

웹기반 다차원 민군겸용 인터랙티브 디지털 콘텐츠의 효과적인 구현 방안

강석훈* · 김대청**

목 차

- 1 서론
2. 관련 연구
- 3 이벤트 드리븐 설계방식
4. 실제 구현 사례 연구
- 5 결론

1. 서론

현대의 발달된 첨단기술력 아래 수많은 제품들이 물밀듯이 쏟아져 나오고 있으며 제품이 어떠한 특징을 가지고 있고, 어떠한 기능을 가지고 있는지 알기란 무척 어렵다. 이러한 상황 속에서 전자 카탈로그의 등장은 가히 획기적이라 할만하다 전자 카탈로그란 상품 또는 기업에 대한 광고가 전자적 파일의 형태로 제작되어 인터

* 대전대학교 컴퓨터공학과 교수

** 대전대학교 컴퓨터공학과

넷을 통하여 전 세계적으로 서비스가 가능한 홈페이지를 말한다. 전자 카탈로그는 상품에 대한 간략한 소개, 사진(동영상 등), 제작업체 URL, 연락처 등 기존의 인쇄물 형태의 카탈로그에 비하여 많은 내용으로 구성된다 이러한 전자 카탈로그의 범주에 속하는 액티브 카탈로그(Active Catalog)는 실제 제품과 동일한 조작과 반응, 형태로 제품을 설명해 소비자들이 PC상에서 실제로 제품을 사용하는 감각으로 제품의 특성을 파악할 수 있는 디지털 콘텐츠다 특히 전자제품이나 기계장비의 UI(User Interface)개발에서 판매, AS(After Service)에 이르는 일련의 기업 활동에 필요한 UI 사양서 작성, UI 기능성 평가, 사용자 매뉴얼 작성, 홍보용 디지털 카탈로그 작성, AS매뉴얼 작성 등의 업무를 유기적으로 통합하고 완벽하게 디지털화하여 기존에 상존했던 비효율성 및 중복투자를 해소할 수 있는 장점을 적극 활용한다. 또한 각 업무 단계에서 발생했던 커뮤니케이션상의 오류를 제거함은 효과적인 개발절차가 된다

본 논문은 FSM(Finite State Machine) 기반의 HMI (Human Machine Interface) 모델링 및 시뮬레이션 저작도구를 활용하여 군용목적으로 다양한 기능성을 보유한 전자장비, 각종, 기계, 설비 등의 제품기능을 가상으로 조작할 수 있는 인터랙티브 멀티미디어 콘텐츠를 제작함에 있어 PlayMo를 이용한 효과적인 구현방안을 제시하며 구성은 다음과 같다 2절에서는 PlayMo의 특징과 기존의 저작도구와 비교·분석해본다. 3절에서는 이벤트 드리븐(Event Driven) 설계방식에 대해서 기술하고, 4절에서는 PlayMo 2D, 3D 실제 구현 사례를 연구해본다 마지막으로 5절에서는 결론 및 향후 연구방향을 제시한다.

2. 관련연구

현재 인터넷은 디자인적 요소와 사용자의 편리함을 내세운 저작 도구들이 많으며 각자 개발자와 사용자에게 이로운 방향으로 발전을 하고 있다. 이번 장에서는 이러한 툴들 중에서 PlayMo가 가지는 이점과 다른 저작도구들과 어떤 특징이 있는지 비교분석을 해본다

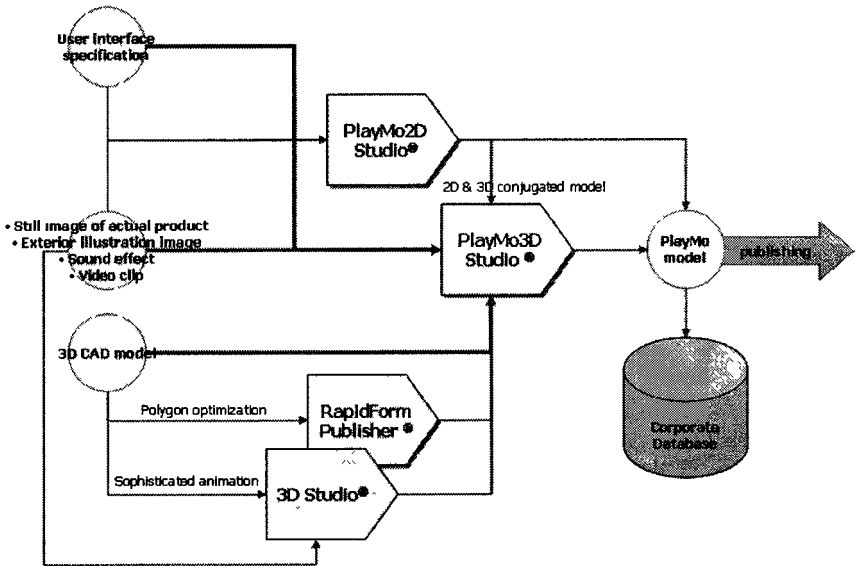
2.1 PlayMo저작도구의 특징

2.1.1 효과적이면서 손쉬운 디자인

PlayMo는 다른 코딩이나 프로그래밍이 필요 없는 위치쥐그(WYSIWYG)기능을 제공한다. 단지 화면상에서 드래그 앤 드랍(Drag and Drop)으로 이벤트를 만들며, 각종 멀티미디어 소스를 선택하여 추가시킨다 이것은 인터랙티브 모델링 경험이 없는 초보자에게 더 쉽고, 빠르고, 효과적인 인터랙티브 모델을 만들 수 있게 해준다[1].

2.1.2 애플리케이션의 통합 환경 제공

그림 1과 같이 PlayMo는 유사 애플리케이션의 통합 환경을 제공해 준다. 이는 2D와 3D 모두를 포함하여 개발되어진 결과물을 PlayMo Player로 사용자에게 브라우저상으로 웹이나 로컬플레이어로 사용자의 PC상에서 결과물을 확인할 수 있게 해준다 이러한 통합 환경을 위해 제공하는 기능으로는 시나리오 기반의 오토 시뮬레이션 모델, 자동 HTML코드 생성, 자동 온라인 도움말 생성, 웹 출판 기능이 있다 그리고 브라우저상에서 사용자와 상호 교류가 가능하도록 Active X 컨트롤과 Java Script 기능을 제공한다[2].



<그림 1> 애플리케이션의 통합 환경 제공

2.2 기존의 방식과의 비교

PlayMo의 대표적인 장점은 단순히 보여주는 것이 아니라 2D, 3D 형태의 실제적인 이미지를 보여준다는 것이다. 이벤트 드리븐(Event Driven) 형식의 HMI(Human Machine Interface)모델링 방법을 통하여 더욱 편리한 작업환경에서 실 제품과 동일한 느낌의 모양을 표현할 수 있다. PlayMo는 Event를 사용하여 모든 기능을 동일하게 구현하는 2D 시뮬레이션을 작성하는 것은 물론이고, 360도 회전 가능한 3D 시뮬레이션까지도 작성이 가능하다.

기존의 관련 소프트웨어로서 대체로 많이 사용되고 있는 Flash나 Director, 3D프로그램들에 비해 개발환경이 개선되어 초보자도 손쉽게 작업할 수 있으며, 압축되는 데이터 크기가 작아 모델링 시간이

줄어들게 되며 그만큼 실행 시간 또한 줄어들게 된다 그림 2를 보면 다른 그래픽 틀에 비해 데이터 크기는 작지만 여러 가지 다양한 기능을 발휘한다는 사실을 알 수 있다. 플래시의 경우 가장 많이 쓰이는 이유는 강력한 인터랙티브의 고품질 애니메이션과 파일의 경량화 때문이다 마찬가지로 PlayMo 또한 이미지를 포함하는 압축 포맷을 지원한다.

정보를 전달하는 가장 효과적인 방법 중의 하나는 사용자와의 상호작용이다 다른 저작도구보다도 우수한 기능을 자랑하는, 플래시는 변수의 사용과 무비 클립의 속성을 이용한 방문자와의 상호작용이 더욱 강력해졌다 PlayMo의 경우 더욱더 사용하기 쉽고 접하기 쉽도록 많은 이벤트들이 추가되었다 플래시의 장점은 그대로 사용하면서 더욱 강력하고 빠른 기능들을 추가시킨 것이다.

