

Assignment #1 Report

공과대학 컴퓨터공학부 2020-19422 권신영

- 구현한 것

움직이는 자동차를 구현하였습니다. 크게 몸체(body), 바퀴 4개(wheel), 몸체 중앙의 기둥(center_cylinder), 회전하는 기둥(head) 및 프로펠러(propeller)로 구성되어 있습니다.

움직임은 바퀴의 회전, 프로펠러 및 기둥의 회전과 이동, 자동차 전체의 이동, 3가지입니다. 이들은 render.py의 update에 구현되어 있습니다.

```
if self.animate:
    if shape.group == "car":
        if shape.type == "Wheel":
            rotate_angle = 0.7 * dt
            rotate_axis = Vec3(0, 1, 0)
            rotate_mat = Mat4.from_rotation(angle=rotate_angle, vector=rotate_axis)

            shape.transform_mat = shape.transform_mat @ rotate_mat

        if shape.type == "Propeller":
            rotate_angle = 5 * dt
            rotate_axis = Vec3(1, 0, 0)

            rotate_mat = Mat4.from_rotation(angle=rotate_angle, vector=rotate_axis)
            shape.transform_mat = shape.transform_mat @ rotate_mat

        if shape.type == "Head":
            rotate_angle = 0.7 * dt
            rotate_axis = Vec3(0, 1, 0)

            rotate_mat = Mat4.from_rotation(angle=rotate_angle, vector=rotate_axis)
            shape.transform_mat = shape.transform_mat @ rotate_mat

        if shape.type == "Propeller" or shape.type == "Head":
            y_offset = 2 * dt * math.sin(time.time() * math.pi * 2) # 1초 주기로 왕복
            wing_translation = Mat4.from_translation(vector=Vec3(x=0, y=y_offset, z=0))
            shape.transform_mat = wing_translation @ shape.transform_mat

    global_translation = Mat4.from_translation(vector=Vec3(x=-0.7 * dt, y=0, z=0))
    shape.transform_mat = global_translation @ shape.transform_mat
```

회전 속도는 dt 에 곱해진 계수에 따라 다르며, 프로펠러 및 기둥의 이동은 주기적인 왕복 운동입니다.

Primitives.py에선 원통(Cylinder)와 프로펠러(Propeller) Class를 추가로 구현하였습니다.

- 실행 방법

Conda 가상환경 `snu_graphics` 활성화 및 `main.py` 실행

(필요시) `conda init`

(필요시) `source ~/.zshrc`

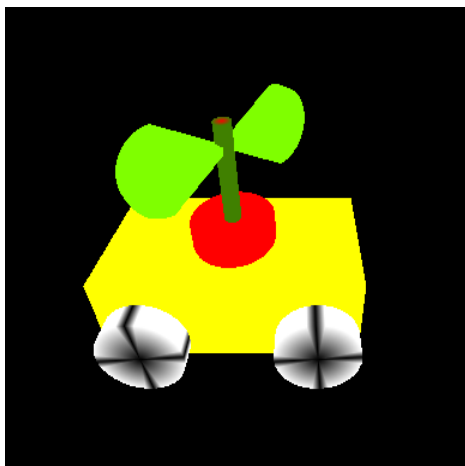
`Conda activate snu_graphics`

`Python main.py`

- 실행 결과

스크린샷 및 영상을 따로 첨부하였고(demo1, demo2), 리포트에 스크린샷을 추가로 첨부하겠습니다.

Git에 제공해주신 코드와 동일하게, 스페이스바 토글로 정지 및 움직임을 제어할 수 있습니다.



- 추가 자료

추가로 import하여 사용한 것은 없고, git에서 제공해주신 skeleton code를 토대로 하였습니다. 그래서 `environment.yaml`도 동일합니다.