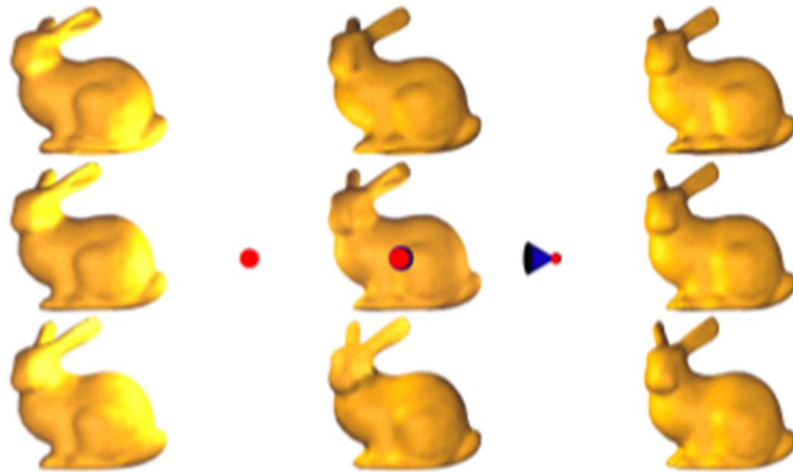


# Computer Graphics



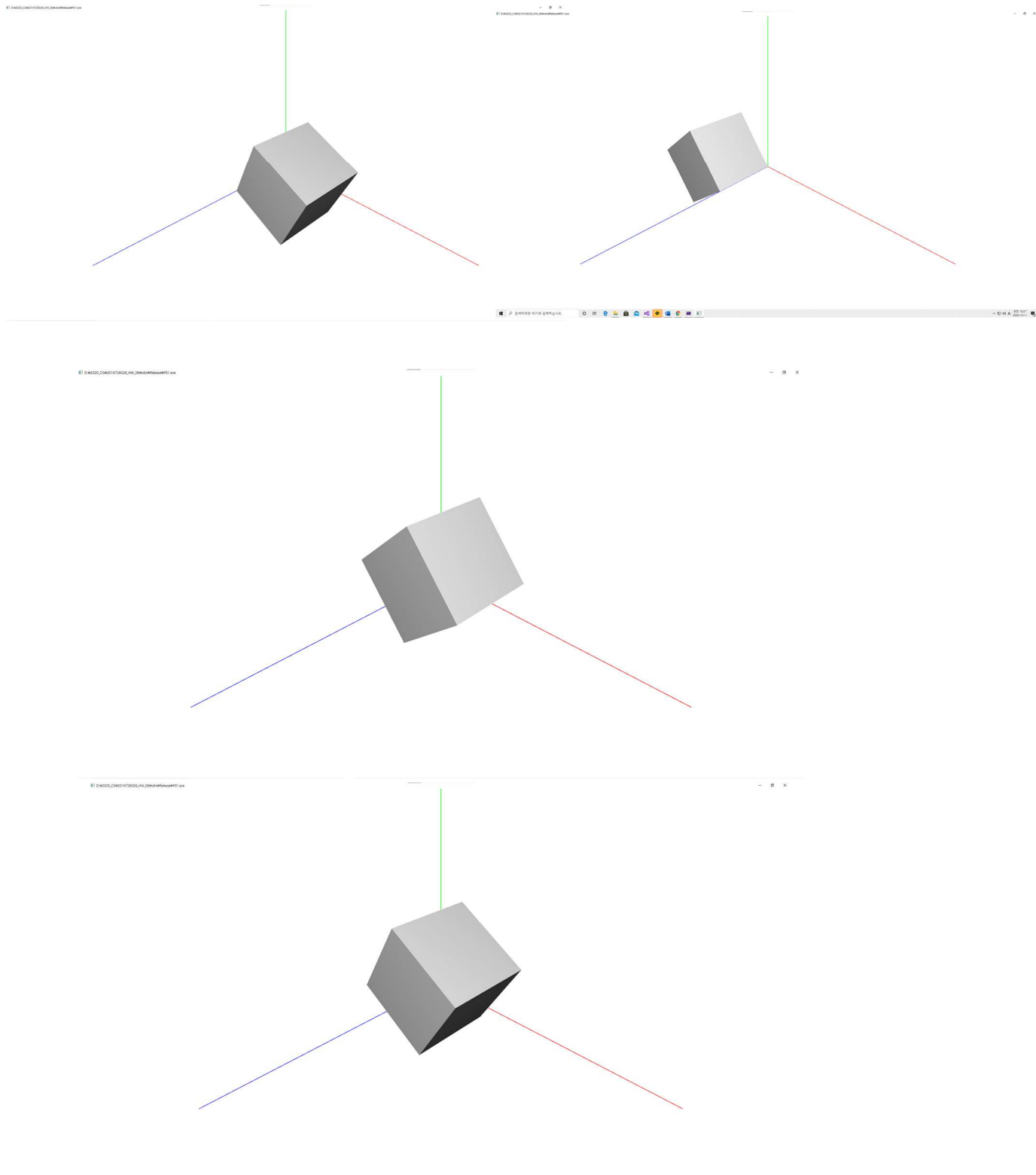
P06\_TRANSFORMATION

