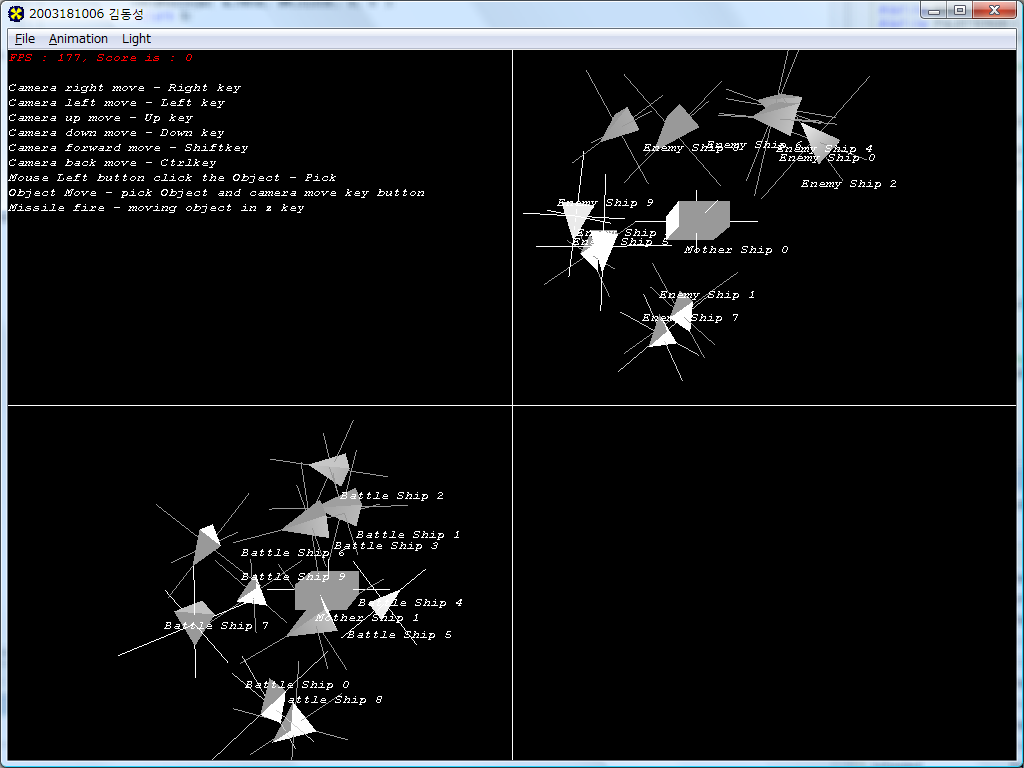
Direct 3D 프로그래밍 과제

조명을 메뉴를 사용해서 변경하고, 조명색을 바꿔보기

2003181006 김동성

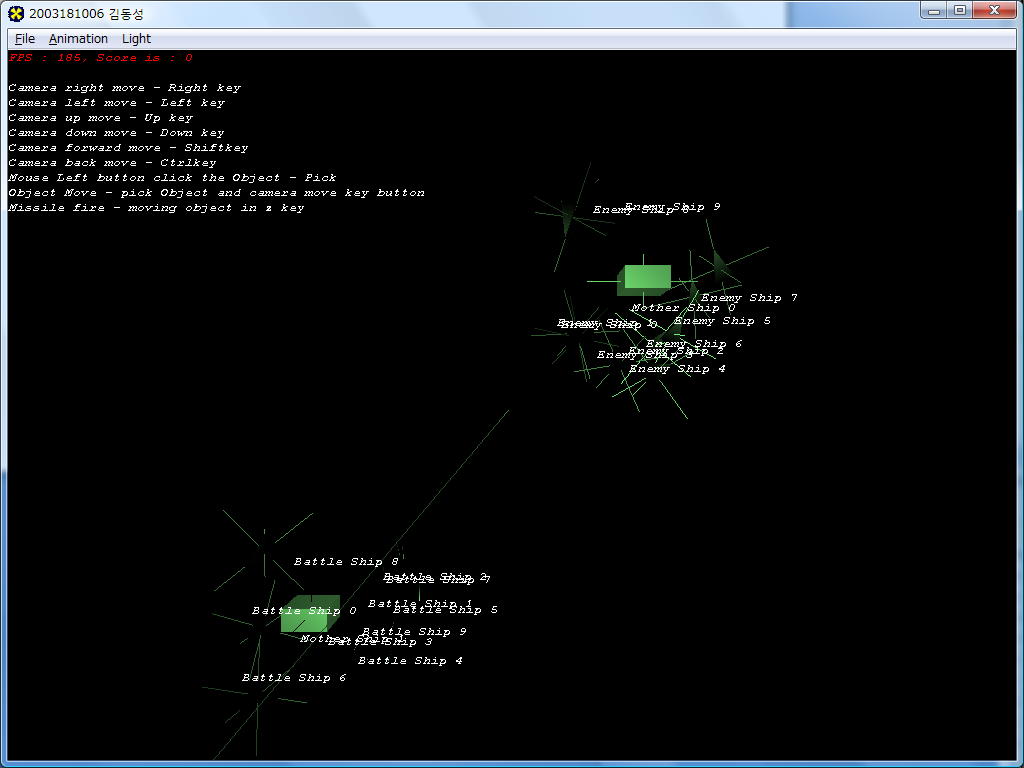
실행화면 1. - 방향성 광원을 사용했을 때. (방향은 +x 축 방향입니다.)



