.

대상이 조작되어 질 때 중요한 또 다른 점은, 여러분이 선택 기술을 잘 다루지 못한다는 것을 고려하고 선택에 대한 피드백을 이용자에게 주어야 한다는 것이다. 이것이 이루어지지 않으면, 심각한 문제가 발생할 수 있다. 예를 들어, 이용자는 현재 선택된 대상을 취소하려고 하는데, 시스템은 이것을 새로운 대상을 선택하려는 것으로 해석할 수 있다.

물체를 놓을 때 어떤 일이 발생할 것인지를 고려하는 것은 중요하다. 마지막 위치에 남아 있는가, 떠 있을 가능성이 있는가? 뭔가 단단한 것에 부딪힐 때까지 중력에 의해 떨어지는가? 어플리케이션 필수 요소는 이러한 선택을 결정하게 할 것이다.

세가지 일반적인 조작 기술은 HOMER, Scaled-World Grab, World-in-Miniature 이다. 이러한 기술 각각을 위해서 게임의 대상 조작은 다양한 장르에 활용되며, 1 인칭 및 3 인칭 슈팅 게임의 함정 높기, 액션/어드벤처 게임에서의 퍼즐 경쟁, 새로운 형태인 리듬 게임 지원하기 등이 대표적이다.

HOMER

Hand-Centered Object Manipulation Extending Ray-Casting (HOMER) 기술은 1997 년에 처음 논의되었다. 선택을 위한 레이 캐스팅을 이용하고 가상 손을 손 중심의 조작을 위한 대상물로 이동시킨다. 오브젝트의 깊이는 선형적 맵핑에 따른다.

최초의 토르소-물리 손 거리는 최초의 토르소-대상 거리에 맵핑되고, 이에 따라 물리 손이 멀리서 두 번 움직이면, 오브젝트 또한 멀리서 두 번 움직인다. 또한. 물리 손을 토르소 가까이로 다시 움직이면, 오브젝트 또한 이용자의 토르소 가까이로 움직인다.

Go-Go 와 같이, HOMER 는 토르소를 필요로 하는데, 여러분이 이용자의 몸(토르소)와 물리 손 사이에 가상 손을 두고자 하기 때문이다. 여기서의 문제는 HOMER 가 선택에 따라 오브젝트의 물리 손으로부터 가상 손으로 움직인다는 것이고, 토르소, 물리 손 및 오브젝트가 항상 일렬로 라인업을 한다는 것을 보장할 수 없다는 것이다.

그러므로, 가상 손이 최초의 빛이 있었던 곳에 있는지를 계산하고, 가상 오브젝트의 지점의 오프셋을 계산한 다음 이 오프셋을 조작 과정에서 유지하도록 한다.

레이 캐스팅을 통해 대상이 선택될 때, 손 트랙커에서 가상 손을 분리한다. 가상 손 모델이 물리적 손 위치에서 이동했음에도 불구하고 분리되지 않고 남아 있다면, 물리적 손의 회전이 가상 손의 회전과 변형을 초래할 것이기 때문이다. 그 다음, 월드 조종 시스템에서 선택된 대상물의 위치로 가상 손을 움직여서, 스크린 그래프에 있는 가상 손으로 대상을 합친다.

선형 뎁스 맵핑을 실행하기 위해서, 우리는 토르소와 물리적 손 사이d_h 및 토르소와 선택된 대상 d_o 사이의 거리를 알아야 한다. d_o/d_h 비율은 스케일링 요인이 될 것이다.

각 프레임에서, 우리는 가상 손의 위치와 방향을 설정해야 한다. 선택된 대상은 가상 손에 붙어 있는데, 그래서 이것은 따라 다니게 될 것이다. 방향을 설정하는 것은 상대적으로 쉽다. 간단히 손 트랙커의 매트릭스를 가상 손에 복사해 두고, 이것으로 방향을 매칭한다.

위치를 정하기 위해서, 우리는 올바른 깊이와 올바른 방향을 알아야 한다. 깊이는 현재 물리 손의 깊이와 선형적인 맵핑을 적용하여 찾을 수 있다. 물리 손의 거리는 간단하게 토르소 사이에서 구할 수 있고, 우리는 이것을 가상 손 거리에서 구한 factor d_o/d_h를 곱한다. 그리고 나서 물리 손과 토르소 사이의 일반화된 벡터를 구하여 이 벡터를 가상 손 거리에 곱한다. 그 다음, 토르소 위치 결과값을 더하여 가상 손 위치를 구한다.

