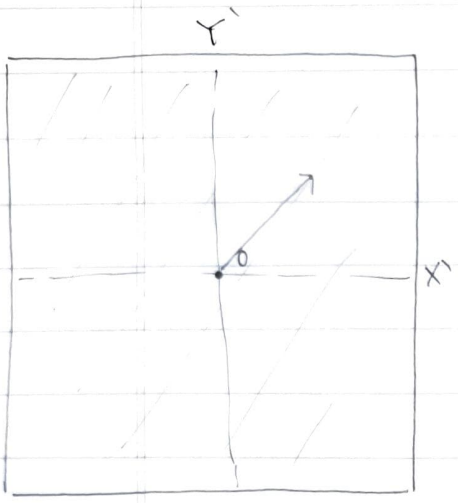
9월 3-4주차 레포트 기계공학부 이원목

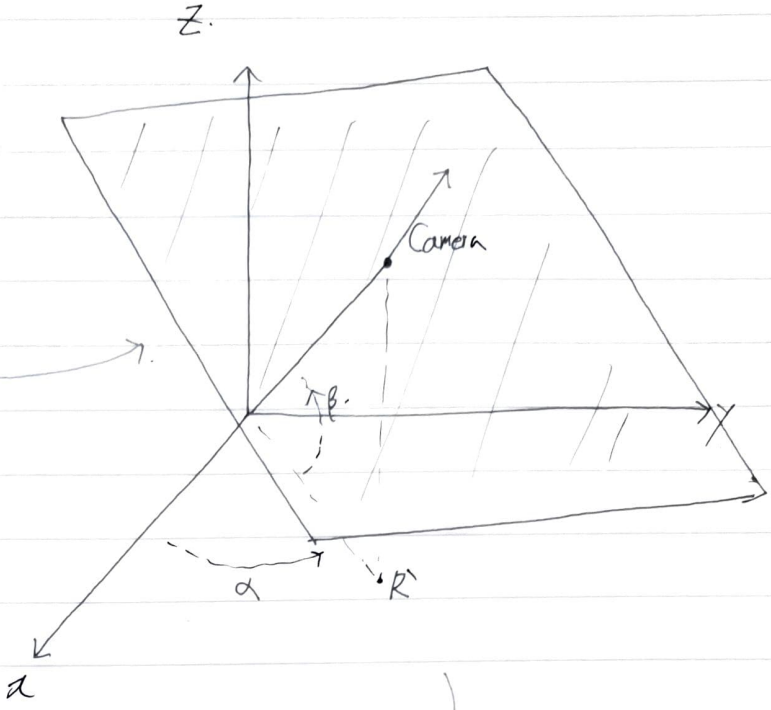
전체 코드
https://github.com/Winteradio/_OPENGL_make_3D_Spring

○ 카메라의 회전.

↳ 마우스를 활용



↳ 사용자의 프로그램



마우스 왼쪽클릭 시

드래그의 벡터

↳ X`Y` 공간 상의 벡터를

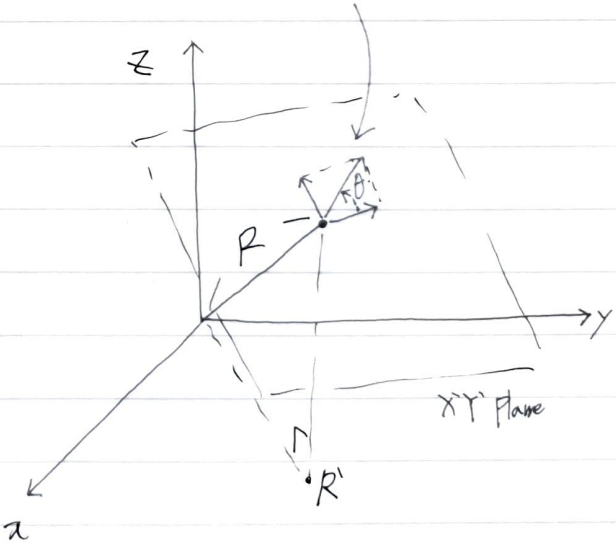
XYZ 3차원 계의 좌표로 변환.

