

Computer Graphics Assignment 2

2019-18499 김준혁

1. 구현 내용

primitives.py

CustomGroup을 상속받은 PointGroup, LineGroup, TriangleGroup, DerivedLineGroup, DerivedSurfaceGroup, DerivedSubdivisionLineGroup, DerivedSubdivisionSurfaceGroup, NullGroup class를 추가하였습니다. 이들은 도형 추가 및 이들의 조작을 용이하게 하기 위해 세분화하여 만든 클래스입니다.

- PointGroup : obj 파일의 점들의 집합 그룹
- LineGroup : obj 파일의 선들의 집합 그룹
- TriangleGroup : obj 파일의 면을 삼각형으로 표시한 것의 그룹
- DerivedLineGroup : obj에서 유래한 Bezier/B-Spline Surface의 선들의 집합 그룹
- DerivedSurfaceGroup : obj에서 유래한 Bezier/B-Spline Surface의 면들의 집합 그룹
- DerivedSubdivisionLineGroup : obj에서 유래한 Catmull-Clark Subdivision의 선들의 집합 그룹
- DerivedSubdivisionSurfaceGroup : obj에서 유래한 Catmull-Clark Subdivision의 면들의 집합 그룹

PointSet, LineSet, TriangleSet, DerivedLine, DerivedSurface, DerivedSubdivisionLine, DerivedSubdivisionSurface 도형 클래스를 추가하였습니다. 도형의 조작을 위하여 이들을 모두 나누어 구현하였습니다. 각각은 위의 그룹들에 대응되어 도형으로 추가가 됩니다.

- PointSet, LineSet, TriangleSet은 obj 파일의 내용을 그대로 표현하는 데에 사용됩니다.
- DerivedLine, DerivedSurface는 Spline Surface를 나타내기 위한 용도로, 10 * 10 개의 점으로 표현됩니다.
- DerivedSurface, DerivedSubdivisionLine는 Subdivision을 나타내기 위한 용도로, 원본 obj 파일에서 유래될 Subdivision의 꼭짓점 개수를 미리 구하여 그리드 점을 편집할 때 같이 변화가 가능하도록 하였습니다.

main.py

obj 파일을 불러오는 load_object 함수를 추가하였습니다. obj 파일은 v~f가 각각의 도형을 의미하며, 이는 f 다음에 v가 나오는 경우 다른 도형으로 인식한다는 뜻입니다. 이 함수에서 미리 각 도형의 꼭짓점, 선, 출력 용 삼각형, 면, 시작 꼭짓점 인덱스를 반환합니다.

add_pointSet, add_lineSet, add_triangleSet, add_derivedSurface, add_derivedLine, add_derivedSubdivisionLine, add_derivedSubdivisionSurface 함수를 추가하였습니다. 이들은 primitives.py에서 정의한 각 도형들을 추가해주는 함수입니다.

프로그램을 실행시킨 이후, 콘솔창을 통해 불러올 obj 파일의 위치를 입력받아 불러오고, 각 도형 및 그것들로 유래될 Spline Surface와 Subdivision의 베이스(모든 점을 0,0,0 으로 하여 표시가 안되게 하며, 이후 vertex의 좌표와 연결 상태를 바꿀 수 있도록)를 추가하도록 하였습니다.

render.py

camera_move 함수를 추가하여 키보드 및 마우스로 시점을 바꾸어주는 기능을 추가하였습니다. 이에 필요한 변수를 __init__에서 초기화하였으며, update 함수를 통해 호출됩니다. 불러온 obj 파일의 위치 또한 저장합니다.

on_draw 함수에서 스크린에 표시되는 점의 크기와 선의 굵기를 지정해주었습니다.

control.py

__init__에 컨트롤에 필요한 변수를 추가하였습니다.

- selected_point : 마우스 왼쪽 클릭으로 선택된 점을 저장합니다.
- mouse_move_camera : 마우스 오른쪽 클릭으로 카메라를 움직이는 상태를 저장합니다.
- dragging : 마우스 드래그 상태를 저장합니다.
- surface_mode : 표시할 Object / Spline Surface / Subdivision 종류를 저장합니다.

Bezier Spline Surface = 1, B-Spline Surface : 2, Catmull-Clark Subdivision : 3

비표시 상태에서는 각 상태의 음수값, 최초 실행시 0

- vertex_attach : Q를 눌러 다른 점에 붙이는 상태인지를 저장합니다.
- vertex_integer : E를 눌러 좌표값을 정수로 반올림할지를 저장합니다.
- grid_visible : Z를 눌러 그리드를 표시할지를 저장합니다.

on_key_press, on_key_release, on_mouse_press, on_mouse_release, on_mouse_drag
에 키를 추가하였습니다.