Scaled-World Grab

scaled-world grab 기술(그림 3 참조)는 1997 년에 처음 논의된 폐색 선택을 이용하는 방식이다.

여러분이 이미지 평면에서 대상을 선택한다는 생각에서 기인한 것으로, 이미지의 모호함을 이용하여 마술적인 이미지를 구현한다. 선택이 이루어진 다음, 이용자는 시점을 위로 이동하게 되는데, 그래서 가상 손이 실제로는 보이지 않았던 것을 터치하게 된다.

이용자가 움직이지 않는다면(그리고 그래프가 스테레오가 아니라면), 스케일링 이전과 이후의 이미지 사이의 인식적인 차이점은 없다.

그런, 이용자가 움직이기 시작하면, 자신이 거인이 아니라는 것을 깨닫고(또는 세상이 너무 작은 것이 아님을 깨닫고) 대상을 직접 조작할 수 있다.

scaled-world grab 을 실행하기 위해서, 올바른 액션이 선택과 취소에 동시에 수행되어야만 한다. 이 둘 사이에 어떤 특별한 것도 필요하지 않다. 대상은 단순히 가상 손에 붙어 있고, 간단한 가상 손 기술로 영위되기 때문이다. 선택과 동시에, (눈과 대상물 사이의 거리/눈과 손의 거리) 비율로 이용자를 조정하라.

이 스케일링은 고정된 지점에서 눈으로 이루어진다. 그래서 눈은 움직이지 않고 3 차원에서 동일해야만 한다. 가상 대상을 가상 손에 합일하라. 떨어지게 될 때, 반대 액션이 반대로 이루어져야 한다. 월드와 대상을 분리할 때, 스케일링 팩터를 동일하게 조정하고, 다시 고정된 지점에 눈을 이용하여라.

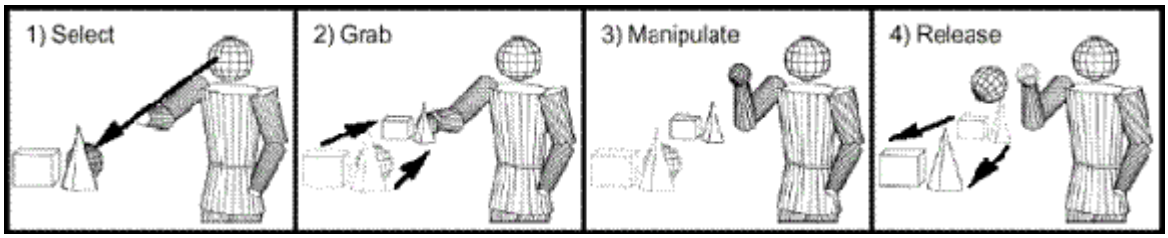

그림 3. scaled-world grab 기술의 일러스트레이션

World-in-Miniature

world-in-miniature (WIM) 기술은 작은 “인형 집” 버전을 이용하여 이용자가 간접적으로 대상을 조작할 수 있게 해 준다(그림 4 참조). WIM 내의 각각의 대상물은 간단한 가상 손 기술을 이용하여 선택되는데, 이 대상을 움직이는 것은 같은 방식을 움직이는 세상의 풀 스케일 오브젝트를 발생시킨다. WIM 은 또한 3 차원을 포함한 맵 기반의 이동 기술과 비슷한 방식으로 이용자를 재연하여 이동할 수 있도록 해 준다.

