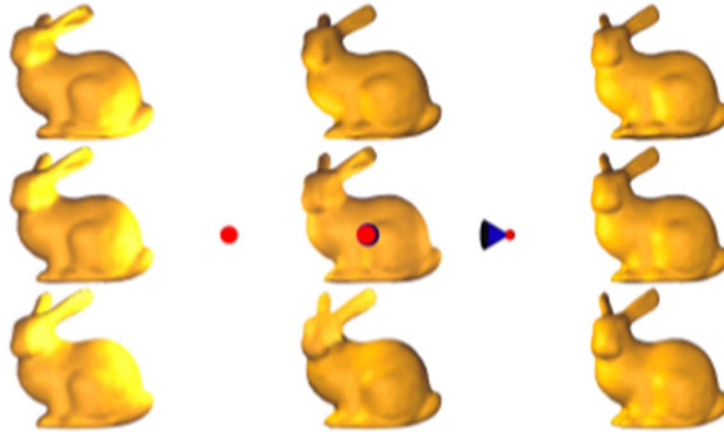


Computer Graphics



Linear Algebra

P04_LINEAR_ALGEBRA

이민재 | Computer Graphics [심화전공실습 1] | 2020/09/27

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

P01 (Vectors and vector operations (part 1 and 2))

<SNAPSHOT>

```
ca 명령 프롬프트
'ls'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

D:\2020_CG\2016726028_HW_04\Exe>P01.exe 1
3x1 vectors
a = vec3(0.000000, 0.000000, 0.000000)
b = vec3(3.000000, 2.000000, 1.000000)
a = vec3(1.000000, 2.000000, 3.000000)
a[0] = 1
a.z = 3

D:\2020_CG\2016726028_HW_04\Exe>P01.exe 2
Vector operations
a = vec3(1.000000, 2.000000, 3.000000)
b = vec3(3.000000, 2.000000, 1.000000)
a + b = vec3(4.000000, 4.000000, 4.000000)
a - b = vec3(-2.000000, 0.000000, 2.000000)
-a = vec3(-1.000000, -2.000000, -3.000000)
1.5*a = vec3(1.500000, 3.000000, 4.500000)
dot(a,b) = 10
cross(a,b) = vec3(-4.000000, 8.000000, -4.000000)
length(a) = 3.74166

D:\2020_CG\2016726028_HW_04\Exe>
```

P02 (Matrices and matrix operations (part 3 and 4))

<SNAPSHOT>

```
ca 명령 프롬프트
cross(a,b) = vec3(-4.000000, 8.000000, -4.000000)
length(a) = 3.74166

D:\2020_CG\2016726028_HW_04\Exe>P01.exe 3
3x3 matrices
A = mat3x3((0.000000, 0.000000, 0.000000), (0.000000, 0.000000, 0.000000), (0.000000, 0.000000, 0.000000))
A = mat3x3((1.000000, 0.000000, 0.000000), (0.000000, 1.000000, 0.000000), (0.000000, 0.000000, 1.000000))
B = mat3x3((1.000000, 0.000000, 0.000000), (2.000000, 1.000000, 0.000000), (3.000000, 0.000000, 1.000000))
B = mat3x3((1.000000, 0.000000, 0.000000), (2.000000, 1.000000, 0.000000), (3.000000, 0.000000, 1.000000))
3rd col of B = vec3(3.000000, 0.000000, 1.000000)
3rd col B = vec3(3.000000, 0.000000, 1.000000)
3rd row of B = vec3(0.000000, 0.000000, 1.000000)
1st row 3rd col of B = 3
1st row 3rd col of B = 3

D:\2020_CG\2016726028_HW_04\Exe>P01.exe 4
Matrix operations
A = mat3x3((1.000000, 0.000000, 0.000000), (0.000000, 1.000000, 0.000000), (0.000000, 0.000000, 1.000000))
B = mat3x3((1.000000, 0.000000, 0.000000), (2.000000, 1.000000, 0.000000), (3.000000, 0.000000, 1.000000))
A + B = mat3x3((2.000000, 0.000000, 0.000000), (2.000000, 2.000000, 0.000000), (3.000000, 0.000000, 2.000000))
A - B = mat3x3((0.000000, 0.000000, 0.000000), (-2.000000, 0.000000, 0.000000), (-3.000000, 0.000000, 0.000000))
-A = mat3x3((-1.000000, -0.000000, -0.000000), (-0.000000, -1.000000, -0.000000), (-0.000000, -0.000000, -1.000000))
A x B = mat3x3((1.000000, 0.000000, 0.000000), (2.000000, 1.000000, 0.000000), (3.000000, 0.000000, 1.000000))
transpose(B) = mat3x3((1.000000, 2.000000, 3.000000), (0.000000, 1.000000, 0.000000), (0.000000, 0.000000, 1.000000))
inverse(B) = mat3x3((1.000000, -0.000000, 0.000000), (-2.000000, 1.000000, -0.000000), (-3.000000, -0.000000, 1.000000))

inverse(B) * B = mat3x3((1.000000, 0.000000, 0.000000), (0.000000, 1.000000, 0.000000), (0.000000, 0.000000, 1.000000))

D:\2020_CG\2016726028_HW_04\Exe>
```

