**REPORT**

**| Computer Graphics**

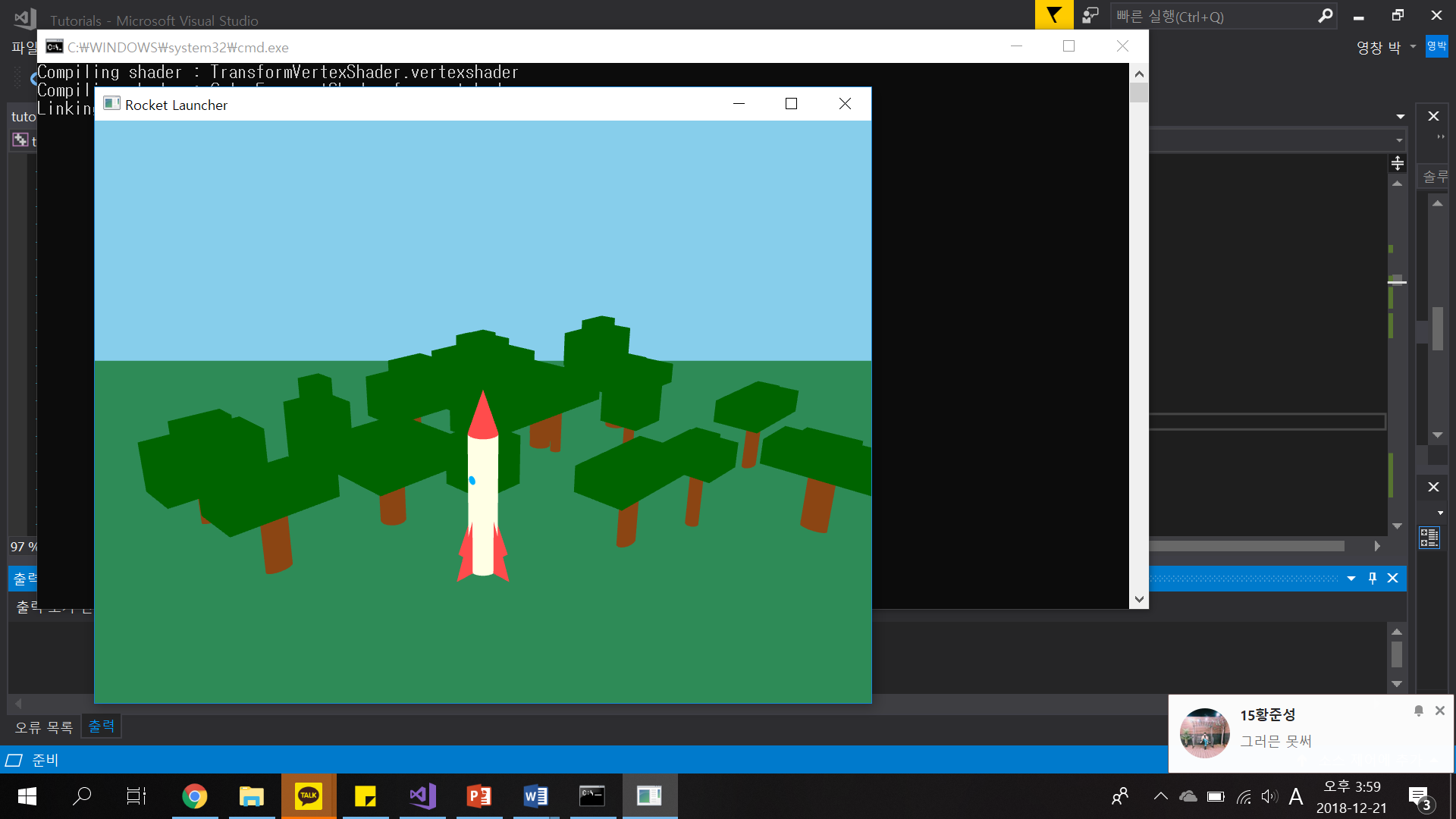