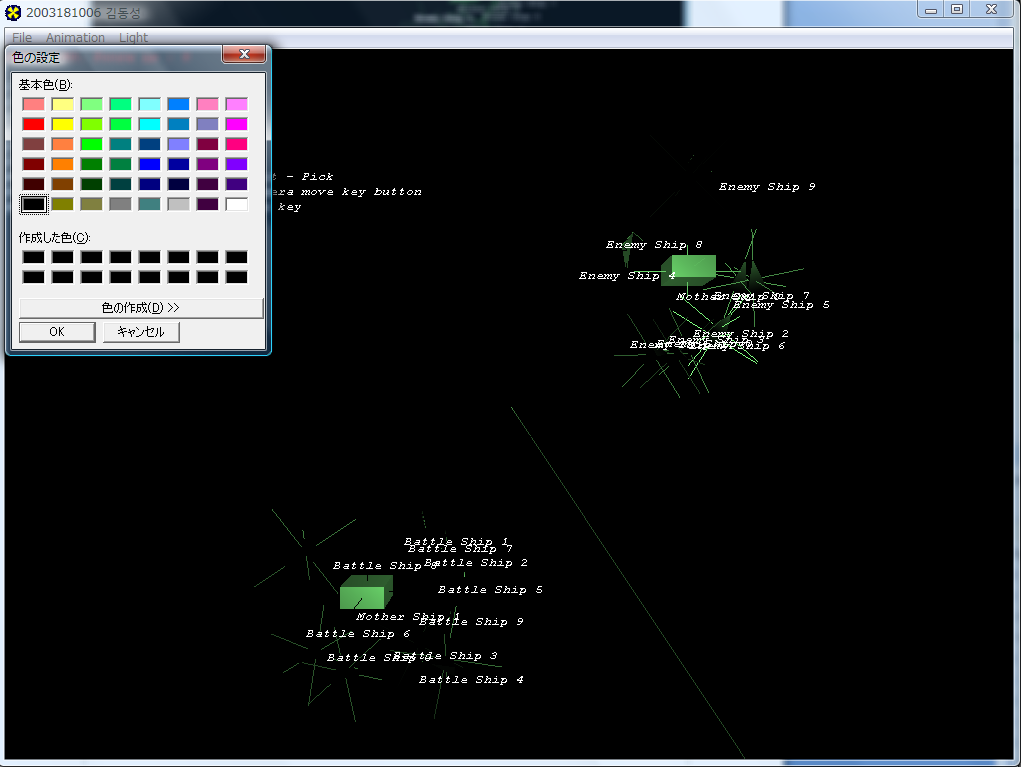
실행화면 2. – 점 광원을 사용했을 때 (중심점은 0,0,0 입니다)



실행화면 3 – 스포트 광원을 사용했을 때 (위치는 (0,0,-15) 이며, 방향은 +z축 방향으로 했습니다)



실행화면 4 – 광원색을 바꾸고 싶을 때 화면입니다.



구현 방법

1. 우선 조명을 주기 위해 이전에 버텍스를 만들면서 색을 사용했던 것을 모두 제거했습니다.
2. 추가적으로 법선벡터를 심었습니다.
3. 각각의 버텍스에 대하여 법선벡터를 계산하는 ComputeNormal 함수를 만들었습니다.  
    (CGameApp.h / CGameApp.cpp 593Line)
4. 메뉴를 생성하였습니다. 각각 방향, 점, 스포트, 광원색을 설정할 수 있도록 하였습니다.
5. 광원색의 기본색은 매크로로 미리 지정하였습니다. (CObject.h) 지정 내용은 아래와 같습니다.

const D3DXCOLOR WHITE( D3DCOLOR\_XRGB(255, 255, 255) );

const D3DXCOLOR BLACK( D3DCOLOR\_XRGB( 0, 0, 0) );

1. 조명색의 경우 CHOOSECOLOR의 윈도우 공용 Dialog를 사용하여서 색을 받게 했습니다.  
   받는 컬러값이 COLORREF 라 D3DXCOLOR로 변환하기 위해서   
   D3DXCOLOR D3DCOLOR\_XRGB(GetRValue(COL.rgbResult), GetGValue(COL.rgbResult), GetBValue(COL.rgbResult));  
   로 값을 받았습니다.  
   해당 함수의 구현은 SetLightColor() 로 해놓았습니다. (CGameApp.cpp 571Line)
2. 광원을 설정하는 함수를 만들었습니다.   
   SetLights( ) 함수(CGameApp.cpp 858Line)로 구현하였습니다. 이곳에서 구현은  
     
   1. SetRenderState 함수를 사용하여 D3DRS\_LIGHTING 을 사용합니다.  
   2. SetRenderState(D3DRS\_NORMALIZENORMALS 를 사용해서 각 버텍스의 법선벡터를 정리합니다.  
   3. D3DLIGHT9 light; 를 선언합니다.  
   4. 이제 메뉴에서 받은 값에 따라 바뀌는 m\_LightType으로 조명 설정을 분기시킵니다.  
   5. 각각의 switch 분기에 따라 조명을 설정해줍니다.  
   6. 마지막으로 m\_m\_pD3DDevice에 설정한 Light를 지정하고, Light를 활성화 시킵니다.
3. 마지막으로 SetLights 함수를 FrameAdvance() 함수안에 넣어서 조명을 활성화 시켰습니다.