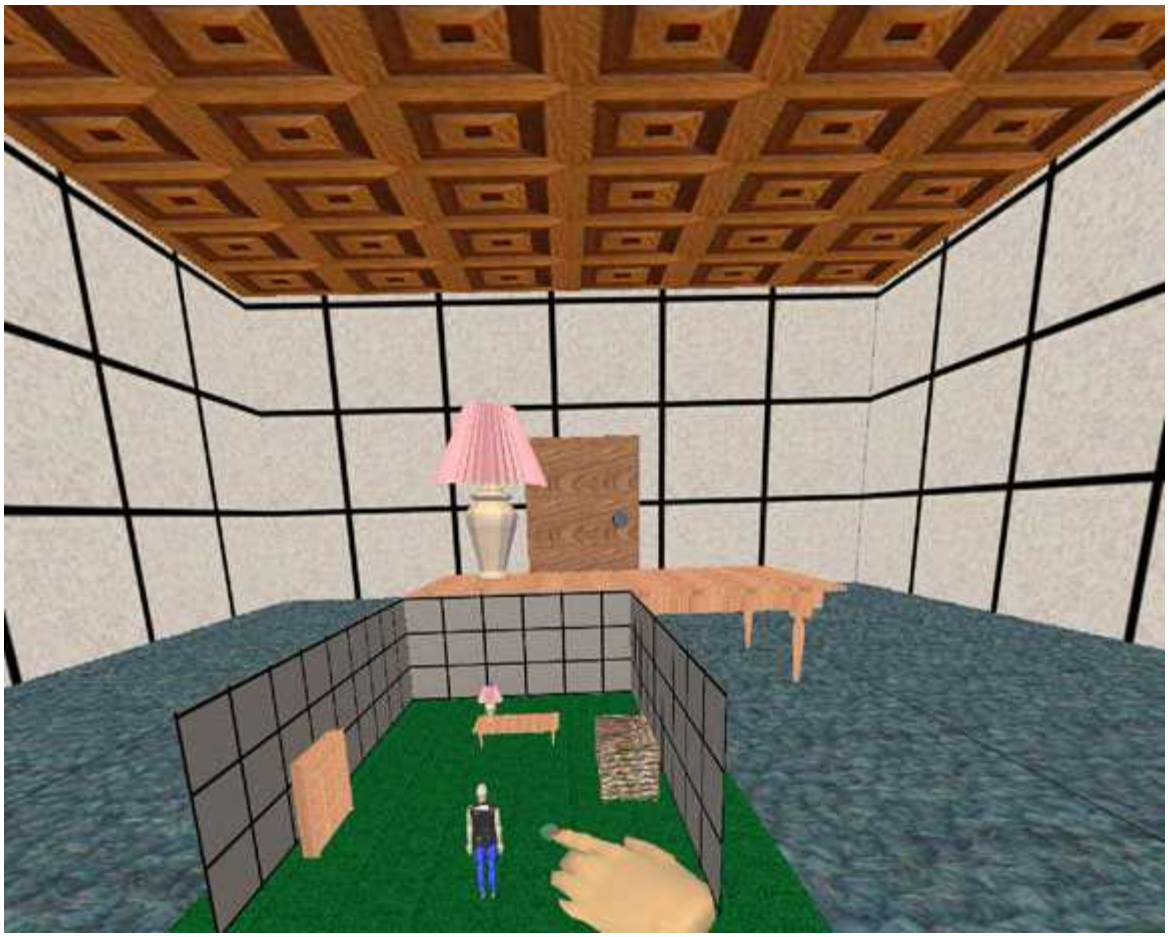

그림 4. WIM 의 예. 이 이미지는 1996 년에 만들어졌다.

WIM 기술을 실행하기 위해서, WIM 을 먼저 만들어야 한다. 이 방에 테이블 대상물이 있다고 간주하자. WIM 은 실제 방을 축소 해 놓은 것을 재연하고 있고, 그래서 가상 손이 부가될 수 있다. 테이블 오브젝트는 크기가 조종 될 필요는 없는데, 부모(WIM 방)로부터 조정된 크기를 물려 받기 때문이다. 따라서 테이블 오브젝트는 간단히 보여지는 그래프 내에서 복사될 수 있다.

WIM 에서 오브젝트는 간단한 가상 손 기술을 이용하여 선택되는데, 먼저 이 오브젝트를 상응하는 전체 크기의 오브젝트에 매치시킨다. 이 오브젝트에 포인터 리스트를 유지하는 것은 이 단계를 효율적으로 할 수 있는 방식이다. 미니어처 오브젝트는 가상 손에 붙어 있으며, 이는 간단한 가상 손 기술에서와 같은 방식이다.

미니어처 오브젝트가 조작되고 있는 동안에, 간단한 위치 매트릭스를 통해 전체 크기의 오브젝트의 위치 매트릭스를 구한다. 우리는 미니어처 오브젝트가 크기가 작은 WIM 조종 시스템에서 조작되는 것처럼 전체 크기의 월드 조종 시스템에서 같은 방식으로 움직이기를 원하기 때문에, 전체 크기 오브젝트를 제대로 움직이는 것이 필수적이다.