이민재 | Computer Graphics [심화전공실습 1] | 2020/10/11

	P01	P02	P03	E01	E02	Total
SCORE	1	1	1	1	1	5

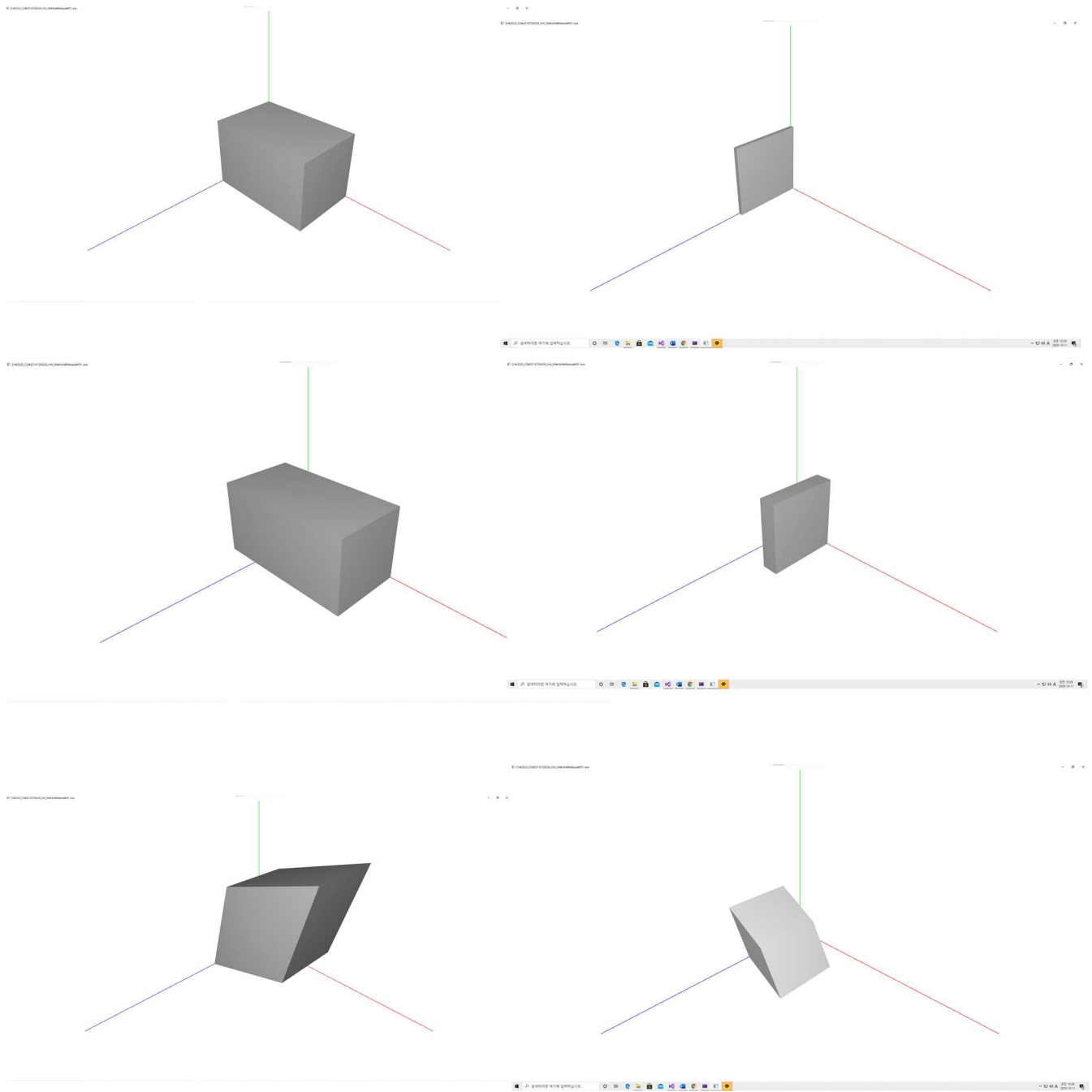
## Po1 (Three rotation examples)