인터넷을 사용하는데 있어 속도문제는 홈페이지를 제작할 때 필수적으로 고려되어야만 한다. 아무리 화려한 테크닉을 자랑하는 홈페이지라 할지라도 방문자를 한없이 기다리게 한다면 곧 외면당하게 된다 플래시에서 만들어진 무비는 그 결과물의 탁월함에 비해 크기가 아주 작다. 파일 전체를 다운받지 않은 상태에서도 실시간 재생이 가능하며, 사운드 또한 MP3로 압축해서 내보내기 때문에 음질이 거의 손상되지 않은 상태에서 파일을 최적화 할 수 있다 PlayMo의 경우 플래시의 용량보다 훨씬 작은 데이터 크기를 제공함으로써 더욱 빠른 접속 속도와 바로 재생되는 느낌을 주어 쇼핑몰이나 카탈로그 사용자로부터 원활한 서비스를 제공할 수 있다[3]

<표 1> 기존 저작도구와의 기능 비교

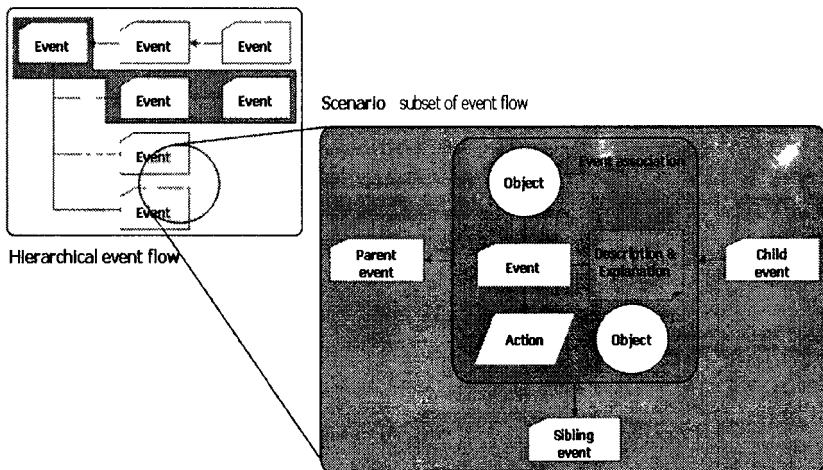
종류 기능비교	PlayMo	Java	Director	Web3D	Flash
프로그래밍 필요성	No	Yes	Yes	No	No
기능성 구현 난이도	Very Easy	Hard	Very Hard	Impossible	Medium
제작기간	Short	Very Long	Long	Long	Medium
데이터 크기	Very Small	Large	Small	Medium	Medium
오브젝트 통신	Yes	Yes	No	No	No
멀티미디어 지원	High	High	Medium	Medium	High
3D 지원	Yes	Yes	No	Yes	예정
자동메뉴얼 생성	Yes	No	No	No	No
실행 로그 생성	Yes	No	No	No	No
확장성	High	High	Low	Low	Low
동작모델	Event Driven	Event Driven	Time Driven	Event Driven	Time Driven
주요응용	전자메뉴얼 CBT MMI 설계 평가	프로그래밍	CD 타이틀 등	웹상의 3차원 비주얼라이제 이션	벡터 이미 지 기반의 동적 웹페 이지 작업

3. 이벤트 드리븐 설계방식

이벤트 드리븐(Event Driven)은 여러 프로그래밍 언어와 디자인 저작도구에서 사용되고 있는 방식으로 이번 장에서는 이벤트 드리븐에 대해서 살펴보고자 한다

3.1 이벤트 드리븐 모델링

이벤트 드리븐(Event Driven) 모델링(Modeling)은 비주얼적인 이벤트 트리를 사용하여 흐름도로 구성된다 이벤트 컨트롤은 명령의 우선순위와 계층에 의해 구체적으로 정의된다 그림 2와 같이 이벤트 트리는 이벤트 노드들 간의 우선순위와 명령이 상호 연관적으로 연결되어 있다 최고 상위 레벨의 루트로부터 시작된 이벤트는 부모 이벤트와 형제 이벤트, 그리고 자식 이벤트로 명령이 상호 연관되어 흘러가는 것을 볼 수 있다[1].

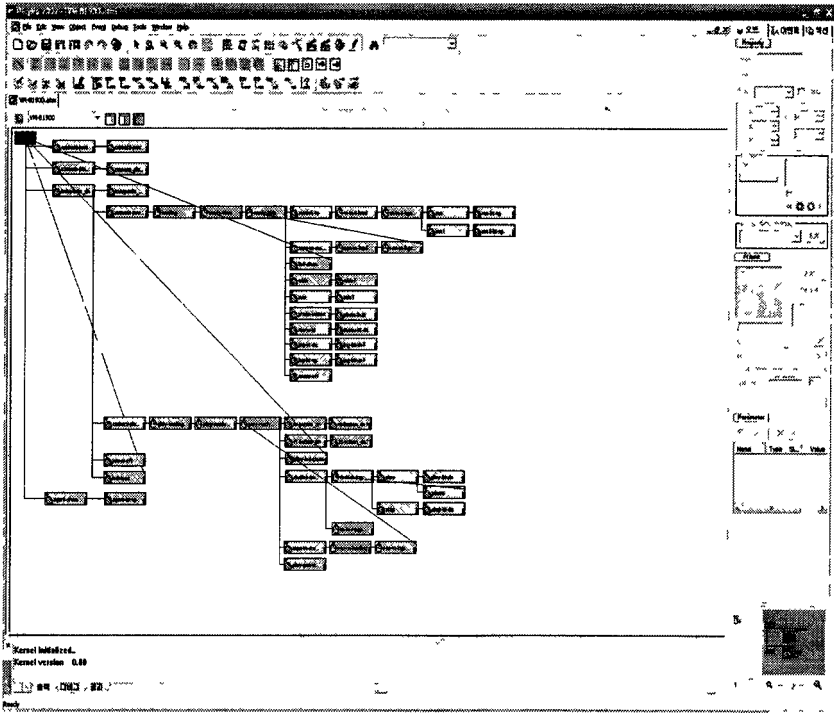


<그림 2> 이벤트 드리븐 모델링

3.2 이벤트 트리

3.2.1 정의

그림 3과 같이 PlayMo에서의 이벤트 트리는 구현하려는 기능들의 위계 또는 순서를 각 이벤트 노드들 간의 연결고리를 통해 적용시킨 모양이다 즉, 결과물을 시뮬레이션(Simulation)할 때 우선적으로 적용시켜야할 명령과 그 뒤에 있어야 적용이 가능한 명령들을 트리 형태의 계층 구조상에서 제 위치에 있도록 구성한 것이다.



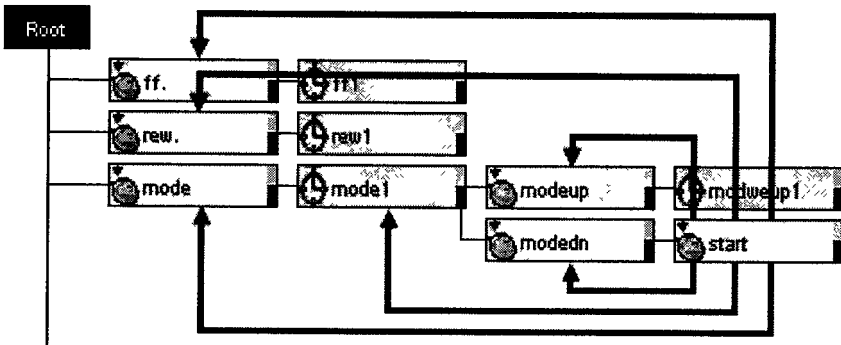
<그림 3> PlayMo의 이벤트 윈도우

3.2.2 이벤트 흐름의 규칙

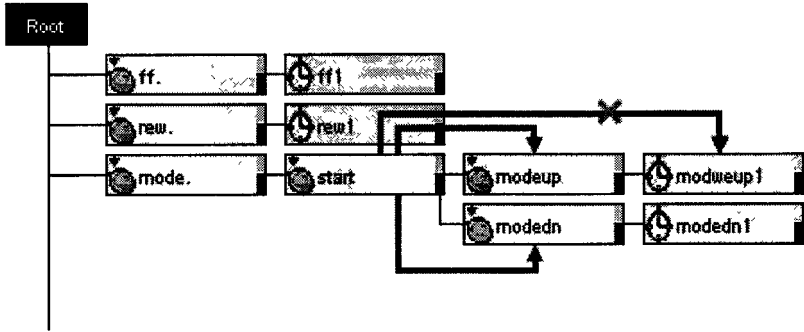
하나의 이벤트 위치에서 다음 이벤트 위치를 수행할 경우에 수행이 되어야 하는 이벤트와 수행이 되지 않아야 하는 이벤트가 있다. 이러한 물리적 논리에 의해 PlayMo에서는 표 2와 같이 몇 가지의 이벤트 흐름의 규칙을 제시하고 있으며 그림 4, 5, 6에서 더 구체적으로 나타내고 있다[3]