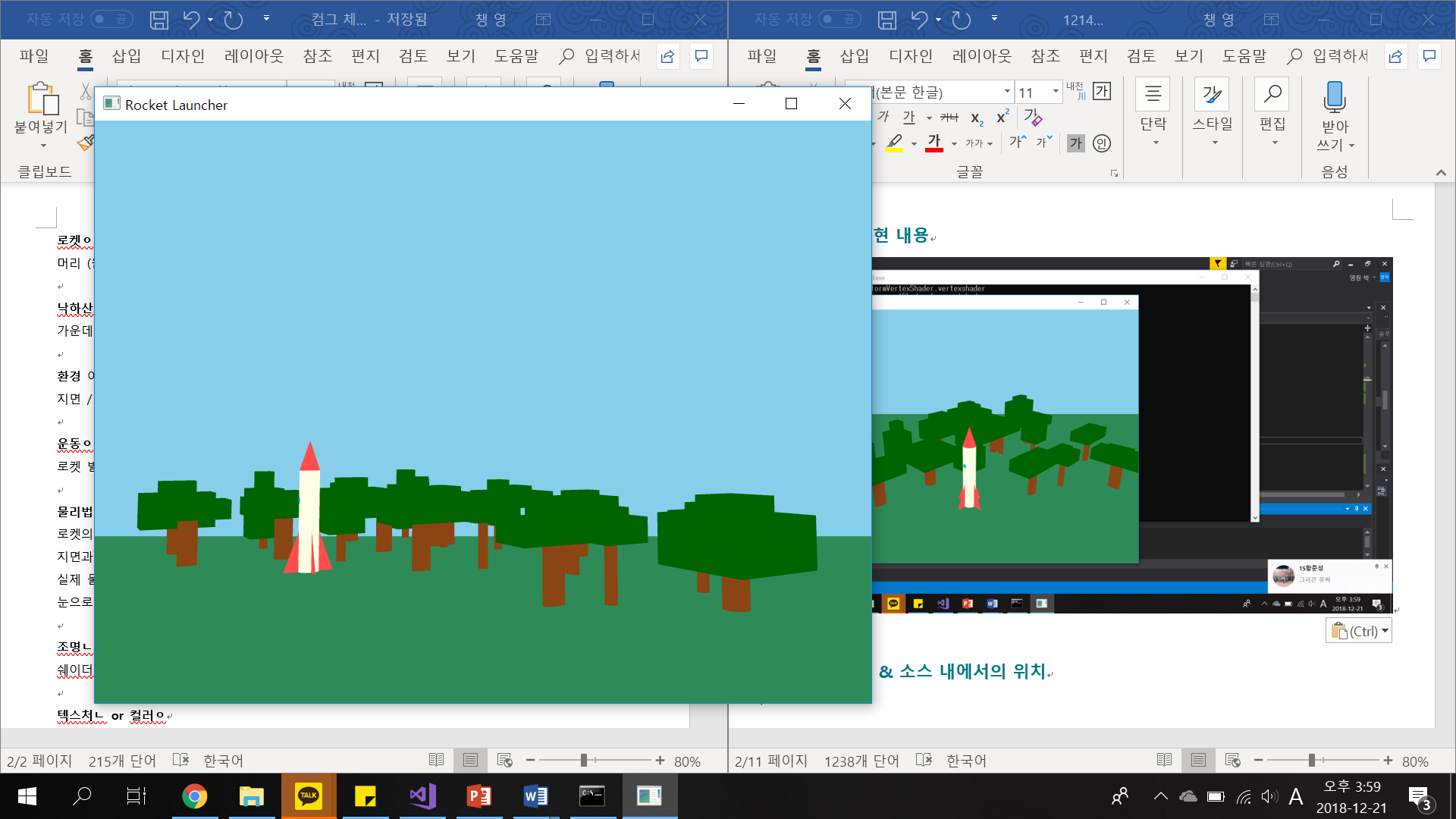
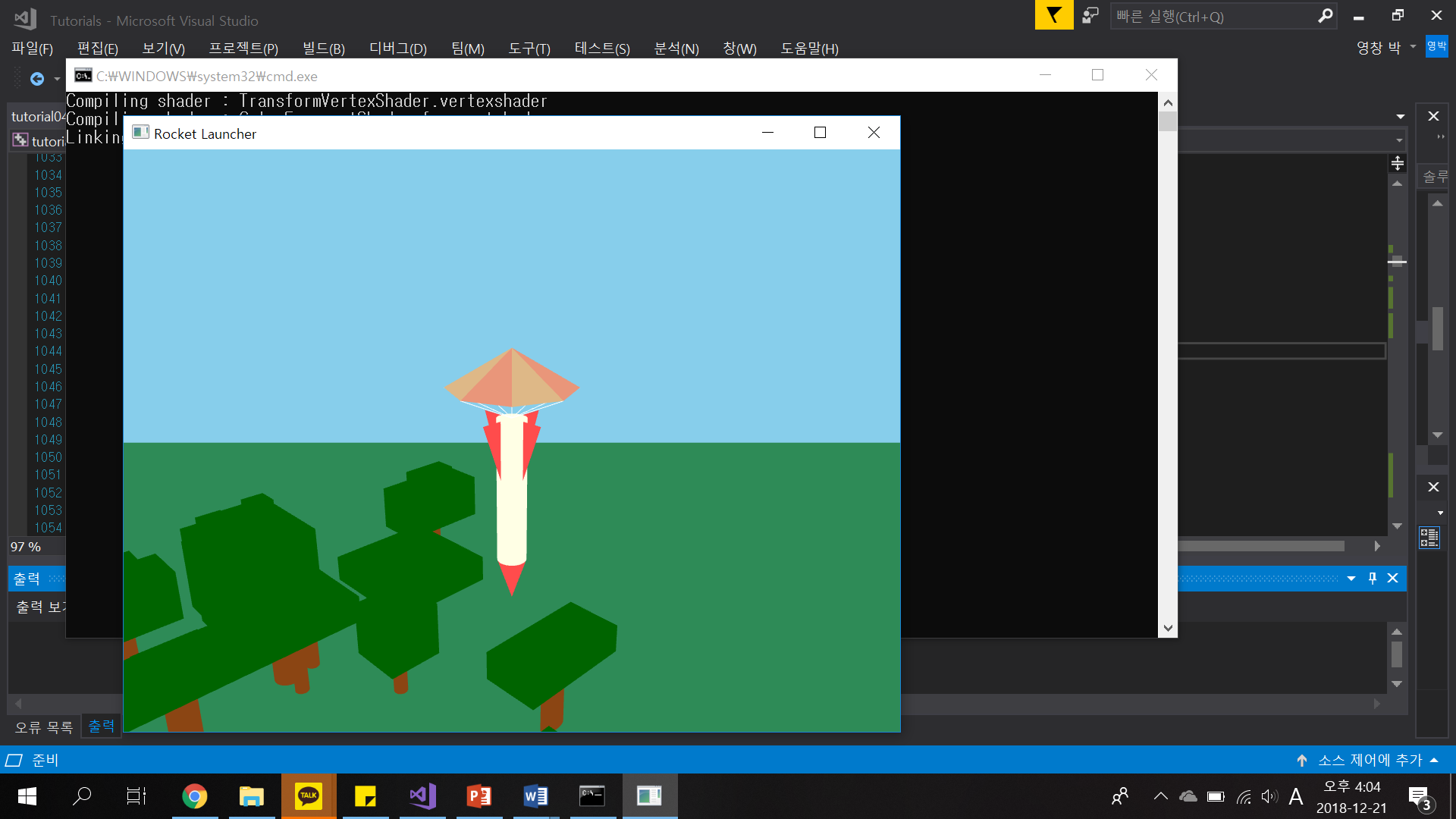
**| 최종 프로젝트**