<SNAPSHOT>

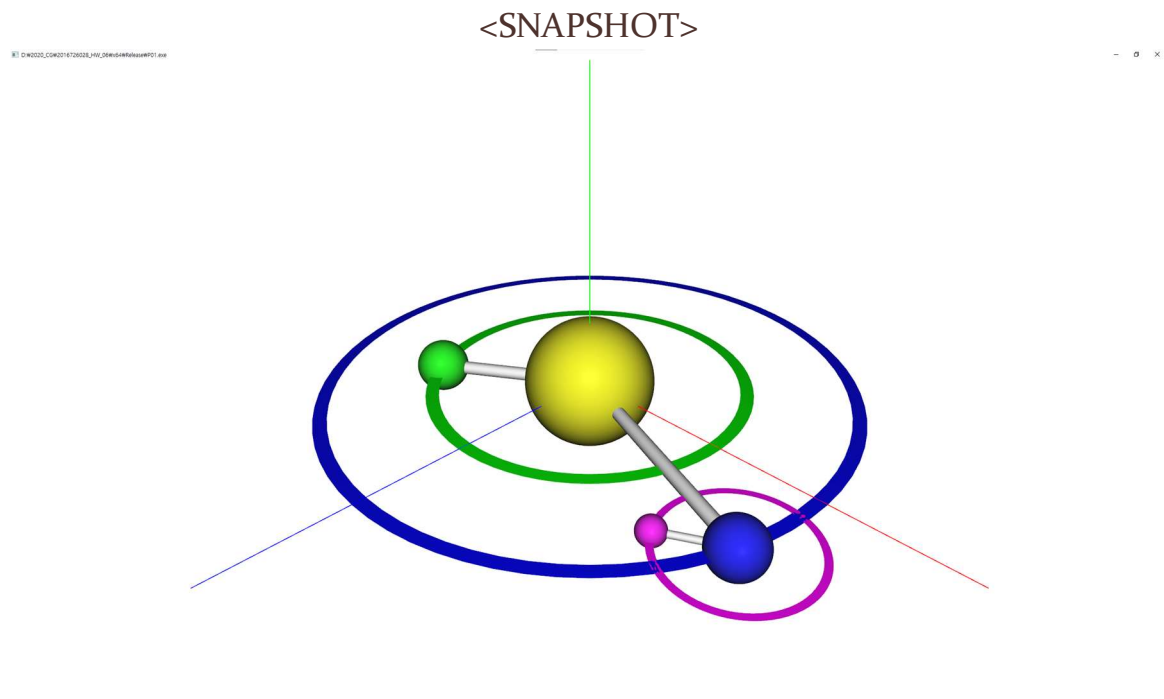


## Po2 (Three scaling examples)

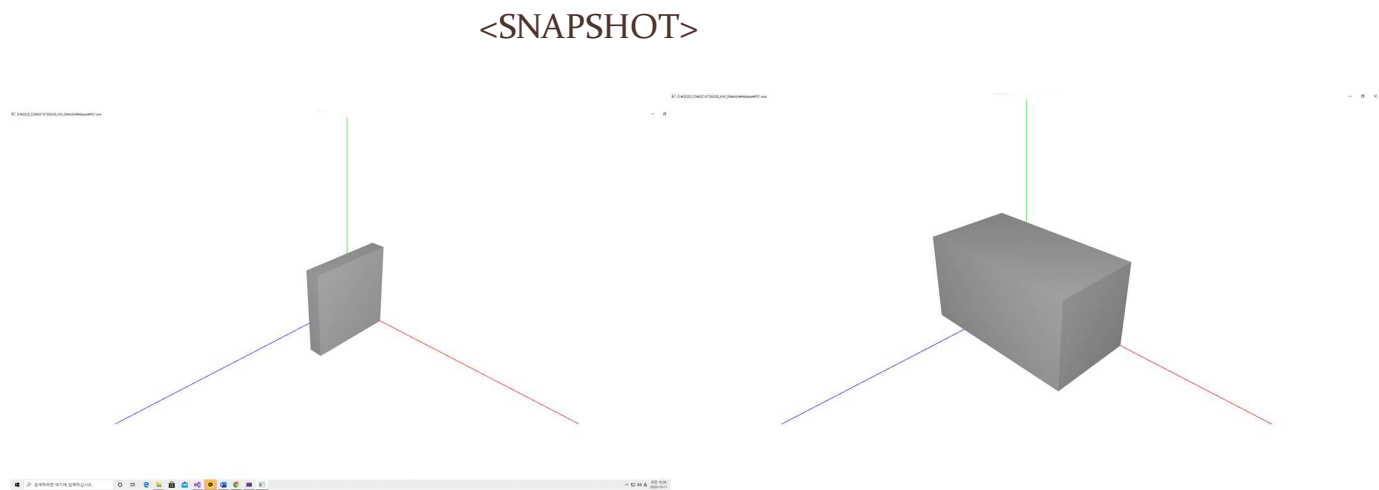
<SNAPSHOT>



## P03 (A solar system example)



## E01 (Scaling wrt a pivot with GLM)



### <EXPLANATION>

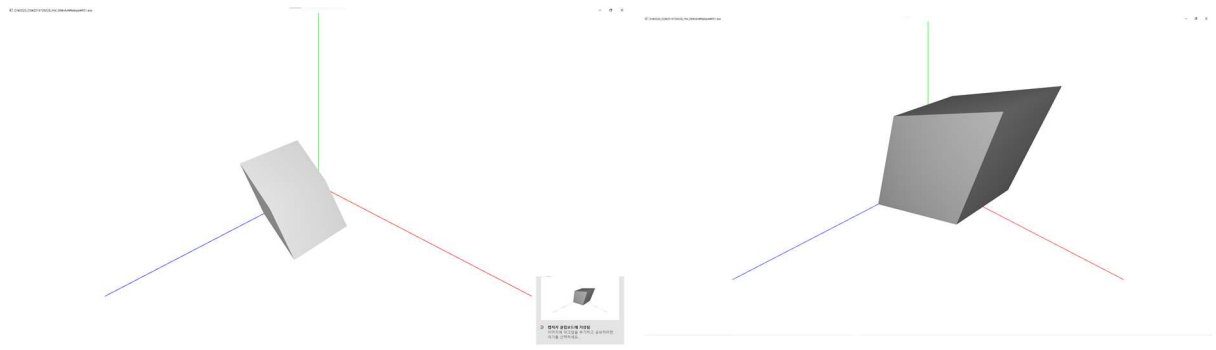
Vec3 타입의 axis 와 pivot 그리고 프레임에 따라 변하는 float s 까지는 OpenGL 을 활용한 예제와 동일하나, Translate - Scale - Translate 로 이루어지는 Scaling Pivot 의 과정을 다음과 같이 변형했다.

```
glm::vec3 axis(s, 1, 1);  
// Identity matrix  
glm::mat4 M(1.0);  
// M = M * translate(pivot)  
M = glm::translate(M, pivot);  
// M = M * scale(axis)... scale (detail::tvec3< T > const &v)  
M = glm::scale(M, axis);  
// M = M * translate(-pivot)  
M = glm::translate(M, -pivot)
```

기준에 glScalef(s, 1, 1)로 x 축방향으로 s 의 변화량을 주었던 것을 vec3 axis 로 만들어 glm::scale 에 이용하였다.

### Eo2 (. Scaling wrt a direction with GLM)

#### <SNAPSHOT>



#### <EXPLANATION>

위와 마찬가지로 각 Theta 의 크기나 s, direction, axis 의 크기와 값은 scalingDirection 예제와 동일하게 설정하였으나, Rotate - Scale - Rotate 로 이루어지는 일련의 과정을 다음과 같이 변형하였다.

```
glm::vec3 axis2(s, 1, 1);  
// Identity matrix  
glm::mat4 M(1.0);  
// M = M * rotate(angle, axis)  
M = glm::rotate(M, glm::radians(theta), axis); // Radians  
// M = M * scale(axis)  
M = glm::scale(M, axis2);  
// M = M * rotate(angle, axis)  
M = glm::rotate(M, glm::radians(-theta), axis); // Radians
```

Glm::scale 을 이용하여 s, 1, 1 방향으로 scaling 이 일어나고, theta 값을 라디안으로 변환시킨 값을 이용하여 회전, 원상복구가 일어나게끔 설정하였다.