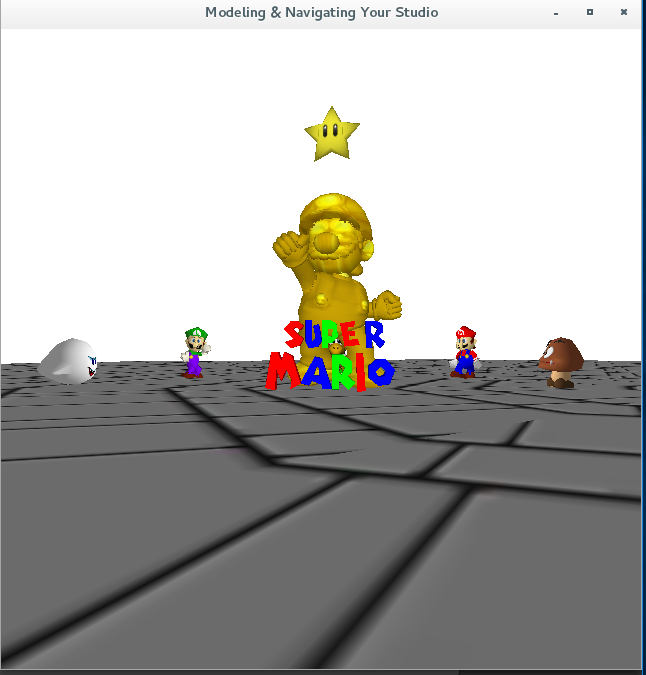
**컴퓨터그래픽스 설계과제:**

**가상 박물관의 모델링 및 네비게이션**

실행결과 # scene 1



**조원 1 : 20123411 전효창**

**조원 2 : 20123432 홍용현**

**개발과정 및 역할**

1. 이번 과제에 주어진 소스를 이용하여 물체에 텍스쳐를 입히고 퐁 리플렉션을 적용 – 홍용현

2. 리소스 파일을 찾아 구현한 소스에 다수 적용. 텍스쳐가 적용된 물체에 TRS변환 – 전효창

3. 5개의 장면을 각각 .dat파일에 저장하고 Save, Load 구현 – 홍용현

4. AABB 알고리즘을 응용한 충돌 감지 및 물체 ON/OFF 추가 기능 구현 – 전효창

5. 서브메뉴를 이용한 물체 선택 및 이동 기능 구현 – 홍용현

6, 프로그래밍 부문 최종 정리 – 전효창

7. 보고서 작성 – 홍용현

**구현기능 - 기본**

.obj .mlt 파일 및 텍스처 이미지 read

삼차원 TRS 변환 및 퐁 리플렉션을 통한 모델 배치

가상 카메라를 통한 네비게이션

서브메뉴를 이용하여 물체를 선택하고 이동

메뉴를 이용해 5개의 장면을 각각 .dat 파일에 저장하여 Save, Load

**구현기능 - 심화**

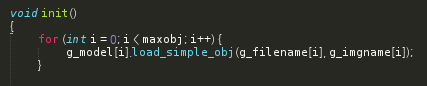
충돌 감지 : 장면에 놓인 물체를 뚫고 지나가지 않으며 충돌시 카메라가 튕겨나간다.

물체ON/OFF: 물체에 접근하여 물체를 OFF시킬 수 있으며 일괄적으로 ON시킬 수 있다.

연속 회전 : 물체의 각도를 바꾸는 것 외에도 시간에 따라 바뀌어 계속 회전하게끔 하였다.