- W : 카메라 시점 기준 앞으로 이동
- S : 카메라 시점 기준 뒤로 이동
- A : 카메라 시점 기준 왼쪽으로 이동
- D : 카메라 시점 기준 오른쪽으로 이동
- LShift : 카메라 시점 기준 아래로 이동
- Space : 카메라 시점 기준 위로 이동
- F : Bezier Spline Surface 표시 모드 활성화/비활성화 (toggle)
- G : B-Spline Surface 표시 모드 활성화/비활성화 (toggle)
- R : Catmull-Clark Subdivision 표시 모드 활성화/비활성화 (toggle)
- Q : 점 붙이기 모드 활성화/비활성화 (Pressing)
- E : 점 좌표 정수로 반올림 모드 활성화/비활성화 (Pressing)
- Z : 그리드 표시 모드 활성화/비활성화 (toggle)
- 마우스 왼쪽 클릭, 드래그 : 점 이동
- 마우스 오른쪽 클릭, 드래그 : 카메라 시점(카메라 위치는 고정) 타겟 변화

point_projection 함수를 추가하여 3D 좌표의 점이 스크린 좌표 어디에 위치하는지 반환합니다.

is_clicked 함수를 추가하여 점이 선택되었는지(T/F) 확인합니다.

find_point 함수를 추가하여 왼쪽 클릭으로 선택된 점을 반환합니다.

move_vertex 함수를 추가하여 마우스 왼쪽 클릭, 드래그를 통해 어떤 꼭짓점이 어떤 좌표로
이동될 것인지를 update_vertex 함수에 넘겨주도록 하였습니다.

update_vertex 함수를 추가하여 선택된 꼭짓점이 이동 좌표로 바뀌도록 하며, 이에 따라

obj/Spline Surface/Subdivision의 선, 면 또한 이에 따라 업데이트되도록 하였습니다.

near_point 함수를 추가하여 Q를 눌렀을 때 스크린 상 근처 점 위치를 확인하여 이동시키는
중의 점이 어떤 점에 붙을 수 있는지 확인, 가능한 경우 해당 위치로 좌표를 업데이트
(update_vertex) 시켜줍니다.

integer_point 함수를 추가하여 E를 눌렀을 때, 해당 점의 좌표값을 정수로 반올림하여 업데이트
시켜줍니다.

exit_work, save_object 함수를 추가하여 프로그램에서 esc를 누르면 종료하며, 저장할 지를 확인하고 어떤 위치에 저장할 지를 콘솔창에 입력하여 각 도형을 모두 저장하도록 하였습니다. default는 불러온 파일과 동일합니다.

surface_mode_control 함수를 추가하여 Raw Object / Bezier / B-Spline / Catmull-Clark 또는 그리드 표시를 바꿀 수 있는 모드를 각 도형 visible을 통해 조정하도록 하였습니다.

bernstein_poly 함수를 추가하여 번스타인 다항식 계수를 반환하도록 하였습니다.

add_bezier_surface 함수를 추가하여 Bezier Spline Surface의 점 위치를 조절하여 표시할 수 있도록 만들어주었습니다.

bspline_matrix 함수를 추가하여 B-Spline Surface의 점의 위치를 구하는데 사용하기 위한 행렬과의 계산값을 반환하도록 하였습니다.

add_bezier_surface 함수를 추가하여 B-Spline Surface의 점 위치를 조절하여 표시할 수 있도록 만들어주었습니다.

subdivision_surface 함수를 추가하여 Catmull-Clark Subdivision을 통한 결과 Surface의 점의 위치와 선 및 면을 구하여 표시할 수 있도록 만들어주었습니다. 해당 Subdivision은 1회만 이루어진 결과를 계산합니다.

2. 실행 방법

Base code와 같이 python3 main.py로 실행되며, 표시할 obj 파일의 위치를 콘솔창에 지정해주어야 프로그램이 정상적으로 작동합니다 (default : SurfaceMeshWSplineWgrid_test.obj). grid_test.obj 는 테스트용으로 사용한 4 * 4 그리드입니다.

프로그램이 obj 파일을 불러온 이후에는 아래의 키를 이용해 편집이 가능합니다.