**| 로켓 발사**

**| 컴퓨터정보공학과**

**| 12141540 박영창**

# **스크린샷 및 구현 내용**

**컴퓨터 그래픽스 최종 프로젝트 ( 로켓 발사 )**

1. 시작할 때 카메라는 로켓이 잘 보이는 위치에 세팅 되어 있으며, 키보드와 마우스로 시점을 변경할 수 있다.
2. 로켓은 원기둥, 원뿔, 원, 날개로 구성되어 있으며, 배경에 나무를 설치해 움직임을 인식할 수 있다.
3. 조작을 통해 로켓을 충전시켜 발사 시키고 낙하산을 펼 수 있다.
4. 물리 법칙에 의해 속도는 최고점까지 감소하다 중력에 의해 다시 빨라진다. ( 물리적으로 정확하지는 않음 )
5. 로켓은 공중에서 회전해 머리 쪽으로 떨어지며 바닥에 닿았을 때 멈춘다.

* 텍스처 / 조명은 구현하지 못했습니다..
* 실습 소스 코드 중 tutorial04의 환경을 사용했습니다.

# **조작 방법**

**로켓 발사 조작**

1. “ A “ – 로켓을 발사 시킬 에너지를 충전한다. 오래 누를수록 멀리, 높이 발사 가능
2. “ S “ – 로켓을 발사한다. 일정 에너지가 충전되어야 발사 가능
3. “ D “ – 낙하산을 편다. 로켓이 최고점을 지나 낙하 중에만 사용 가능

**카메라 시점 이동**

1. “ 방향 키 “ – 방향키를 눌러 상하좌우로 움직일 수 있다.
2. “ 마우스 “ – 마우스를 움직여 바라보는 방향을 바꿀 수 있다.
3. “ Z “ – 현재 로켓의 위치로 이동한다. 꾹 누르면 로켓을 따라간다.
4. “ X “ – 로켓을 멀리서 바라본다. 스크린샷의 두 번째 사진의 시점
5. “ C “ – 프로그램을 실행했을 때 초기의 위치로 이동한다.

# **주요 코드 설명 & 소스 내에서의 위치**

**makeCircleVertexData(GLfloat x, GLfloat y, GLfloat z, GLfloat radius, Glint numberOfSides)**

(x, y, z)가 중점이고 radius의 반지름을 갖는 원 모양의 정점 데이터를 생성합니다.

함수 내용은 tutorial04.cpp의 1534번 줄에 있습니다.