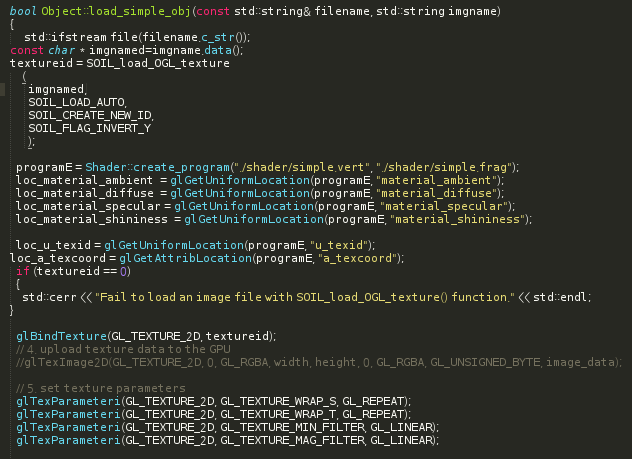
**구현 기능에 따른 소스파일 설명**

.obj .mlt 파일 및 텍스처 이미지 read



Scene파일들로부터 물체를 구성할 파일의 정보를 받아오면 그것을 기반으로 하여 파일을 읽어들인다.

이때 기존 load\_simple\_obj 함수가 obj, mtl만 읽던 것 외에 img파일도 읽게 하였다.



본래 draw함수 있던 것을 load할 때 1회만 처리하게 하여 부하를 줄였다.

그 결과 심각한 렉이 말끔히 사라지고 전체적인 성능 향상이 이뤄졌다.

**삼차원 TRS 변환 및 퐁 리플렉션을 통한 모델 배치**

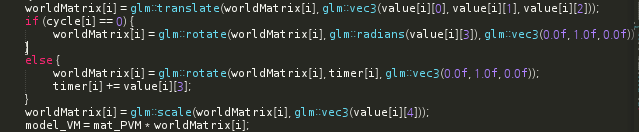
기존 PVM매트릭스와 TRS매트릭스를 통한 연산으로 월드좌표계가 형성된 물체를 랜더링 파이프라인을 통해 실제 화면에 어떻게 보일 것인가가 결정된다.

우선 버텍스 쉐이더에서 연산이 완료된 정보(position, fronfacing) 등이 clipper and primitive assembler 부분으로 넘어가게된다.

Clipping에선 실제 화면에서 표시될 부분을 제외하고 제거하는 작업을 거치며 rasterization에선 이러한 vertex를 실제 픽셀화 시키는 작업을 하게 된다.

한편 opengl 2.0에 이르러 버텍스 쉐이더, 플래그먼트 쉐이더가 병렬화 됨에 따라 varying 을 통해 컬러값을 비롯한 일부 정보를 버텍스에서 처리한 뒤 직접 넘길 수 있게 되었고 그래픽 처리의 향상을 가져왔다.

실제 구현



TRS의 경우 대부분의 변수에 배열을 선언하여 여러 물체를 for문 하나로 처리하도록 했다.



Fragment shader는 두 함수를 곱하여 적용하는 것으로 퐁 쉐이더와 텍스쳐를 한 번에 처리했다.

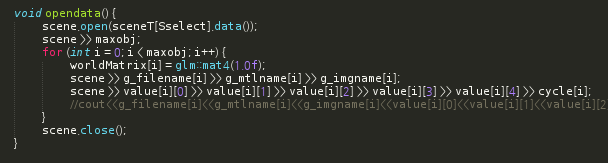
\* Draw 함수 또한 기존 HW3에다 텍스쳐를 추가하여 변화가 있었으나 소스가 유사하므로 이미지 첨부는 생략합니다.

**가상 카메라를 통한 네비게이션**

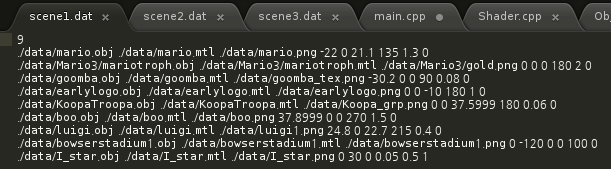
기존 구현된 소스를 재활용하였고 기능상으로도 큰 차이가 없기에 생략합니다.

카메라와 관련된 추가 기능은 후술.

**메뉴를 이용해 5개의 장면을 각각 .dat 파일에 저장하여 Save, Load**



sceneT배열과 Sselect 변수를 통해 장면을 관리하고 dat 파일로부터 정보를 받아들인다.