<표 2> 이벤트 흐름 규칙

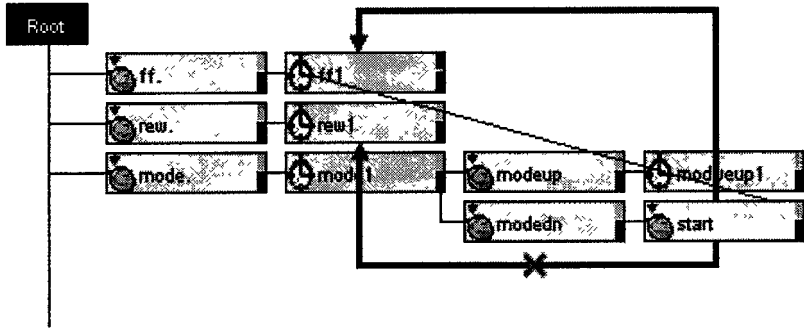
1.	한 단계 하위 이벤트 노드로 연결 가능
2.	동등 계층의 이벤트 노드들로 연결 가능.
3.	여러 단계의 상위 이벤트 노드로 연결 가능.
4.	두 단계 이상 하위 이벤트 노드로는 연결 불가능
5.	동등 계층 또는 상위 이벤트 노드에서 또 다른 하위 이벤트 노드로 꺾인 부분으로는 연결 불가능



<그림 4> 부모와 형제 노드만 검색



<그림 5> 최우선 검색 노드



<그림 6> 형제 노드 검색

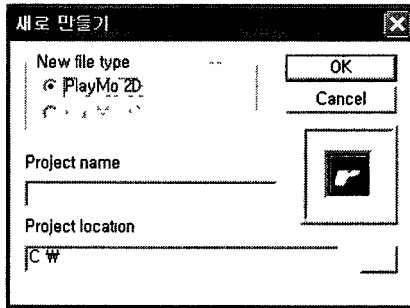
4. 실제 구현 사례 연구

4.1 2차원 인터랙티브 디지털콘텐츠의 저작 과정

디지털 캠코더를 가지고 2D 구현 과정을 설명한다

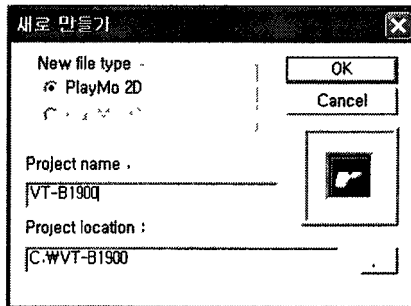
4.1.1 New Project

메뉴에서 File > New 또는 아이콘을 클릭하면, 그림 7과 같이 New Project 설정 window가 나타난다 New file type에서 PlayMo 2D에 체크가 되어있는지를 확인을 하고 Project Location에 저장할 디렉터리를 지정한다



<그림 7> 새로 만들기

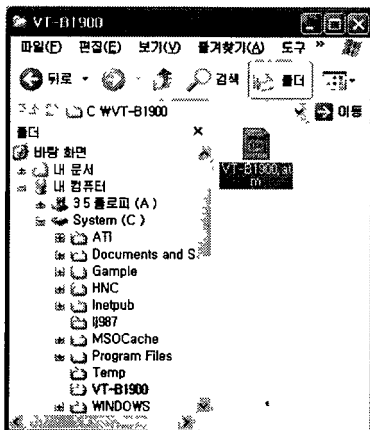
그림 8과 같이 프로젝트 이름을 적고 OK버튼을 누른다.



<그림 8> 프로젝트 이름 생성



지정한 디렉터리가 생성되고, 그 디렉터리 안에 VT-B1900이라는

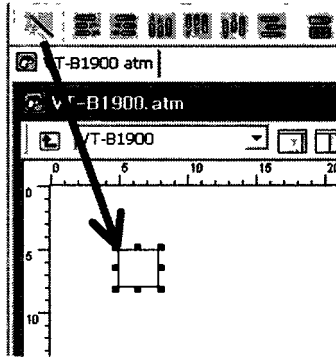
atm확장자 파일이 생성된다



<그림 9> 프로젝트 파일 생성

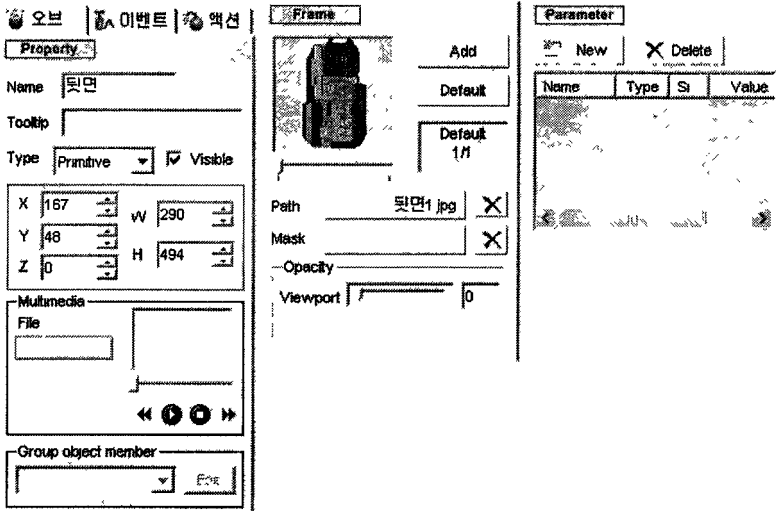
4.1.2 Object Loading

새롭게 생성된 작업 공간 중 그림 10과 같은 Object Window에서 새로운 object를 만들기 위해  (New Object)를 마우스로 클릭하여 작업 공간 안쪽으로 끌어 놓는다 새로 만들어진 Object는 속성이 정해질 때까지  모양으로 되어있다.



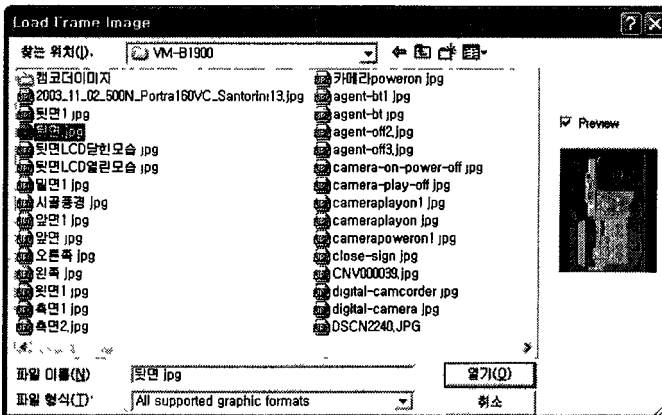
<그림 10> 오브젝트 생성

그림 11과 같이 Object의 속성은 Event Tree Window의 오른쪽에 있는 Object Panel에서 설정할 수 있다. Object Property에서 Name을 '뒷면'이라고 적다. Type은 기본적으로 Primitive라 지정되어 있는데 Image object는 primitive에 그대로 두면 된다. 옆의 Visible에는 체크를 한다. 이를 체크하면 Simulator에서 결과를 확인할 때 처음부터 보이는 Object로서 출발한다 만약 체크를 하지 않으면 처음에는 보이지 않고 Action의 ShowObj 명령을 통해 보이도록 해야만 보이게 된다.



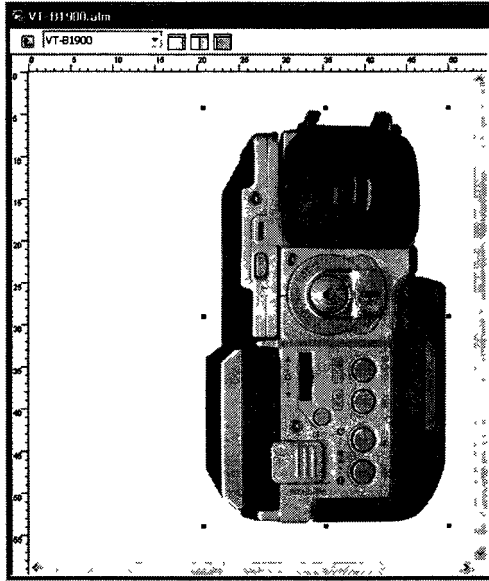
<그림 11> 오브젝트 속성

이번에는 그림 12와 같이 image를 넣도록 한다. 아래쪽 Object frame에서 이 작업을 수행한다. **Add**를 클릭하고 Sample/image/뒷면.jpg를 선택한다



<그림 12> 이미지 삽입

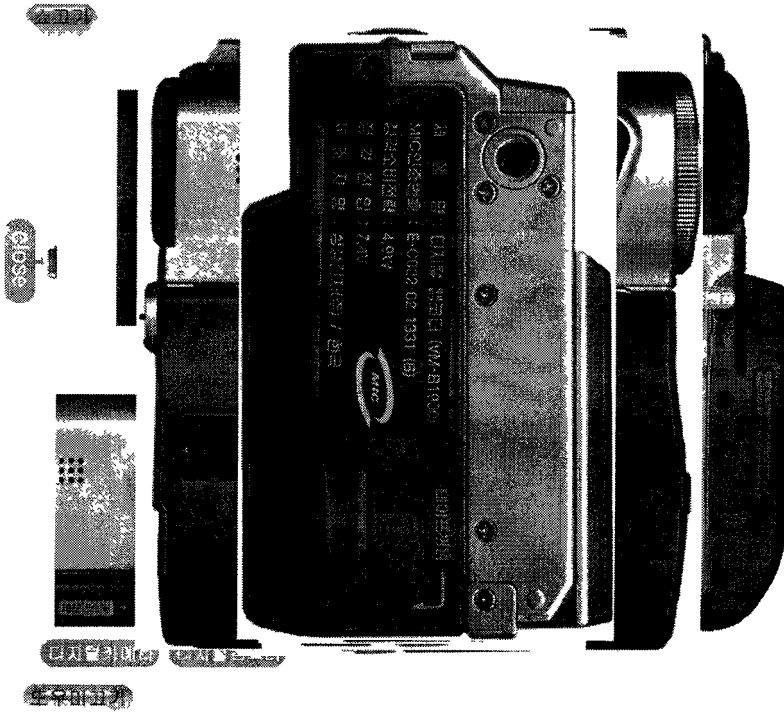
그림 13과 같이 Object window에서 Loading된 image를 볼 수 있다 그 Object를 적절한 위치에 배치한다.