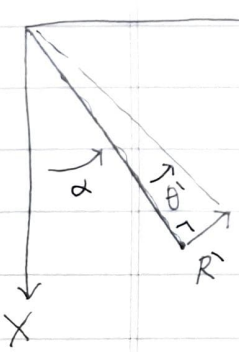
(구형 좌표계 or 직교좌표계)

↳ 회전을 각도를 위주로 사용하기 때문에

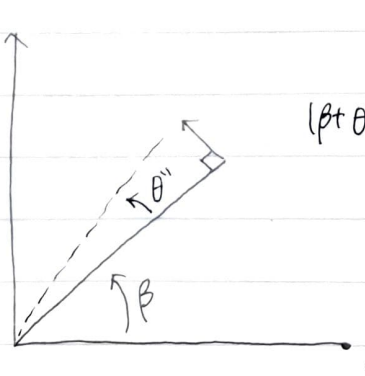
구형 좌표계가 용이

• Camera : (R, α, β) 구형좌표계





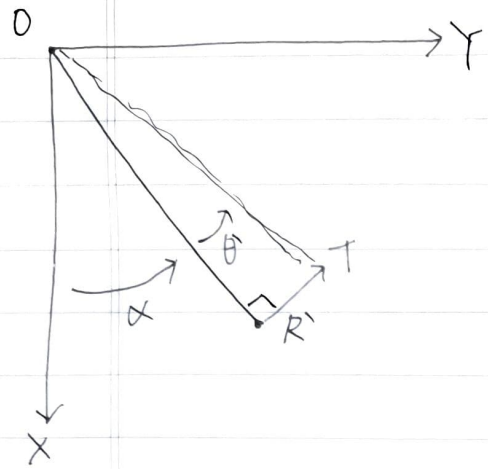
$(\alpha+\theta`)$ = 새로운 α의 각도

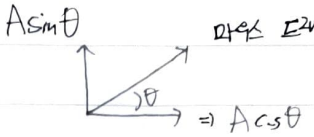


$(\beta+\theta``)$ = 새로운 β의 각도

계산하는 법

↳ 새로운 α.





$\theta` = \tan^{-1}\left(\frac{RT}{OR`}\right)$

$RT = A\cos\theta$ (어떻게 마스 $\vec{A}$의 Acosθ가 $\vec{OT}$ 인가?)

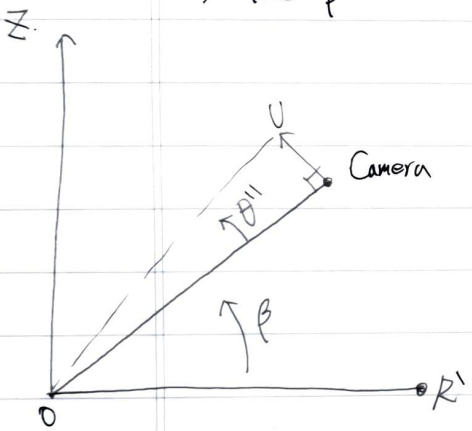
↳ 카메라의 $\vec{U_P}$가 <0, 0, 1> 이기 때

$OR` = R\cos\beta$ ∥ XY-plane ∥ $\vec{A_{XY}}$

$\alpha_{new} = \alpha + \theta`$

$= \alpha + \tan^{-1}\left(\frac{A\cos\theta}{R\cos\beta}\right)$

↳ 새로운 β



$\theta`` = \tan^{-1}\left(\frac{Cam_{era}U}{OC_{amera}}\right)$

$Cam_{era}U = A\sin\theta$

$OC_{amera} = R$

$\beta_{new} = \beta + \theta``$

$= \beta + \tan^{-1}\left(\frac{Cam_{era}U}{OC_{amera}}\right) = \beta + \tan^{-1}\left(\frac{A\sin\theta}{R}\right)$

※ 주의 사항

→ X`Y` 에서의 $\vec{A}$의 크기과 변경되는 θ`, θ``을 적절히 조절해야 사용하기 용이하다.