System Control

시스템 컨트롤은 이용자가 인터랙션 모드나 시스템 상태를 바꾸는 명령을 수행하도록 해 주는 메캐니즘을 제공한다. 명령 수행을 위하여, 이용자는 세트에서 아이템을 선택해야만 한다. 시스템 컨트롤은 넓은 범위의 주제이고 그래픽 메뉴, 몸짓 및 도구 선택기 등의 다양한 선택을 가능하게 해 주는 많은 다른 기술이 있다.

대부분의 경우, 이 기술들은 실행하기에 어렵지 않다. 대부분 선택에 관여되어 있기 때문이다. 예를 들어, 가상 메뉴 아이템은 레이 캐스팅을 이용하는데 선택된다. 모든 기술에는 좋은 비주얼 피드백이 필수적인데, 이용자가 무엇을 선택하고 있는지를 알도록 해 줄 뿐 만 아니라 언제 그것을 선택할지를 알려 줄 필요가 있기 때문이다. 이러한 부문에서, 우리는 몇 가지의 일반적인 시스템 컨트롤 기술에 대하여 주목하고자 한다.

Graphical Menus

그래픽 메뉴는 2D 메뉴와 똑같이 3D 로 보여질 수 있다. 배치는 메뉴 접근에 영향을 줄 수 고(올바른 배치는 검색을 위한 강력한 공간 인지를 제공할 수 있다), 필드의 폐쇄를 가능하게 하는데 영향을 미친다. 배치는 surround-fixed, world-fixed, display-fixed windows 로 분류될 수 있다.

배치의 하부 분류는 점도 묘연하다. World-fixed 와 surround-fixed windows 는 월드에 자유스럽게 배치되거나 또는 대상에 연결되는 메뉴로 세분화 될 수 있다.

Display-fixed windows 는 이름을 다시 설정할 수 있으며, 좀더 정확 의미를 만들 필요가 있다. 실제로 이용하고 있는 프레임을 설명하는 용어를 활용하면 더 좋을 듯 한데, 신체를 활용한다는 뜻에서 Body-centered menus 로 명명하고, 신체를 활용하거나 머리를 이용하는 등으로 활용되기 때문에 강력한 공간 인지 프레임을 제공할 수 있다.

신체 중심의 메뉴의 흥미로운 가능성은 "eyes-off" 이용인데, 이용자는 시스템 컨트롤을 메뉴를 보지 않고 수행할 수 있다. 최근의 프레임은 디바이스 중심 메뉴의 집합체 있다. 디바이스 중심 배치는 이용자가 물리적인 레퍼런스 프레임을 가질 수 있도록 제공한다(그림 5 참조).

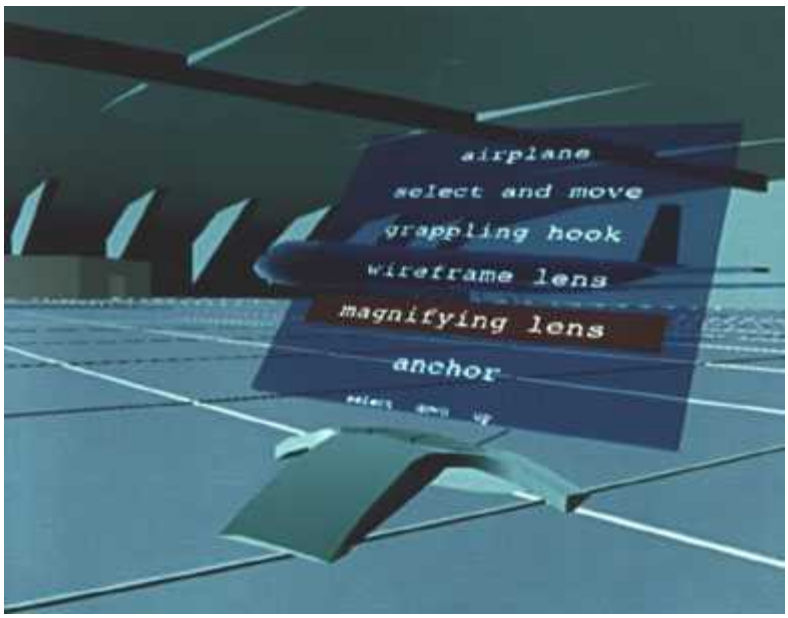

그림 5. 가상 트리코더: 디바이스 중심의 배치 그래픽 메뉴의 예시. 이 이미지는 1995 년에 만들어졌다.