<그림 13> 위치 지정


Act를 클릭해서 이미지를 불러온 뒤 다시 한 번 눌러서 다른 이미지를 부르면 두 번째 frame에 자리를 잡게 된다 또 다른 이미지를 부르면 세 번째 Frame으로 들어간다 이런 형식으로 여러 개의 이미지를 하나의 object에 불러들인 뒤 나중에 Action을 통해 상황에 따라 Frame이 바뀌게 설정하여 이미지 변화의 효과를 낼 수가 있다.

다음 그림 14는 전체 Object들이 정리된 Object Window이다



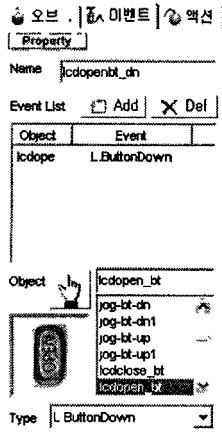
<그림 14> 오브젝트를 배치한 모습

4.1.3 Event Node 생성

Object Window에 정렬된 object들에 임의의 명령 신호를 보내주고(event) 그에 따른 반응(action)을 유도하는 첫 번째 작업으로 Event Tree에 event node를 생성하도록 한다 Event tree 위의  아이콘을 Event Tree window로 끌어내리면 새로운 event node가 생성된다. 생성된 event node에 왼쪽의 event panel과 action panel을 사용하여 적절한 기능성을 부여한다.

첫 event node는 캠코더의 LCD를 여는 것부터 시작이 된다.

New event node를 클릭하여 활성화 시키고 그림 13과 같이 오른쪽 Event panel로 간다. 우선 name을 'lcdopenbt_dn'이라 넣는다. 이 이벤트 이름도 상당히 중요하다 나중에 event node가 상당히 많아지면 event tree상에서 그 이름으로 기능 파악을 할 수 있어야 하기 때문이다

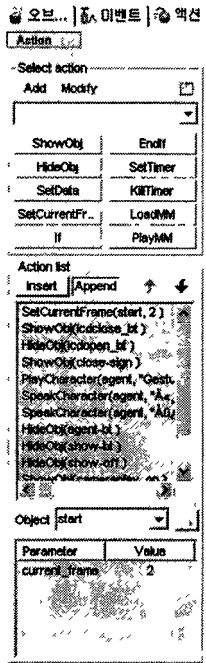


<그림 15> 이벤트 지정

Event List 옆의 Add 버튼을 클릭한다 그러면 임의의 object 이름과 event 이름이 나올 것이다. 그 아래 object List에서 lcdopen_bt인 object를 선택한다 우리가 가상으로 만들어 놓은 버튼인 lcdopen_bt 가 그것이다 그 이름을 더블클릭하면 위 event List의 내용이 바뀌게 된다 그 object를 마우스 왼쪽 버튼으로 누를 것이므로 L button down이 맞는 Event이다. 이렇게 설정을 하고 나면 Event Tree 상의 New Node는 마우스 이벤트 node로 모양이 바뀌게 된다.

이제 Action을 지정할 차례이다 위에 설정한 event를 발생시키

면(즉, lcdopen_bt 를 사용자가 마우스 왼쪽 버튼으로 클릭하면) 수행될 결과를 지정하는 것이다. 물론 캡코더의 LCD가 열려지는 것이다





<그림 16> 액션 지정

우선 LCD가 열리면서 모양이 바뀌는 형태부터 만들도록 한다. start에 여러 이미지를 frame으로 지정했었다. 그중 두 번째 이미지, 즉 2번 frame으로 변화를 시킨다.


Action List에서 add 버튼을 클릭하고 object를 start로 선택하고, Action type을 Frame의 SetCurrentFrame으로 지정한다. SetCurrentFrame이 바로 Object의 Frame을 지정하는 명령이며 아마도 가장 많이 쓰는 명령이 될 것이다 그러면 아래 parameter에

current_frame이라는 옵션이 '0'이라는 value값을 가지고 나타날 것이다. 그 Value값을 '2'로 바꾸면 된다. 이것으로 start는 2번 frame으로 바뀌게 된다

LCD가 열렸으므로 LCD를 닫는 버튼을 생성하도록 한다. 자주 쓰는 액션은 따로 버튼으로 만들어져 있다. ShowObj도 자주 쓰는 액션중 하나로 그 버튼을 눌러준다. 그리고 object를 'lcdclose_bt'을 선택한다 똑같은 방법으로 보여줄 오브젝트와 화면에서 없어져야 할 오브젝트들을 추가해준다

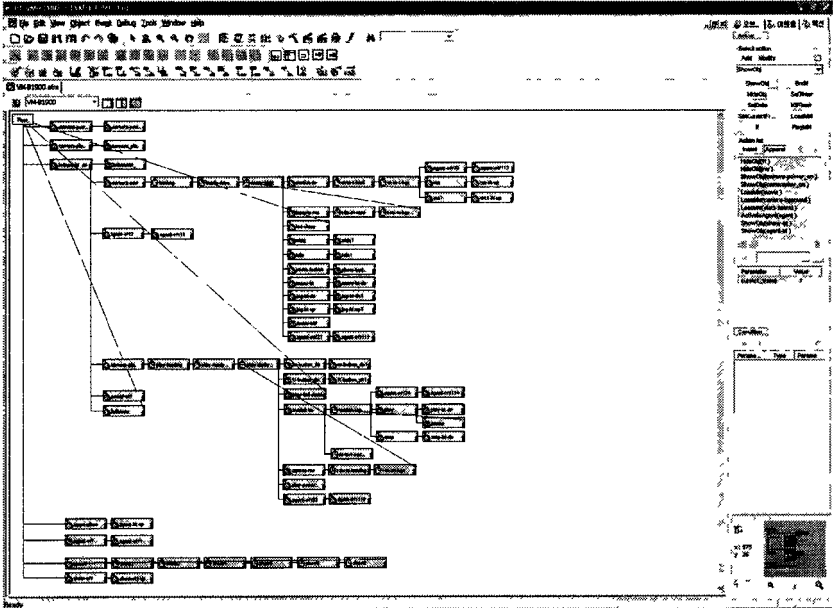
이제 또 하나의 event node를 만든다. lcdopenbt_dn 노드를 활성화 시키고 를 클릭하면 lcdopenbt_dn 바로 뒤쪽으로 새로운 event node가 연결된 상태로 생성된다. 만일 lcdopenbt_dn의 바로 뒤로 연결이 되어있지 않다면 lcdopenbt_dn의 오른쪽 아래에 검정색 박스 를 새로 만든 node로 Drag&Drop하면 검은 선이 따라오게 된다. 그것을 새 event node위에서 마우스버튼을 놓으면 연결라인이 생성된다

오른쪽 event panel에서 새로 만든 node의 이름을 'lcdopenbt_hide'이라 입력하고, event type을 L_ButtonUp으로, object를 lcdopen_bt으로 설정한다

이제  (simulator)를 열고 결과를 확인한다. 그러면 power를 켜는 때의 동작을 확인할 수 있을 것이다.

4.1.4 Event Tree 구조

그림 17은 전체 이벤트 트리 모습이다. root에서 시작을 하고, LCD를 여는 이벤트와 닫는 이벤트, 파워를 켜고 끄는 이벤트 그리고 캠코더와 디지털 카메라 기능, 도우미 기능, 전체 모습을 애니메이션 하는 기능, 찍은 사진과 영상을 보는 이벤트들로 구성되어 있다



<그림 17> 전체 이벤트 트리 모습

그림 18은 완성된 캠코더를 시뮬레이션하고 있는 모습이다. 구현된 결과물을 브라우저상에서 보려면 PlayMo의 기능인 Publish HTML 로 브라우저 출판 기능이 있다 이 기능으로 브라우저상에 삽입을 하여 인터넷 상에서 볼 수 있다



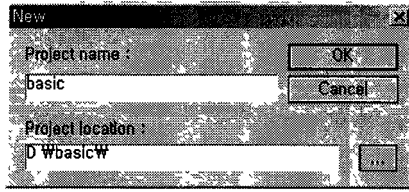
<그림 18> 캠코더를 시뮬레이션 하는 모습

4.2 3차원 인터랙티브 디지털콘텐츠의 저작 과정

3D 구현 사례로써 기본 파이프를 설명한다.