→ 위의 계산은 카메라 초점으로 회전 (X)

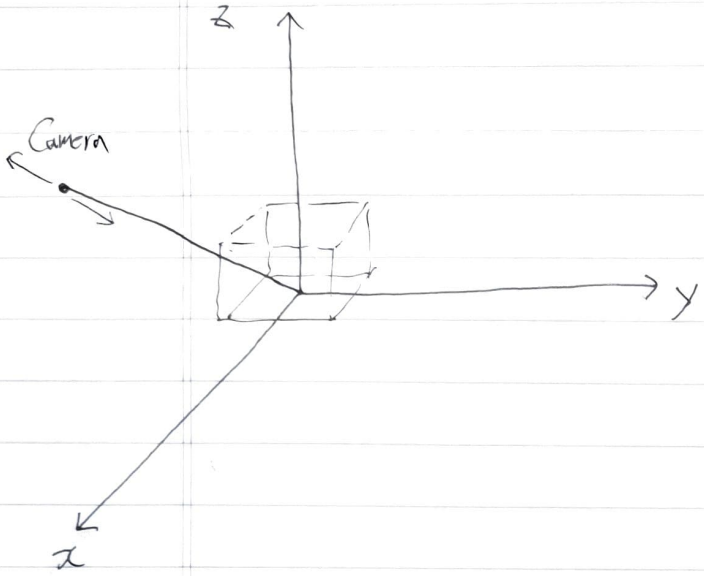
〃 원점으로 회전 (O)

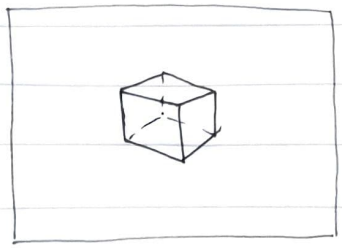
조절(카메라)

원점 기준으로 회전 하는 것이 필요

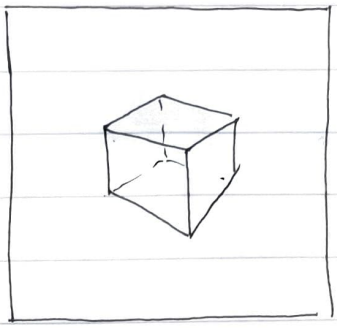
○ 카메라 확대

↳ 마우스 스크롤 활용

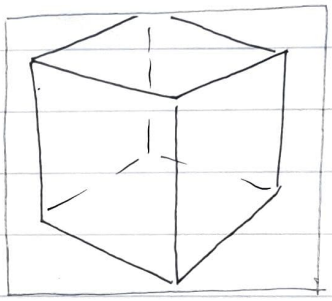




← 멀어지는 시야



기존 시야
(프로그램상 위치는)



↘ 가까워지는 시야

계산하는 법

↳ 스크롤 방향 ↻ =>

R(카메라 반경)

$R_{new} = R / 2$

↳ 스크롤 방향 ↺ =)

$R_{new} = R * 2$

* 주의 사항

→ 거리가 너무 멀어지거나 가까워질시 =) 최대, 최소 값 지정 필요.

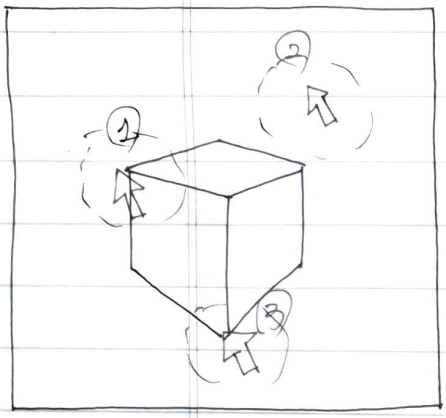
사용자 착오 ↲ ↳ 화면에 깨진다

(화면을 보고 무언가 하기 어렵다)