**makeCylinderVertexData(GLfloat x, GLfloat y, GLfloat z, GLfloat height, GLfloat radius, Glint numberOfSides)**

(x, y, z)가 중점, radius의 반지름, height의 높이를 갖는 원기둥의 정점 데이터를 생성합니다.

함수 내용은 tutorial04.cpp의 1561번 줄에 있습니다.

**makeConeVertexData(GLfloat x, GLfloat y, GLfloat z, GLfloat height, GLfloat radius, Glint numberOfSides)**

(x, y, z)가 중점, radius의 반지름, height의 높이를 갖는 원뿔의 정점 데이터를 생성합니다.

함수 내용은 tutorial04.cpp의 1588번 줄에 있습니다.

**정점 / 색상 데이터 선언**

로켓, 배경, 낙하산의 정점 데이터들을 배열에 저장하는 부분입니다.

로켓 몸통, 로켓 뚜껑, 로켓 창문, 나무 기둥은 위에 설명된 함수를 사용해 데이터를 생성했습니다.

선언 부분은 tutorial04.cpp의 77 ~ 1113번 줄에 있습니다.

**로켓의 움직임**

“ A “, “ S “, “ D “를 눌러 로켓이 움직이도록 조작했을 때 실행되는 부분입니다.

charge가 일정 값 이상일 때 launch가 활성화되면 glfwGetTime()과 아래 변수들을 사용해 움직임을 나타냅니다.

* + Yangle : 물리 법칙을 유사하게 표현하기 위해 사용된 변수로 sin 함수를 통해 물리 법칙을 구현했습니다. 시간이 지남에 따라 증가하며 높이를 계산하는 데에 쓰입니다.
  + Pangle : 바닥에 닿았을 때, 낙하산을 폈을 때에 Yangle은 계속 증가하므로 마지막 Yangle값을 저장해 대신 사용하는 변수입니다.
  + angle : 로켓이 공중에서 회전해 머리 부분으로 떨어지게 만들기 위해 사용하는 변수입니다.
  + currentH : 현재 높이를 나타냅니다. Yangle 또는 낙하산 여부에 의해 계산됩니다.
  + flag : 최고점에 도달했을 때 활성화되며, 활성화 시 낙하산을 펼 수 있습니다.
  + flag2 : 어색하게 움직이는 것을 피하기 위해 로켓이 공중에서 도는 타이밍을 맞춰줍니다.
  + parachute : 최고점을 지나 “ D “를 눌렀을 때 활성화되며 낙하산을 폅니다.

소스 코드는 tutorial04.cpp의 1204 ~ 1232번 줄, controls.cpp의 92~103번 줄에 있습니다.

**Rendering**

실제로 정점 / 색상 데이터를 이용해 화면에 표현하는 부분입니다.

* + MVP : 로켓의 MVP 매트릭스로, 위 움직임 코드를 통해 계산된 translation, rotation 행렬에 의해 움직입니다. 로켓 몸통, 로켓 뚜껑, 로켓 창문, 로켓 날개에 해당합니다.
  + MVPGround : 배경 중 바닥의 MVP 매트릭스로, 따로 변환하지 않습니다. 바닥에 해당합니다.
  + MVPPara : 낙하산의 MVP 매트릭스로, “ D “를 누른 뒤에만 나타나며, 위 움직임 코드를 통해 계산된 translation, rotation, scaling 행렬에 의해 나타나고 움직입니다. 낙하산 줄, 낙하산에 해당합니다.
  + MVPtree : 배경 중 나무의 MVP 매트릭스로, 미리 선언된 translation, scaling 행렬 배열을 반복문을 통해 표현합니다. 나무 기둥, 나뭇잎에 해당합니다.

소스 코드는 tutorial04.cpp의 1195 ~ 1202, 1234 ~ 1520번 줄에 있습니다.

**카메라 조작**

마우스, 키보드를 이용한 카메라 조작에 사용되는 코드 부분입니다..

extern 변수를 사용해 tutorial04.cpp 내에서 사용되는 변수를 가져와 사용합니다.

초기화 부분은 controls.cpp의 28 ~ 36번 줄입니다.

현재 시점 계산 부분은 42 ~ 74번 줄입니다.

키보드 입력에 따른 동작 부분은 77 ~ 89, 109 ~ 134번 줄입니다.

**Shading**

실제로 각 정점마다 색상을 부여해 각 면의 색을 표현하는 부분입니다.

텍스처와 조명을 구현하지 못해 각 정점의 색상만을 지정했습니다….

TransformVertexShader에서 layout에 따라 정점 좌표와 각 정점에 대한 색상을 계산하고 이를 ColorFragmentShader로 전달해 색상을 나타냅니다.

선언 부분은 tutorial04.cpp의 73 ~ 74번 줄에 있습니다.

소스 코드는 ColorFragmentShader.fragmentshader , TransformVertexShader.vertexshader에 있습니다.