우리는 그래픽 메뉴를 손 중심의 메뉴로, 전환된 2D 메뉴로 그리고 3D 위젯으로 분류할 수 있다. 손 중심 메뉴를 두 가지로 분류할 수 있는데, 1DOF 메뉴는 몇 가지 아이템을 분류하여 원형 물체를 이용하는 메뉴이다. 이용자는 하나의 축을 중심으로 손을 회전할 수 있으며, 원하는 아이템이 선택 바구니에 담길 때까지 손을 회전할 수 있다.

유저 퍼포먼스는 손과 손목의 물리적인 움직임에 매우 의존되며, 우선적인 회전 축은 조심스럽게 선택되어야 한다. 1DOF 메뉴는 몇 가지 형태로 이루어지는데, 링 메뉴, 썬다이얼, 나선 메뉴 및 회전식의 도구 선택기 등이 포함된다. 손 중심 메뉴의 두 번째 분류는 손에 들 수 있는 위젯으로, 메뉴는 신체와 관련된 자세에 저장된다.

두번째 그룹은 가장 자주 시스템 컨트롤 인터페이스로 적용되는 것이다. 전환된 2D 위젯. 이 위젯은 기본적으로 데스크 톱 환경에서와 같이 기능한다. 2D 위젯에서 아이템을 선택할 때 좀더 많은 DOF 를 처리할 수 있는 차이는 있다. 가장 대표적인 예는 당겨 내리는 메뉴, 끌어 올리는 메뉴, 나는 위젯, 툴바 및 슬라이더이다.

마지막 그래픽 메뉴의 그룹은 3D 위젯으로 알려져 있다. 3D 월드에서, 위젯은 종종 월드나 오브젝트에서 기능성을 조절하는 움직인 시스템의 의미를 가진다. 이것은 "오브젝트에 메뉴 기능성을 움직이는 것"이라는 생각을 불러일으키게 한다.

위젯을 이용하여 배치할 때 매우 중요한 문제가 있다. 3D 위젯은 이전에 논의 되었던 메뉴 기술(1DOF 나 전환된 2D 메뉴)와 달리, 다른 방식으로 기능이 맵핑된다. 기능은 오브젝트 근처에서 동일 이치에 있는 것으로 대체되고, 이에 따라 매우 상황에 민감한 "메뉴"를 구성하게 된다.

Gestures and Postures

몸짓 인터랙션을 이용할 때, 우리는 "손을 도구로 활용"하는 은유를 적용한다: 손은 문자 그대로 도구가 된다. 몸짓 인터랙션을 이용할 때, 몸짓은 초기 설정인 동시에 명령 수행의 역할을 한다.

몸짓 인터랙션에 대하여 말 할 때, 경우에 따라서 우리는 제스츄어와 자세를 의미하며, 인터랙티브 화이트 보드나 태블렛 PC 에서 사용되는 제스처 인풋은 의미하지 않는다. 제스츄어와 자세 사이에는 중요한 차이가 있는데, 자세는 정적인 움직임(꼬집기와 같은 것)인 반면, 제스처는 손의 모양과 방향의 변화를 포함한다. 제스처의 좋은 예는 수화의 사용이다.

제스처 인터랙션은 매우 강력한 시스템 컨트롤 기술이 될 수 있으며 또한 조작, 선택, 이동 등에도 유용하다. 사실, 제스처는 상대적으로 비디오 게임에 이용하게 될 때 제약이 없다. 제스처는 다른 플레이어와 의사 소통하는데 이용될 수 있으며, 롤 플레잉 게임에서 주문을 외우는데, 야구 게임에서는 사인을 보내는 것에 액션 게임에서는 공격의 조합 명령에 이용할 수 있다. 그러나, 제스처 인터랙션의 문제는 이용자가 모든 제스처를 배워야 한다는 것이다.

이용자가 일반적으로 일곱 가지 이상의 제스처를 기억할 수 없기 때문에(우리의 워킹 메모리의 제한된 수용력 때문에), 경험이 없는 이용자는 제스처 인터랙션에 심각한 문제를 가지게 된다. 특히 어플리케이션이 복잡하고나 많은 양의 제스처를 필요로 하게 되면 더욱 그렇다.

이용자는 종종 제스처 인터랙션을 이용할 때 그래픽 메뉴에 집중하지 않는다. 가능한 제스처의 구조는 완전히 보이지 않는다. 제스처 인터랙션을 덜 친숙한 이용자가 사용하기 쉽도록 만들기 위해서, 명령을 수행한 이후에 시각적인 실마리를 주는 것처럼 강력한 피드백이 필요하다.