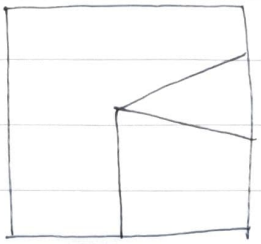
→ 회전과 마찬가지로 초점이 아닌 원점기준 (0) =) 초점 기준으로 축소 확대 필요

⁂ 가능하다면

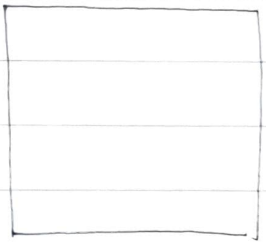
Inventor 화면



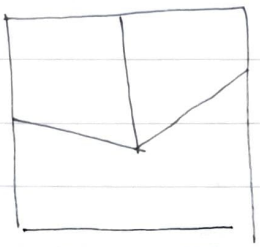
① 확대/축소



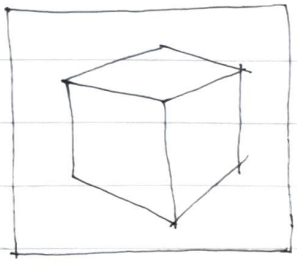
② 〃



③ 〃



④ 돌아감



↳ 마우스 커서 있는 방향으로 초점이동

↳ 초점에 대한 확대

↳ 축소할시에도 커서가 있는 방향을 초점으로 멀어지게

⇒ 어떻게 되는지 생각하기

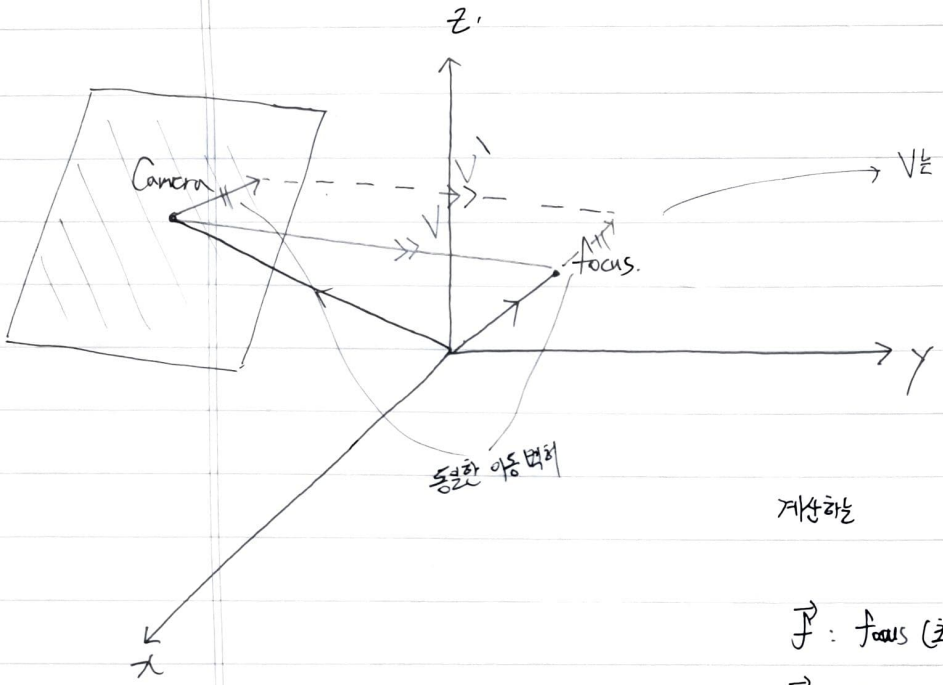
↳ ① 확대할 때만 커서로 초점이동
축소시는 원래 초점으로

② 확대/축소 둘다 커서로 초점이동

○ 카메라 이동

↳ 마우스 오른쪽 활용.

① 벡터의 차를 일정하다를 활용



→ V는 일정하다: 초점과 카메라가 같은 방향의 벡터로 움직인다.