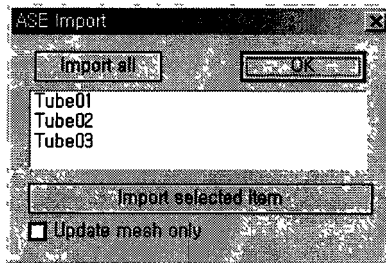
4.2.1 New Project

그림 19와 같이 File/New를 클릭하고 Basic이란 프로젝트를 새로 만든다



<그림 19> New Project

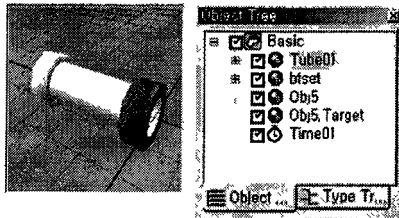
그림 20과 같이 화면이 구성되면 File/import/Ase를 선택하고 ../Example/Basic_Src의 pipe ase를 선택한다 그리고 Import All을 클릭한다.



<그림 20> ase 파일 import

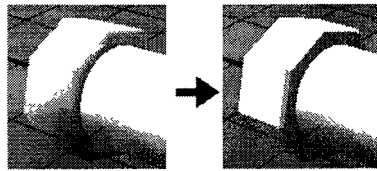
화면을 그림 21과 같이 구성한다.

- Rotate View : Shift + Mouse 이동
- Pan View . Ctrl + Mouse 이동
- Zoom View : Mouse Center Wheel 이동





<그림 21> 오브젝트 구성

그림 22와 같이 Tube01과 Tube03을 보면 모서리가 뾰족하게 스무딩(smoothing) 처리가 되어 있는 것을 볼 수 있다 그 모서리들을 살리기 위해서는 오른쪽 오브젝트 패널에서 Crease angle의 수치를 바꾸어 주어야한다. 이 수치는 스무딩이 적용될 최소 각도로서 초기 값은 110으로 되어 있을 것이다. 이 값을 170으로 바꾸어 준다.



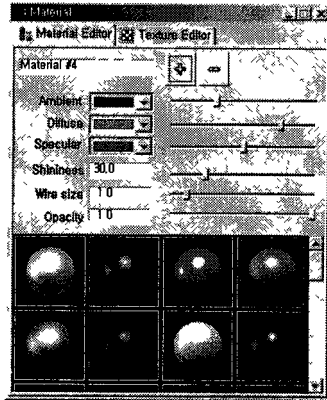
<그림 22> 스무딩(smoothing) 처리 수치 변경

4.2.2 Material / Texture 생성 및 적용

아이콘 바에서 Material Editor Icon을  클릭한다. New Material 을 클릭하여 새 모델을 생성하고 그림 23과 같이 설정한다.

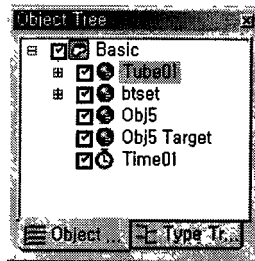
- Ambient는 오브젝트의 빛을 받지 못하는 어두운 부분의 색상이다.
- Diffuse는 오브젝트 본래의 색상이다.
- Specular은 오브젝트의 빛을 받는 밝은 부분의 색상인데 주로 재질이 번들거리느냐 둔탁하냐를 결정짓는데 쓰인다
- 위 3가지 색상 메뉴 옆의 바는 그 색상의 명도를 조절해 준다. 기본적으로 Diffuse의 색상을 먼저 정해주고 나머지도 같은 색으로 지정한 후 Ambient는 좀 더 어둡게 해주고, Specular은 좀 더 밝게 해주면 실제와 유사한 색감을 표현할 수 있다.
- Shininess는 빛을 받는 영역의 크기를 결정한다

- Wire Size는 라인과 같은 형상 오브젝트(Shape Object)에 두께를 준다.
- Opacity는 투명도를 결정한다.



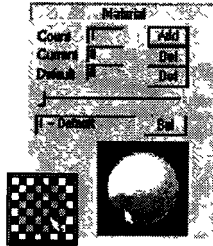
<그림 23> Material/Texture edit window

위에서 만든 머티리얼(Material)을 Tube02에 적용시키기 위해서 Tube02를 선택하고 오른쪽 오브젝트 머티리얼 패널(Object Material Panel)을 선택한다 오브젝트를 선택할 때 그림 24와 같이 오브젝트 트리(Object Tree)를 열고 선택하면 편리하다.



<그림 24> 머티리얼(Material)을 Tube02에 적용

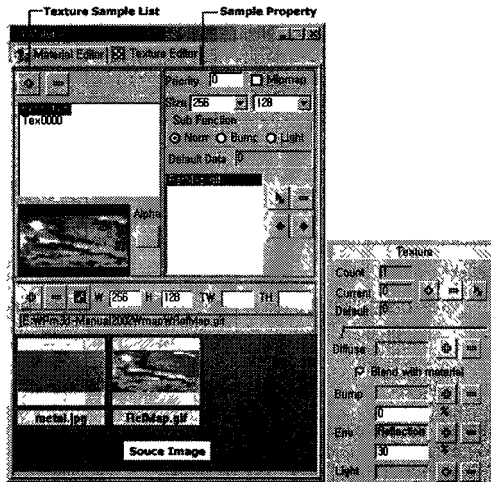
Add 버튼을 클릭하고 1-Default를 선택한다. Material Editor에서 Tube02를 선택한 뒤, 오브젝트 머티리얼 패널의 구 이미지 쪽으로 드래그 앤 드롭 한다 한 오브젝트에 대해 머티리얼은 여러 개를 지정해 줄 수 있다. Add 버튼을 눌러 원하는 개수만큼 생성한다 그리고 액션에서 3DsetObjMaterial로 상황에 따라 그 머티리얼을 바꿀 수 있도록 한다



<그림 25> 머티리얼(Material) 지정

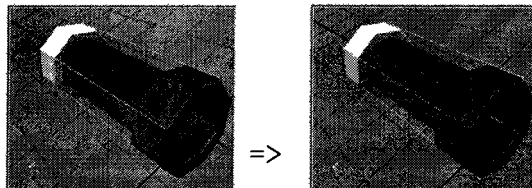
이번엔 Tube02에 텍스처(Texture)를 입히도록 한다.

- Texture Editor를 열고 Texture Sample List의 Add 버튼을 누른다.
- 이름을 Reflection이라 정한다
- 아래 Source Image 윈도우의 Add 버튼을 누르고 /Basic_Map/ RefMap gif를 불러온다.
- Reflection을 선택하고 아래 RefMap gif를 오른쪽 데이터 리스트로 드래그 앤 드롭 한다.
- Sub Function은 Environment Map으로 쓸 것이므로 Normal로 한다
- 크기는 그림 26과 같이 소스의 크기와 동일하게 조정을 해준다.



<그림 26> 텍스처(Texture) 입히기

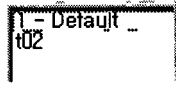
Tube02를 선택하고 오른쪽 오브젝트 머티리얼 패널을 선택한다. Texture 탭에서 Add 버튼을 누른다. 그 다음, 앞서 Material Editor의 Texture Sample List에서 만든 Reflection 텍스처를 선택하여 Env 뒤의 빈 칸으로 드래그 앤 드롭 한다 그리고 강도 수치는 30%로 한다.



<그림 27> 강도 수치를 30%로 변환

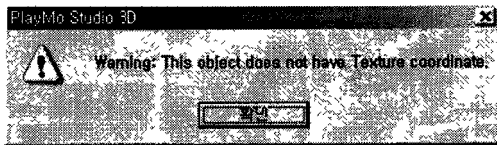
Tube03에도 머티리얼을 입혀보자. Tube03을 선택하고 오른쪽 오브젝트 머티리얼 패널에서 맨 위 Add 버튼을 누르고 그림 28과 같

이 나온 리스트에서 1-Default을 선택한다.




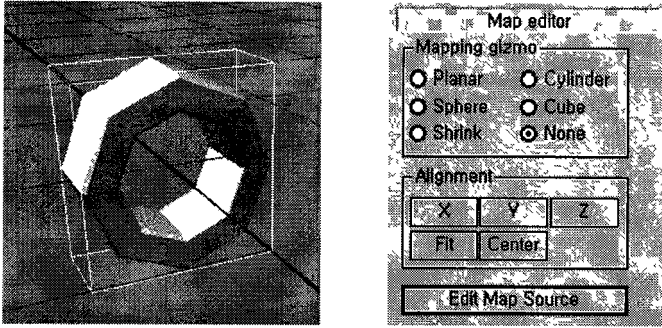
<그림 28> Tube03에 머티리얼 변환

Texture 탭의 Add버튼을 누르면 그림 29와 같은 경고 메시지가 뜰 것이다. 이는 원본 파일인 Ase 파일에서 Tube03에 대한 텍스처 맵핑 좌표계(Texture mapping coordinate)를 설정하지 않았다는 말이다. 3D 모델링 툴에서 UvwMap을 설정하지 않은 것이다. 그렇다면 PlayMo3D에서 직접 설정을 해주도록 한다.