Tools

우리는 두 가지 다른 정류의 툴을 정의할 수 있는데, 물리적 툴과 가상의 툴로 명명할 수 있다. 물리적 툴은 상황에 민감한 인풋 디바이스이고, 소도구 라고 불리기도 한다. 소도구는 가상 현실에서 복제되는 현실의 대상물이다.

물리적 툴은 복합 공간 또는 복합 시간이 될 수 있는데, 툴이 (일반적인 데스크 톱 마마우스처럼)한번에 다양한 기능을 수행할 수 있다. 간단히 손만 뻗음으로써 물리적 툴에 접근할 수 있고, 또는 인풋 디바이스 그 자체에 모드를 바꾸어 접근할 수 있다.

가상 툴은 툴 벨트가 전형적인 예인 툴이다. 이용자는 가상의 툴 벨트를 허리에 매고, 이용자가 특정한 장소에 벨트를 움켜 쥠으로써 특정한 기능에 접근할 수 있게 된다. 가상 툴 벨트는 잠재적으로 1 인칭 및 3 인칭 슈팅, 롤 플레잉 게임 및 액션/어드벤처 게임에서 이용될 수 있으며 아이템과 무기를 어떻게 저장하는지에 대한 구조를 개혁할 수 있다.

때로, 툴 벨트의 기능은 그래픽 메뉴와 같은 원리를 통해 접근되기도 하는데, 메뉴 그 자치를 보는 것에 집중하도록 한다. 툴의 구조는 그렇게 복잡하지 않다. 일전에 언급했듯이, 물리적 툴은 특정한 용도의 디바이스이거나 여러 개의 기능을 하나의 툴로 이용할 수 있도록 하고 있다. 물리적 툴은 그래픽 메뉴를 위한 디스플레이 매체이다. 이 경우에, 이것은 그래픽 메뉴와 같은 방식으로 개발되어야 한다. 가상 툴은 구조화를 위한 자기수용적 신호를 이용한다.

결론

이 글에서 내가 언급한 기술은 수년 동안 3D 유저 인터페이스 연구와 가상 현실에서 다루어진 여러 가지에 대하여 살짝 건드리기만 했을 정도이다. 비디오 게임 산업이 점점 모션 중심의 인터페이스를 활용한 게임으로 발전해 가면서, 이 영역에서의 연구는 점점 중요해 지고 있다.

나는 비디오 게임 산업이 학술적으로 기술의 과잉 이라는 관점을 제공하며 이것을 이용하는 방법에 대한 교훈을 얻게 될 것이라고 생각한다. 비디오 게임의 인기만큼, 가상 현실 및 3D 유저 인터페이스 영역에서의 학술이 계속 탐구되고 새로운 인터페이스 기술을 개발하여 게임에 이용될 수 있도록 되기를 바란다. 게임 산업과 학계가 함께 작업하여 게임 플레이 메캐닉과 게임 인터페이스를 격상시키는 공생적인 관계가 되기를 바란다.

Reading List

Here is a short reading list for anyone interested in learning about work academics have done with 3D spatial interaction.

Bott, J., Crowley, J., and LaViola, J. "Exploring 3D Gestural Interfaces for Music Creation in Video Games", *Proceedings of The Fourth International Conference on the Foundations of Digital Games 2009*, 18-25, April 2009.

Bowman, D., Kruijff, E., LaViola, J., and Poupyrev, I. *3D User Interfaces: Theory and Practice*, Addison Wesley, July 2004.

Charbonneau, E., Miller, A., Wingrave, C., and LaViola, J. "Understanding Visual Interfaces for the Next Generation of Dance-Based Rhythm Video Games", *Proceedings of Sandbox 2009: The Fourth ACM SIGGRAPH Conference on Video Games*, 119-126. August 2009.

Chertoff, D., Byers, R., and LaViola, J. "An Exploration of Menu Techniques using a 3D Game Input Device", *Proceedings of The Fourth International Conference on the Foundations of Digital Games 2009*, 256-263, April 2009.

Wingrave, C., Williamson, B. , Varcholik, P., Rose, J., Miller, A., Charbonneau, E., Bott, J. and LaViola, J. "Wii Remote and Beyond: Using Spatially Convenient Devices for 3DUIs", *IEEE Computer Graphics and Applications*, 30(2):71-85, March/April 2010.