계산하는

$\vec{F}$: focus (초점)의 위치 벡터

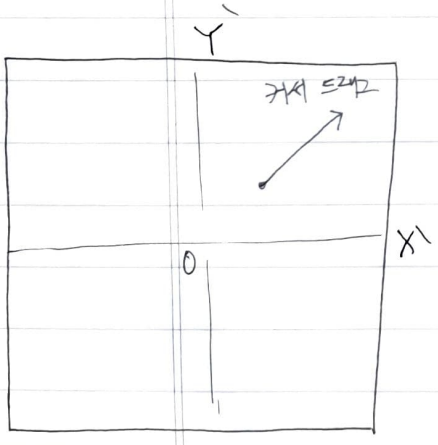
$\vec{C}$: Camera (카메라)의 위치 벡터

$\vec{V} = -\vec{C} + \vec{F} = \vec{V'}$

$\vec{C}_{new} = \vec{A} + \vec{C}$

$\vec{V'} = -\vec{C}_{new} + \vec{F}_{new} = -\vec{C} + \vec{F}$

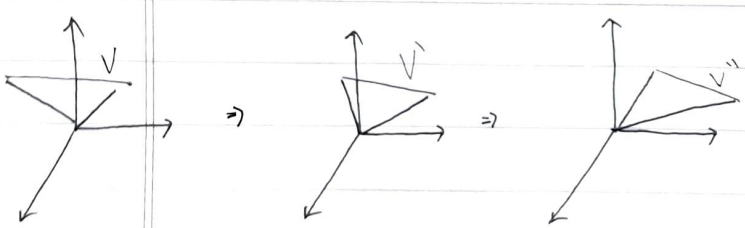
↳ $\vec{F}_{new} = \vec{F} + \vec{A}$



사용자가 본 프로젝션 화면

*주의 사항.

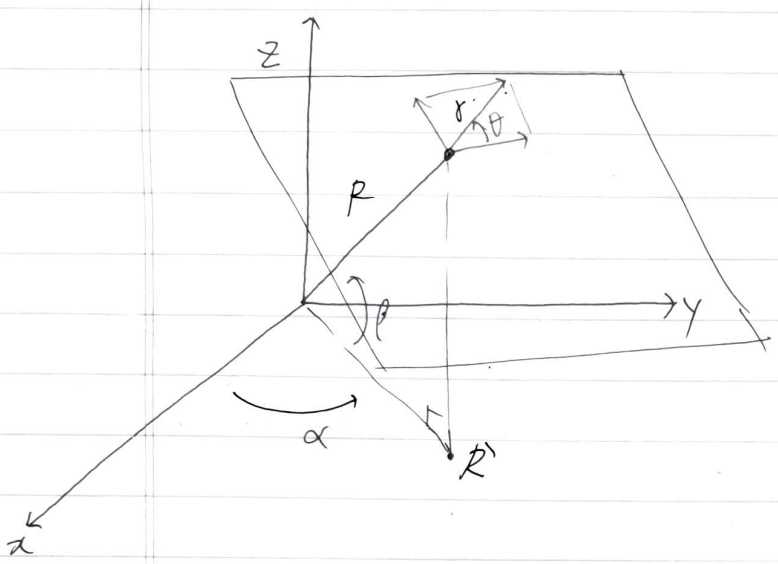
카메라의 이동은 10 ms 마다 Time Func을 통해 진행된다.

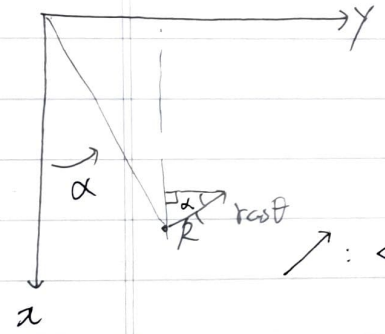


↳ 지금현재는 평면상 이동처럼 보이는 이동이 아니다 매우 큰 구면상에서의 이동처럼 보인다.