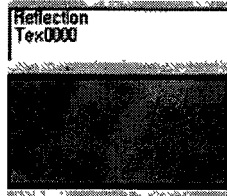
<그림 29> 경고 메시지

아이콘 바에서 Edit Map 아이콘  누르면 Edit Map 탭이 활성화되면서 뷰포트에는 그림 30과 같이 Tube03만 남고 다 사라질 것이다. Edit Map에서 Cylinder를 체크한다 그러면 Tube03에는 노란 실린더 모양의 텍스트 맵핑 좌표계가 생성될 것이다.




<그림 30> Edit Map

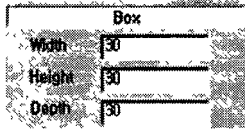
Texture Editor에서 그림 31과 같이 Tex0000을 오브젝트 머티리얼 패널의 Texture 탭에 있는 Diffuse 옆 빈칸으로 드래그 앤 드롭한다. 이제 모든 머티리얼 설정은 끝났다.



<그림 31> Diffuse 옆 빈칸으로 드래그 앤 드롭

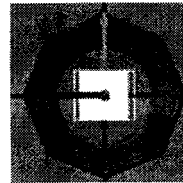
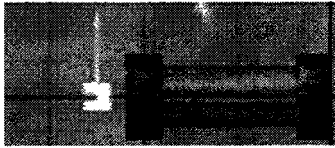
4.2.3 Animation 제작

아이콘 바에서 를 눌러 육면체 하나를 생성하고 오브젝트 패널에서 이름을 Box01이라 하고 크기를 30, 30, 30으로 한다.



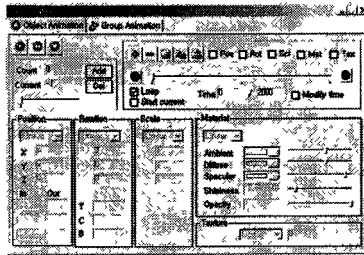
<그림 32> 육면체 생성

👉을 누르고 Box01을 이동하여 그림 33과 같이 Tube02의 가운데에 위치시킨다.




<그림 33> 생성된 육면체 위치 지정

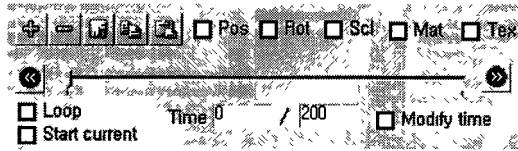
아이콘 Set에서 Animation Editor 아이콘 📄을 누른다. 그림 34와 같이 Object Animation Editor에서 좌측 Add 버튼을 눌러 새로운 애니메이션을 하나 만든다






<그림 34> Animation Editor

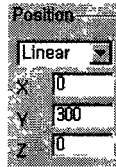
Position에 체크를 하고 Time0 에서 Add Key 📄버튼을 누른다.

그리고 Time의 뒤 0을 2000(2초)으로 바꾸고 Key 탭을 맨 끝으로 옮긴다. 그리고 다시 Add Key  버튼을 눌러 Position을 추가한다.



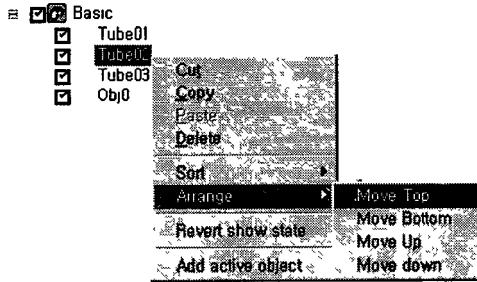
<그림 35> 설정

Position 설정에서 그림 36과 같이 Time 2000의 Y 값을 300으로 하고 Modify Key  버튼을 누른다. 그리고 Play 버튼  을 눌러 애니메이션을 확인한다. 이러한 애니메이션 설정은 화면상에서 애니메이션 아이콘  을 누르고 이동시켜 직접 설정할 수도 있다.



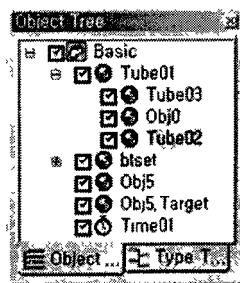
<그림 36> Position 지정

그런데 반투명한 Tube02의 안에 들어간 Box01이 보이지 않게 됨을 볼 수 있다. 그것은 오브젝트의 Z-order와 관련된 것으로 투명한 오브젝트는 그 뒤에 놓이는 물체보다 Z-order의 순위가 위쪽이어야 한다. 어느 것이 먼저 그려지느냐의 문제인 것이다. 그러므로 오브젝트 트리에서 정렬순서(Sort)를 Z-order Sort로 하고, 그림 37과 같이 Tube02의 Arrange를 Top으로 이동한다. 그러면 Box01이 보일 것이다.



<그림 37> 정렬순서 지정

이번에는 계층구조(Hierarchy)를 이용한 애니메이션을 해보도록 한다. 즉, 부모(Parent)에 지정한 애니메이션을 그 자식(Child)들도 그대로 따르도록 하는 것이다 오브젝트 트리에서 Tube03을 Tube01로 드래그 앤 드롭 하면 그림 38과 같이 그 아래에 속하게 된다 Box01과 Tube02도 같은 방식으로 Tube01의 아래 속하게 한다. Tube01을 선택하고 옆의 오브젝트 패널에서 상속성(Inheritance)을 보면 그 아래에 속한 자식 오브젝트(Child Object)들이 어떠한 항목을 쫓아오는지 알 수 있다 여기에서 체크를 끄면 그 항목은 자식(Child) 오브젝트에 영향을 주지 않는 항목이 된다



<그림 38> 상속성 지정

Tube01을 선택하고 Animation Editor를 실행한다 그림 39와 같

이 Add 버튼을 눌러 애니메이션을 하나 만들고 옆의 Rot에 체크한다. 그리고 Time의 뒤 공간에 2000을 입력한다.



<그림 39> 위의 박스에서 Rot 체크

이제 Time0 에서 (Add Key)를 눌러서 Rotation 키 프레임 (Keyframe)을 만든다 그리고 Y축의 값은 그대로 0으로 둔다. Time 1000에서 () 또 Add Key를 누르고 활성화된 Rotation의 Y값으로 120을 입력한다. 그리고 (modify)를 누르면 뷰포트에서 오브젝트 전체가 120도 돌아가 있는 것을 볼 수 있다. 언제나 변형시킨 Key값을 확정하기 위하여 를 꼭 눌러야만 한다. 전체가 돌아간 이유는 Tube01에 나머지 오브젝트들이 자식 (Child)으로 속해 있어서 변화를 따라가기 때문이다 다시 Time 2000으로 가서 를 누르고 Rotation Y값을 0으로 바꾸고 를 누른다. 를 눌러 결과를 확인한다

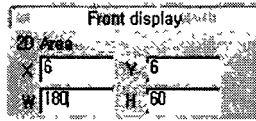
Loop에 체크를 한다. Loop 그러면 방금 만든 애니메이션은 계속 반복할 것이다. Loop는 해당 애니메이션을 정지시킬 때까지 무한정 반복하도록 하는 명령이다.

4.2.4 Display Object로 버튼 만들기

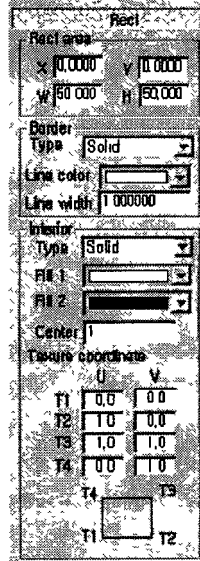
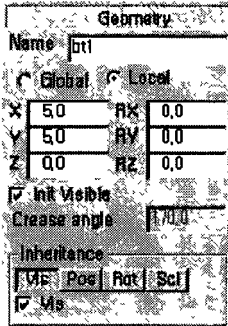
이제 제작한 애니메이션을 사용자가 제어할 수 있도록 제어 버튼을 만들어본다 버튼은 사용자가 화면을 둘러거나 줌을 해도 항상 제 위치에 놓여있는 것이 좋으므로 2차원적인 이미지로 항상 화면 앞에 떠있는 프론트 디스플레이 오브젝트(Front Display Object)를

만들도록 하겠다.