P03 (Matrix-vector multiplication and assembling (part 5))

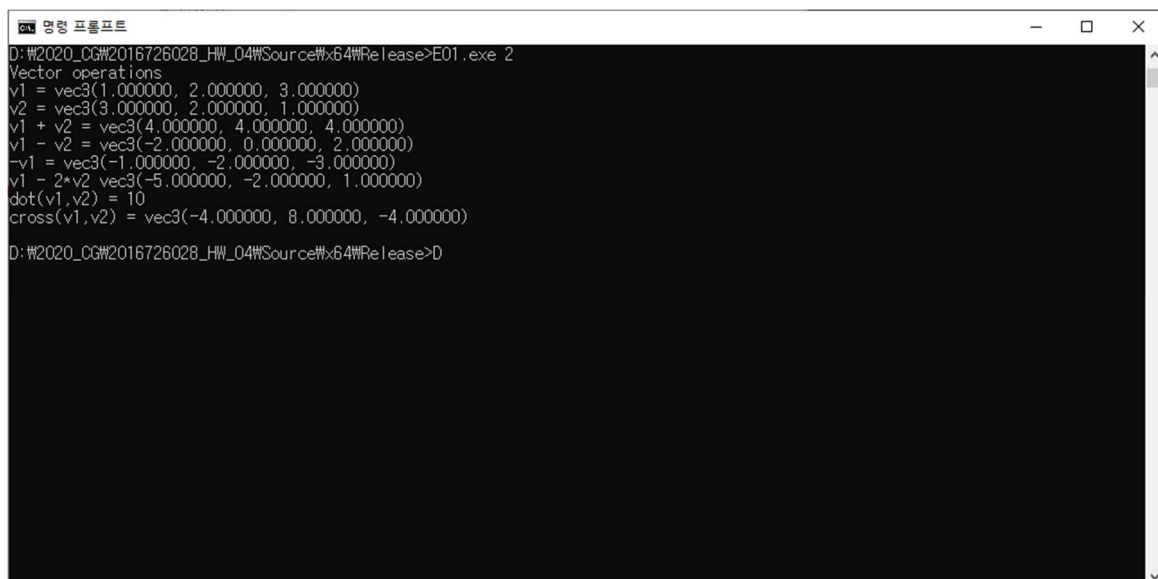
<SNAPSHOT>



```
D:\2020_CG\2016726028_HW_04\Exe>P01.exe 5
Matrix-vector multiplication and assembling
a = vec3(1.000000, 2.000000, 3.000000)
B = mat3x3((1.000000, 0.000000, 0.000000), (2.000000, 1.000000, 0.000000), (3.000000, 0.000000, 1.000000))
B x a = vec3(14.000000, 2.000000, 3.000000)
(1.0, a) = vec4(1.000000, 1.000000, 2.000000, 3.000000)
(a, 1.0) = vec4(1.000000, 2.000000, 3.000000, 1.000000)
(1.0, a) = vec4(1.000000, 1.000000, 2.000000, 3.000000)
C = mat4x4((1.000000, 0.000000, 0.000000, 0.000000), (2.000000, 1.000000, 0.000000, 0.000000), (3.000000, 0.000000, 1.000000, 0.000000), (0.000000, 0.000000, 0.000000, 1.000000))
D:\2020_CG\2016726028_HW_04\Exe>
```