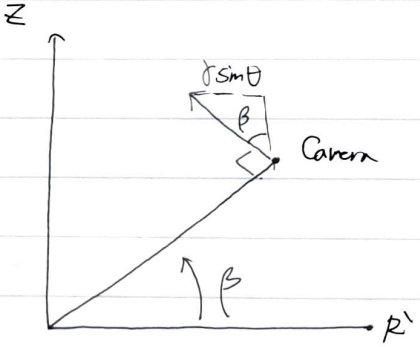
② 드래그 벡터의 직교좌표계의 변환 식, 초점과 카메라의 이동.

↳ ①과 마찬가지로 'V'는 일정하다 라는 전제 하에서 진행된다.





↗ : $\langle -r\cos\theta\sin\alpha,\ r\cos\theta\cos\alpha,\ 0\rangle$

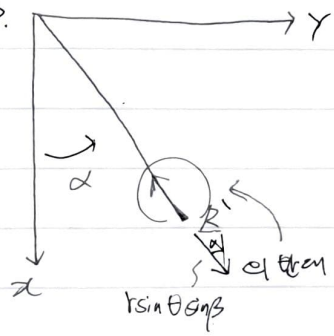


↖ : $\langle\ ?\ ,\ ?\ ,\ r\sin\theta\cos\beta\rangle$

↙
XY 평면으로

↗ 의 직교좌표계 벡터

$\langle -r\cos\theta\sin\alpha - r\sin\theta\sin\beta\cos\alpha,\ r\cos\theta\cos\alpha - r\sin\theta\sin\beta\sin\alpha,\ r\sin\theta\cos\beta\rangle$



↖ : $\langle -r\sin\theta\sin\beta\cos\alpha,\ -r\sin\theta\sin\beta\sin\alpha,\ 0\rangle$

*주의 사항

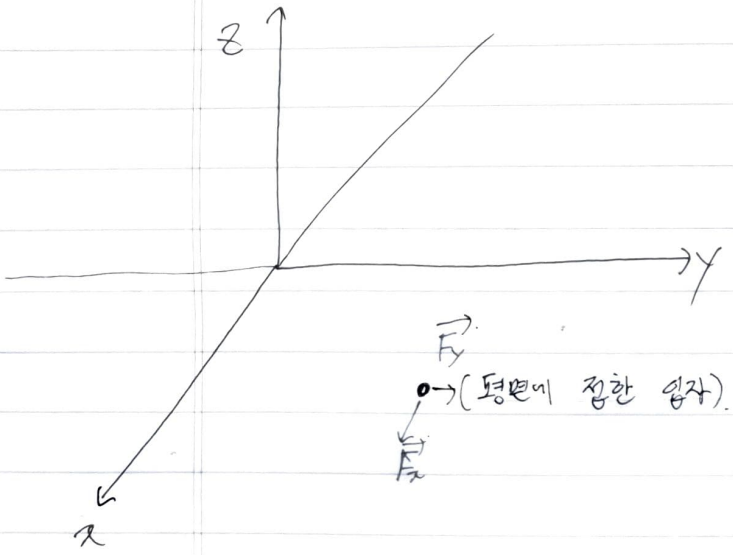
↳ 현재조건 값 식 대입할 시,

구형 ⇒ 직교 ⇒ 구현 에서 오류 발생

초점과 카메라 위치 변경

(다행히도 구형 ⇒ 직교 과정에선 원하는 움직임 구현, 드래그 시 다시 되돌아가는 문제점)

○ 마찰력 계산.