D(Create Front Display Object)를 클릭하고 Perspective View의 좌측 상단 구석에서부터 그림 40과 같이 드래그 한다. 그리고 **키**반드시 Esc 키를 누르거나 **S**(Set default State)를 클릭하여 Create 모드에서 나오도록 한다. 그렇지 않으면 뷰포트에 마우스를 댈 때 마다 오브젝트가 새로 만들어 질 것이다. 이름을 bt-set이라고 정하고 좌표 값과 크기는 오브젝트 패널에서 그림 41과 같이 설정한다.



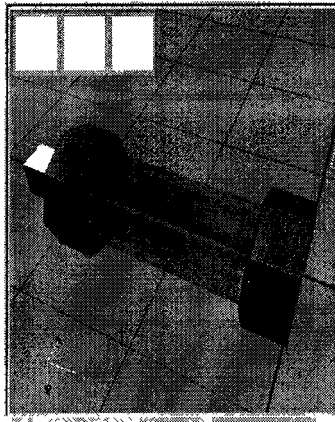
<그림 40> 드래그한 좌표 값



<그림 41> 버튼의 설정값

만들어진 프론트 디스플레이 오브젝트에 버튼을 3개 만들도록 한다. 버튼은 □사각형을 프론트 디스플레이 오브젝트에 만들어서 색상을 넣거나 텍스처(Texture)를 맵핑하여 만들 수 있다. 먼저 bt-set이 선택된 가운데 (Edit Display Object)를 클릭한다. 그러면 뷰포트에는 프론트 디스플레이 오브젝트만 남게 된다

이 프론트 디스플레이 오브젝트 안에 그림 42와 같이 사각형을 그린다. 그리고 이름을 차례대로 bt1, bt2, bt3라 고치고 사이즈와 상대 위치 값을 다음 패널과 같이 바꾼 뒤 Y 위치 값을 5, 65, 125로 각각 고친다




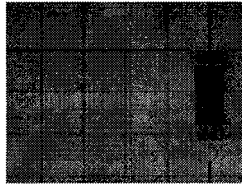
<그림 42> 사각형 배치

오브젝트 패널에서 Fill 1의 색상을 각각 Red, Blue, Green으로 바꿔준다. 그 바로 위 Type에서 Solid 대신 Image를 선택하면 텍스처 맵핑을 하여 이미지를 넣을 수도 있다.


4.2.5 Camera 설정

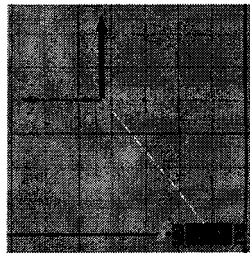
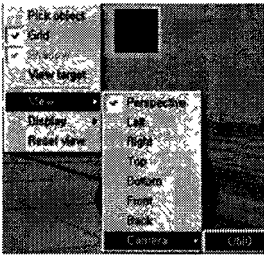
이제 가장 시뮬레이션 하기에 좋은 각도로 카메라를 만들어 본

다.  (Create Camera)를 클릭하고 Top 뷰에서 그림 43와 같이 카메라 위치부분을 클릭한 상태에서 타겟쪽으로 드래그 한다. 그림 하늘색 원뿔(Cone)이 보일 것이다.





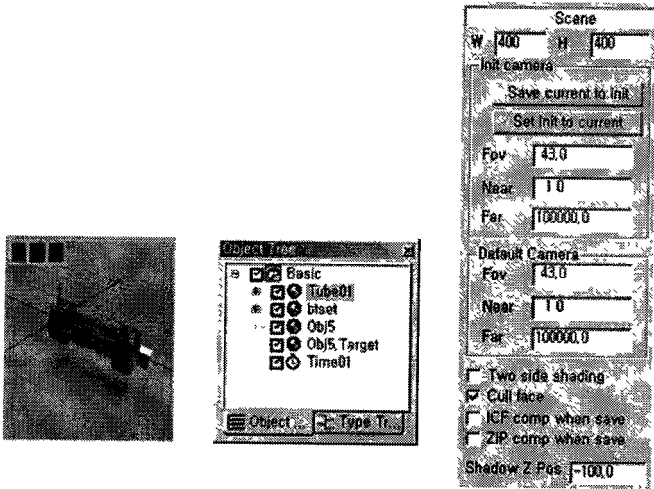
<그림 43> 카메라 각도 설정

Esc 키를 눌러 Create 모드를 해제한 후 Perspective 뷰의 오른쪽 버튼 메뉴에서 그림 44과 같이 카메라 뷰로 변경한다. 화면에는 아무것도 보이지 않을 것이다. Front 뷰를 줌 아웃(Zoom Out)해서 카메라의 위치를 확인해본다. 그리고  버튼으로 그림 45와 같이 카메라와 타겟의 위치를 변경한다



<그림 44> 카메라 뷰 변경 <그림 45> 타겟 위치 변경

 과  을 사용하여 카메라 뷰를 그림 46와 같이 한다 그리고 오브젝트 트리에서 그림 47과 같이 맨 위 Basic을 선택하면 오브젝트 패널에 Scene에 대한 설정이 나온다 그림 48과 같이 설정 창에서 시뮬레이션의 가로 세로 크기를 400, 400으로 한다

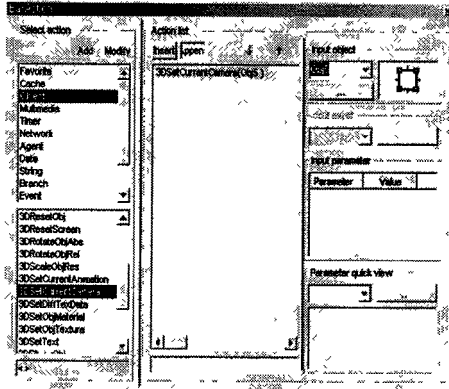


<그림 46>카메라뷰 설정 <그림 47> 오브젝트 선택 <그림 48> 크기 설정

4.2.6 기능 구현 (Event, Action)

이제 PlayMo3D의 강력한 인터랙션(Interaction)을 이끌어내는 이벤트(Event)와 액션(Action)을 사용하여 기능 구현을 한다. 우선 이벤트 트리에서 **Root** (Root)를 선택한다. Root는 시뮬레이션이 시작하자마자 진행될 내용을 넣는 이벤트이다

[Action Bar]를 눌러 그림 49과 같이 액션 윈도우(Action Window)를 띄운다. 그리고 오브젝트에서 3DsetCurrentCamera를 더블 클릭한다. Action List에 항목이 나오면 Input Object에서 사용할 카메라 이름인 Obj5를 선택한다. 그러면 시뮬레이션이 시작할 때부터 Obj5 카메라의 뷰로 보이게 될 것이다 저장하고 **[Simulator]**를 눌러서 결과를 확인해본다.



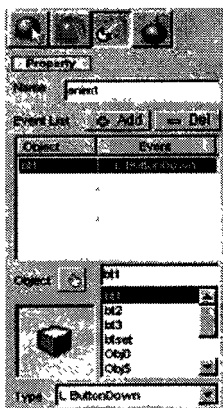
<그림 49> Bar Action window

이벤트 트리의 빈 공간을 클릭하고 키보드 E를 누르면, 그림 50와 같이 새 이벤트 노드가 생성될 것이다 이 이벤트에 Obj0의 애니메이션을 실행시키는 명령을 넣도록 한다.



<그림 50> 이벤트 추가

옆의 이벤트 패널에서 그림 51과 같이 이름을 Anm1이라 하고 옆의 이미지와 같이 bt1에 대해 LButtonDown인 이벤트 속성을 만든다 즉, 프론트 디스플레이 오브젝트에 속한 버튼 형태인 bt1을 마우스 왼쪽 버튼으로 누른다는 의미이다. 그러한 행위가 일어났을 때 발생할 액션을 이제 정의 내려 주면 된다.



<그림 51> 이벤트 패널 설정

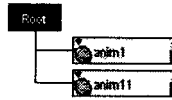
그림 52과 같이 액션 윈도우(Action Window)에서 이번에는 오브젝트의 3DplayAnimation을 더블클릭하고 오브젝트로 obj0을 선택한다 Input Parameter에서 anim_number가 0임을 확인하고 나머지는 그대로 두면 된다. 시뮬레이션으로 확인해보면 빨간색 버튼을 누를 때마다 육면체가 애니메이션 되는 것을 확인할 수 있다.