E1 (Vector operations)

<SNAPSHOT>



```
D:\2020_CG\2016726028_HW_04\Source\x64\Release>E01.exe 2
Vector operations
v1 = vec3(1.000000, 2.000000, 3.000000)
v2 = vec3(3.000000, 2.000000, 1.000000)
v1 + v2 = vec3(4.000000, 4.000000, 4.000000)
v1 - v2 = vec3(-2.000000, 0.000000, 2.000000)
-v1 = vec3(-1.000000, -2.000000, -3.000000)
v1 - 2*v2 = vec3(-5.000000, -2.000000, 1.000000)
dot(v1,v2) = 10
cross(v1,v2) = vec3(-4.000000, 8.000000, -4.000000)
D:\2020_CG\2016726028_HW_04\Source\x64\Release>
```

<EXPLANATION>

기존 코드와 크게 차이점은 없어서 변수명과 출력명을 바꾸어서 진행하였다. $v1 - 2 \cdot v2$ 의 경우에만 상수(2.0f)에 주의하였다.

Eo2 (Matrix operations)

<SNAPSHOT>



```
cross(v1,v2) = vec3(-4.000000, 8.000000, -4.000000)
D:\#2020_CG\#2016726028_HW_04\Source\#x64\Release>E01.exe 4
Matrix operations
A1 = mat3x3((1.000000, 2.000000, 1.000000), (2.000000, 3.000000, 1.000000), (3.000000, 2.000000, 2.000000))
A2 = mat3x3((2.000000, 2.000000, 1.000000), (1.000000, 2.000000, 1.000000), (2.000000, 1.000000, 1.000000))
A1 + A2 = mat3x3((3.000000, 4.000000, 2.000000), (3.000000, 5.000000, 2.000000), (5.000000, 3.000000, 3.000000))
A1 - A2 = mat3x3((-1.000000, 0.000000, 0.000000), (1.000000, 1.000000, 0.000000), (1.000000, 1.000000, 1.000000))
-A1 = mat3x3((-1.000000, -2.000000, -1.000000), (-2.000000, -3.000000, -1.000000), (-3.000000, -2.000000, -2.000000))
A1 - 2*A2 = mat3x3((-3.000000, -2.000000, -1.000000), (0.000000, -1.000000, -1.000000), (-1.000000, 0.000000, 0.000000))
A1 x A2 = mat3x3((9.000000, 12.000000, 6.000000), (8.000000, 10.000000, 5.000000), (7.000000, 9.000000, 5.000000))
A2 x A1 = mat3x3((6.000000, 7.000000, 4.000000), (9.000000, 11.000000, 6.000000), (12.000000, 12.000000, 7.000000))
A1 x v1 = vec3(14.000000, 14.000000, 9.000000)
A2 x v2 = vec3(10.000000, 11.000000, 6.000000)
D:\#2020_CG\#2016726028_HW_04\Source\#x64\Release>
```

<EXPLANATION>

Glm 의 경우 column-major representation 방식으로 행렬을 표현하기 때문에 이에 유의하여 A1, A2 를 삽입하였고, 주어진 보기대로 계산하여 출력하였다.