① $\vec{F_x}, \vec{F_y} \leq \vec{f_s}$ (정지 마찰력)

↳ $\mu_s \cdot N$

N: $\vec{F_z}$ μ_s: 정지 마찰 계수

$\vec{F_z} = \vec{F_{spring}} + \vec{F_{gravity}}$

② $\vec{F_x}, \vec{F_y} > \vec{f_s}$ 일시

$\vec{F_{x\,new}} = \vec{F_x} - \vec{f_k}$ ↺ $\mu_k \cdot N$ 운동

$\vec{F_{y\,new}} = \vec{F_y} - \vec{f_k}$

N: $\vec{F_z}$

μ_k: 운동 마찰 계수

○ 조금 더 입체적으로 하기 위한 것

(1) 입자들의 크기

D: 입자의 프로그램상에서의 크기. C: Camera의 위치

P: 입자의 위치. D_0: 입자의 실제 크기

↳ $D = k \dfrac{D_0}{|\overline{PC}|}$ ↳ D_0의 기준은 k에 따라 변한다.

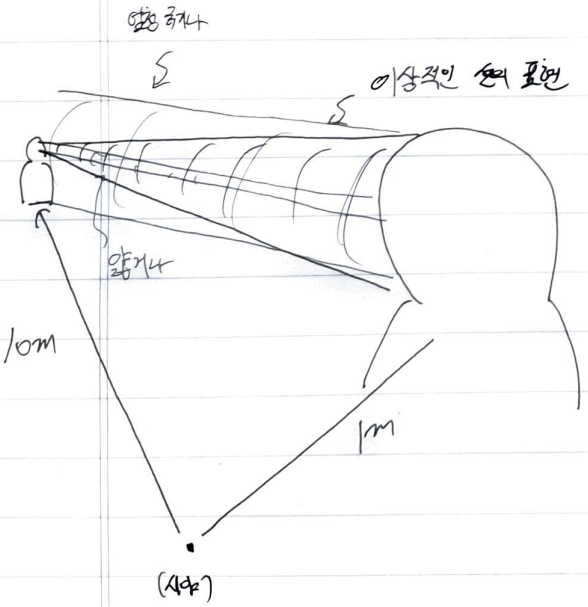
(상수)

↳ $\overline{PC}$의 거리가 1일 기준으로 잡는다.

$k = \overline{P_1C_1} = 2$

(2) 스프링 선들의 굵기 & 좌표계 선들의 굵기

↳ 선의 굵기를 어떻게 지정할 것인가?



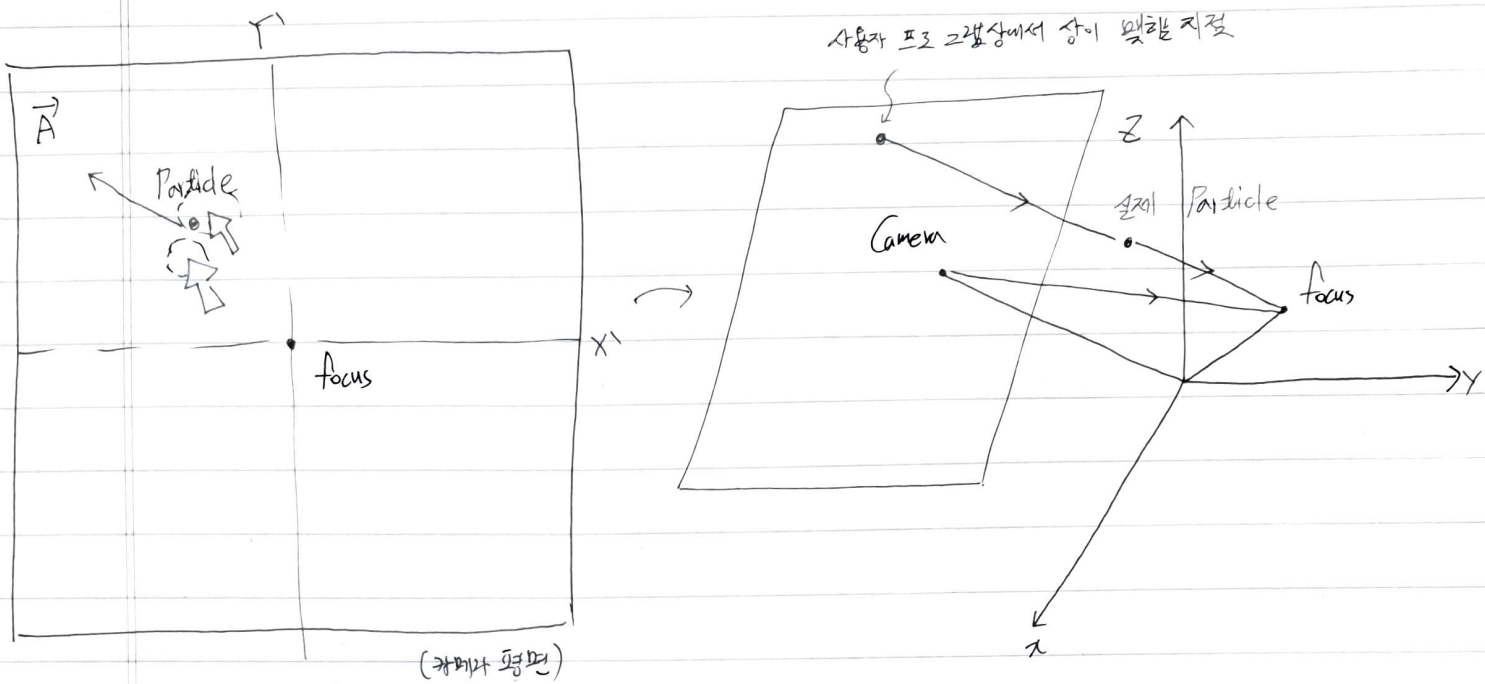
⇒ 하지만 프로그램상에선 선의 굵기는 한번에 정해진다.