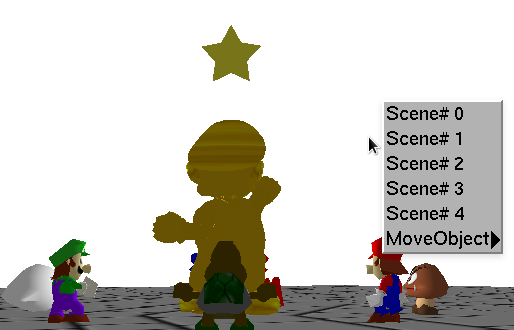
파일 구조는 다음과 같다.

1. 첫번째 줄 obj의 개수 – 위 파일에선 9이므로 9개이다.

2. 두번째 이후의 줄 – 9가지의 정보를 담고 있다.

obj파일 mtl파일 png파일 Translate-x값 y값 z값 회전값 크기값 연속회전여부

3. 연속 회전 여부는 1과 0으로 표현하며 1일 경우 회전 값에 비례하여 실행되는 동안 물체가 회전한다.

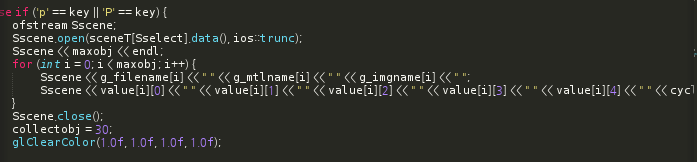
 장면 2

**서브메뉴를 이용하여 물체를 선택하고 이동**



메뉴의 MoveObject 항목을 이용하여 서브메뉴로 진입하면 각각의 물체를 선택할 수 있다. 서브메뉴를 통해 물체를 선택하게 되면 배경이 위 사진과 같이 회색으로 변하고 물체 이동 모드로 진입하게 된다.

방향키가 카메라 대신 물체의 위치 변경을 담당하게 되는데 원하는 위치에 배치가 끝나면 p키를 통해 저장할 수 있다.

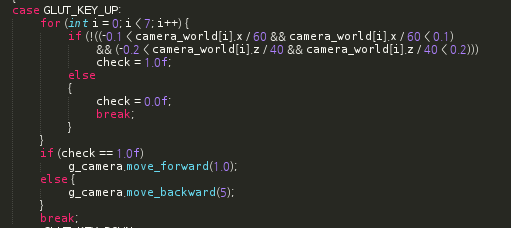


저장이 완료되는 즉시 배경이 다시 흰색으로 변하며 다른 장면을 진입하거나 프로그램을 재실행하여도 결과 값이 저장됨을 알 수 있다.

**연속회전**

장면 데이터 파일에만 차이가 있을 뿐 과제 2의 내용을 그대로 적용하였기에 설명 생략합니다.

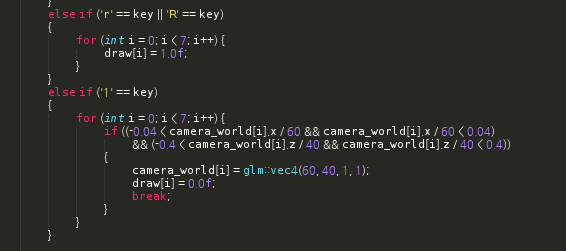
**충돌 감지**



기본 원리는 카메라와 월드좌표의 거리를 이용하는 것이다. 물체의 좌표를 기준으로 가상의 사각형 충돌 박스를 만들어 충돌박스 대비 카메라 x,y,z 좌표 값을 구하여 충돌 여부를 검출한다.

충돌시 카메라는 반대방향으로 밀려난다.

**물체ON/OFF**



충돌과 유사하나 판정을 조금 더 크게 잡아 약간 떨어져도 삭제가 가능하게 했다.

해당 물체에 접근하여 1키를 누르면 대상 물체가 삭제되고 충돌도 일어나지 않는다.

이후 R을 누르면 삭제된 모든 물체들이 복원된다.