<그림 52> 액션 설정

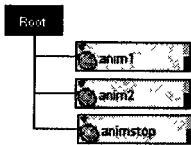
이번에는 부모(Parent)인 Tube01의 애니메이션을 실행시켜본다. 우선 이벤트 anim1을 복사하고 이벤트 트리 빈 공간을 클릭한 후에 붙여 넣는다. 그러면 그림 53과 같이 같은 내용의 이벤트 노드가 생성된다. 이벤트 패널에서 이벤트 이름을 anim2로 바꾸고 이미 있는 이벤트 리스트의 항목을 선택하고 오브젝트를 bt2로 바꾼다. 그리고 액션 바(Action Bar)에서 3DplayAnimation의 오브젝트를

Tube01로 바꾼다 시뮬레이터에서 확인해보면 파란 버튼을 누르면 오브젝트 전체가 지속적으로 회전하는 것을 볼 수 있다



<그림 53> 추가 이벤트 생성


이제 애니메이션을 멈추는 이벤트를 만들어본다 이벤트 트리에서 그림 54와 같이 새로운 이벤트 노드를 생성하고 이름을 animstop이라 정한다. 그리고 이벤트 패널에서 그림 55와 같이 bt3에 대해 LButtonDown을 적용시킨다



<그림 54> 멈추는 이벤트 생성 <그림 55> 이벤트 적용

액션 리스트에 다음과 같은 액션들을 추가한다. 즉 두개의 Animation들을 Stop시키는 것이다

- 3DstopAnimation(Tube01,0,0)
- 3DstopAnimation(Obj0,0,0)

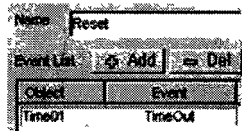
이번에는 애니메이션이 정지된 다음에 1초 후에 다시 처음 상태로 되돌아오도록 해보도록 한다. 우선 타이머 오브젝트(Timer Object)를 하나 만든다. 를 클릭하고 오브젝트 트리를 보면 새로운 타이머 오브젝트가 생성되어있을 것이다. 이름을 time01이라고 변경한다.

그리고 animstop 이벤트의 액션에 타이머의 SetTimer를 그림 56과 같이 추가하고 오브젝트를 time01로 지정한다. 그리고 Period에는 1000이라 입력한다. 타이머에서는 1000란 값이 1초를 나타낸다.

```
3DStopAnimation(Tube01, 0, 0)
3DStopAnimation(Obj0, 1000, 0)
SetTimer(Time01, 1000)
```

<그림 56> 타이머 적용

새로운 이벤트 노드를 하나 더 만들고 그림 57과 같이 이름을 Reset이라 정한다. 그리고 Object는 Time01, Type은 Time Out을 지정한다



<그림 57> Reset 이벤트 설정

액션 리스트에는 다음의 액션들을 추가한다.

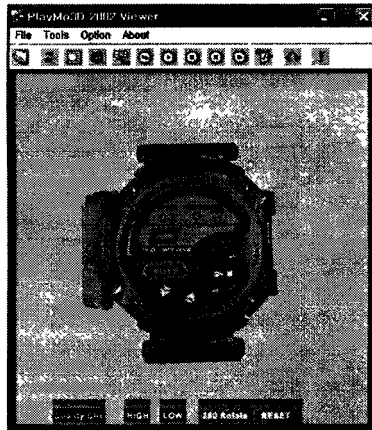
- KillTimer (Time01)
- 3DresetObj (Tube01)
- 3DresetObj (Obj0)

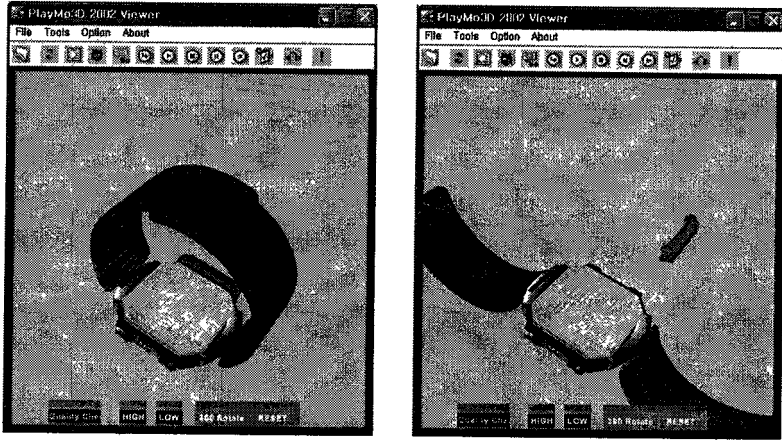
Timer는 KillTimer를 해주지 않으면 계속 반복하는 성질을 가지고 있으므로 한번만 작동하기를 원할 경우 그 실행된 이벤트에 KillTimer를 꼭 넣어줘야 한다. 3DresetObj는 오브젝트가 애니메이션을 하기 이전의 최초 상태로 돌아가도록 해준다.

이제 저장하고 시뮬레이션을 통하여 마지막 결과물을 확인하도록

한다. PlayMo 2D와 마찬가지로 인터넷에서 작성된 결과물을 볼 수 있는 Publish 기능은 PlayMo 3D에서 제작된 콘텐츠를 인터넷에 올릴 수 있도록, PlayMo3D 콘텐츠를 볼 수 있는 HTML 페이지를 자동으로 생성해주는 기능이다. 이것으로 간단하게 웹페이지를 작성할 수 있다

그림 58은 실제 제품을 3D로 구현한 예이다 사용자는 마우스로 실제 모습을 눈앞에서 보는 것과 같이 직접 작동 시켜볼 수 있고, 회전을 시켜가면서 전체 외관을 살펴볼 수도 있다.





<그림 58> PlayMo 3D 구현 모습 예

5. 결론

본 논문은 단순히 보여 주기만 하는 기존 웹 콘텐츠의 단점을 해결하며 FSM(Finite State Machine) 기반의 HMI (Human Machine Interface) 모델링 및 시뮬레이션 저작도구인 PlayMo를 활용하여 군용목적으로 다양한 기능성을 보유한 전자장비, 각종, 기계, 설비 등의 제품의 기능을 가상으로 조작할 수 있는 인터랙티브 멀티미디어 콘텐츠를 제작함에 있어 효과적인 구현방안 알 수 있다. 2D와 3D로 일반 제품과 군용 장비의 복잡한 기능과 모양을 디지털 입력 장비로 디지털화 하고 앞의 구현 사례 연구에서 보듯이 이것을 PlayMo로 구현하여 실제 제품과 똑같은 모양과 기능으로 사용자가 PC상에서 마우스를 통하여 작동 시켜볼 수 있다

이것은 전자제품이나 기계장비의 UI(User Interface)개발에서 판

때, AS(After Service)에 이르는 일련의 기업 활동에 필요한 UI 사양서 작성, UI 기능성 평가, 사용자 매뉴얼 작성, 홍보용 디지털 카탈로그 작성, AS매뉴얼 작성 등의 업무를 유기적으로 통합하고 완벽하게 디지털화하여 기존에 상존했던 비효율성 및 중복투자를 해소할 수 있는 장점을 적극 활용한다 또한 각 업무 단계에서 발생했던 커뮤니케이션상의 오류를 제거함은 효과적인 개발절차가 된다.

근본적으로 PlayMo는 디지털 콘텐츠의 차세대 도약을 위해 다양한 멀티미디어 콘텐츠의 휴먼 인터랙티비티 강화, 표현력의 심미적 레벨 업, 충실한 커뮤니케이션 실현으로 군용 전자교범목적에 상응하는 최고수준의 실시간 사실감을 웹상에서 완벽하게 구현함에 가장 적합하기에 민·군의 응용분야를 체계화할 수 있다.

| 참고문헌 |

Sukhoon Kang, Jaeyoun Lee, Jiexi Lin, Seokhoon Bae, Calvin J. Hur “*PlayMo*. An Event-driven Modeling Tool for Active Catalog”, Proc of 29th EUROMICRO Conference, Belek, Turkey, 2003.

Sukhoon Kang, Calvin J. Hur “An Event-driven Development Environment for Interactive Digital Contents: *PlayMo3D*”, Proc. of 30th EUROMICRO Conference, Rennes, France, 2004.

강석훈, 이환규 “*PlayMo*와 기존 제품과의 기능 비교”, 디지털 기업과 전자상거래를 위한 액티브 전자카탈로그의 활용기법, 생능출판사, pp. 15-16, 2001

An Effective Implementation Method for Dual Use of Web-based Multidimensional Interactive Digital Contents

Suk-Hoon Kang, Dae-Cheong Kim

Active catalog is a kind of digital content that enables consumers to test the functions and features of products from their PCs as if they were using it in real life, by simulating the actions and responses of the product. This new type of interactive digital content can be used extensively to make sales personnel training manuals, sales tools, user manuals and user trouble shooting documents

With active catalogs, companies will be able to compare different designs, show actions according to different functions, and evaluate user reaction to new products without having to produce a single physical prototype or mock-up. At the same time, consumers will be able to understand and 'operate' the product and make well-informed purchase decisions.

In this paper, we present a visual event-driven modeling tool, PlayMo, for creating active catalogs, analyze the advantages of using PlayMo, describe the event-driven method used by PlayMo and also introduce two enhanced characteristics of the Event Flow Chart with which the events in PlayMo are structured.

Interactive digital content by using the PlayMo3D makes easy, simple and effective design for e-learning, e-catalogue, e-marketing/sales, e-prototyping, customer support, etc. Through

its application-ready 3D function visualization solution, engineers and designers can rapidly turn a CAD design model into a 3D interactive virtual product, and the effective function prototyping job can be also completed within a minute

Key words . Interactive Digital Contents, Dual Use Technology, Web 3D.