⇒ 선분과 함수들로 선을 구성할 것인가?

↳ 내가 보고자 하는 선이 표현이 있는가?

↳ 새로운 방식을 찾아야 한다.

○ 마우스로 점 이동하기



↓

◯ : 마우스 커서 반경에 들어올시 사용자가 입자에 제한 적용 가능

↳ 일반적인 드래그의 경우, 마우스의 이동과 겹칠수 있으니

① 반경 안에 점이 있을시 → 입자 이동

" 없을시 → 카메라 이동

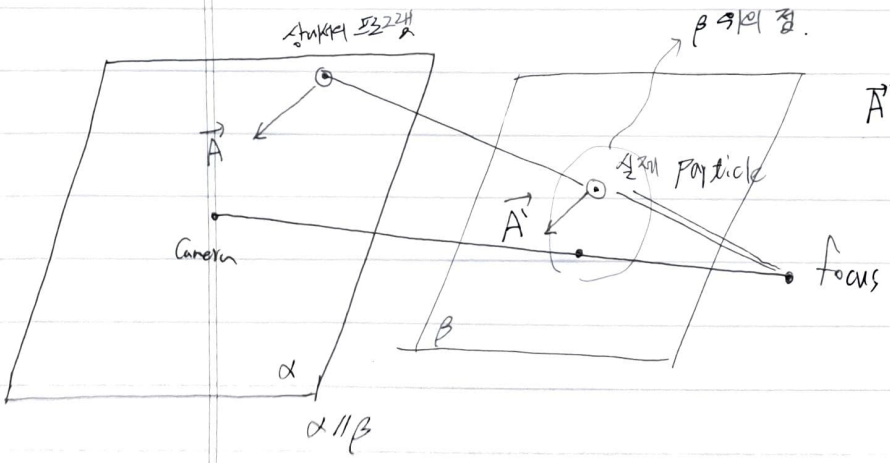
② 특정 버튼 누를시 ⟶ 반경 안에 점 있을시 → 입자이동

" 안누를시 ⟶ 카메라 이동.

+ 카메라 평면과 가장 가까운 점을 사용.

$\vec{A}$: 마우스 드래그 벡터

↓



$\vec{A'}$ 실제 particle 이 움직이는 벡터

★ 주의 사항.

어떻게 거리와 평면 이동,

마우스 반경을 측정할 것인가?