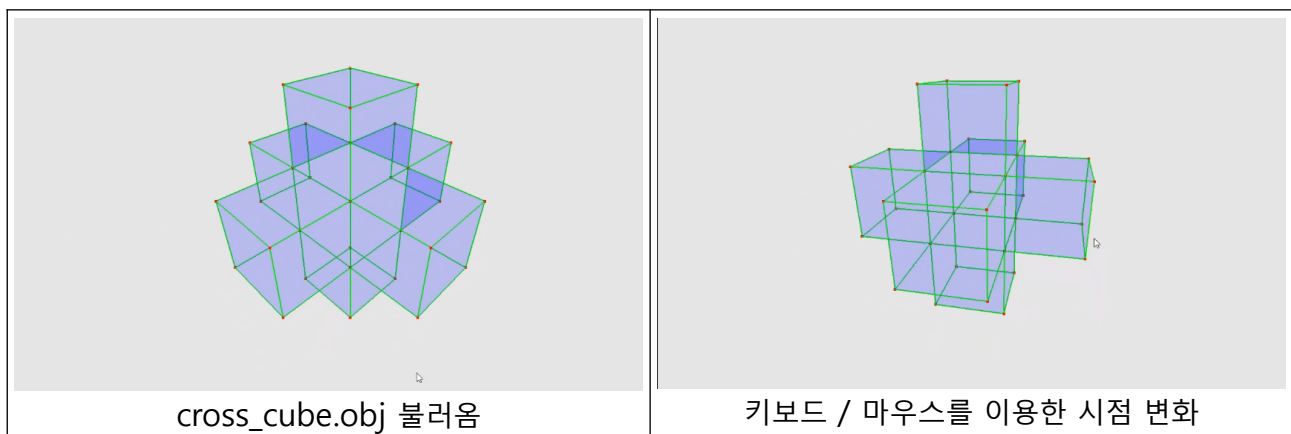
(구현 내용에 서술한 것과 동일합니다.)

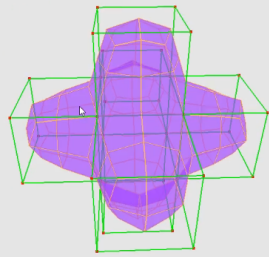
- W : 카메라 시점 기준 앞으로 이동
- S : 카메라 시점 기준 뒤로 이동
- A : 카메라 시점 기준 왼쪽으로 이동
- D : 카메라 시점 기준 오른쪽으로 이동

- LShift : 카메라 시점 기준 아래로 이동
- Space : 카메라 시점 기준 위로 이동
- F : Bezier Spline Surface 표시 모드 활성화/비활성화 (toggle)
- G : B-Spline Surface 표시 모드 활성화/비활성화 (toggle)
- R : Catmull-Clark Subdivision 표시 모드 활성화/비활성화 (toggle)
- Q : 점 붙이기 모드 활성화/비활성화 (Pressing)
- E : 점 좌표 정수로 반올림 모드 활성화/비활성화 (Pressing)
- Z : 그리드 표시 모드 활성화/비활성화 (toggle)
- 마우스 왼쪽 클릭, 드래그 : 점 이동
- 마우스 오른쪽 클릭, 드래그 : 카메라 시점(카메라 위치는 고정) 타겟 변화

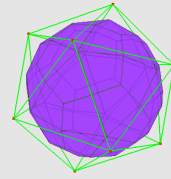
ESC 키를 눌러 종료를 할 수 있으며, 저장할 지 여부를 콘솔창에서 묻습니다. 오른쪽 위 X를 눌러 종료하거나 저장 여부에서 n을 입력할 경우 저장하지 않습니다. 저장 여부는 y / n 으로 입력받으며, y를 입력하여 저장할 경우에 저장 위치를 입력받습니다. default는 불러온 obj 파일과 동일합니다. 저장이 완료되면 안정적으로 프로그램을 종료합니다.

3. 실행 결과

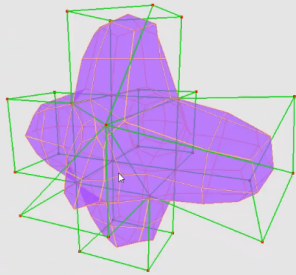




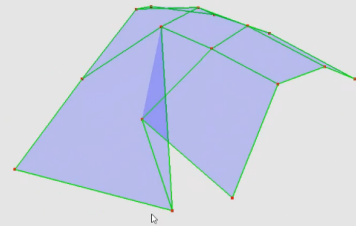
Catmull-Clark Subdivision Surface
(cross_cube.obj)



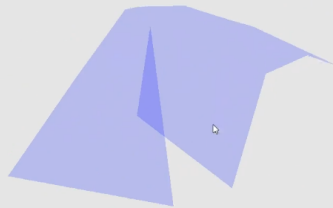
Catmull-Clark Subdivision Surface
(cross_cube.obj)



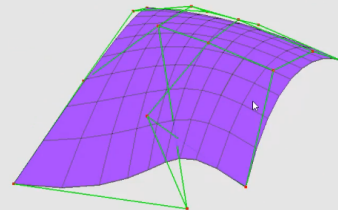
마우스 클릭, Q를 이용하여 점 편집 및 점 붙이기



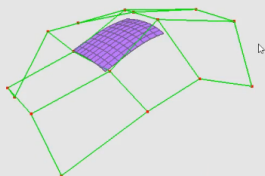
마우스 클릭, E를 이용하여 점 편집, 정수 반올림



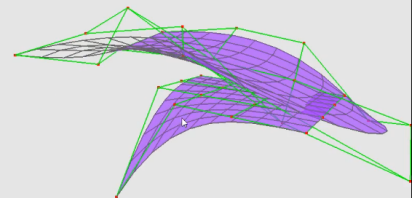
Z를 눌러 그리드 비표시



F를 눌러 Bezier Spline Surface 표시



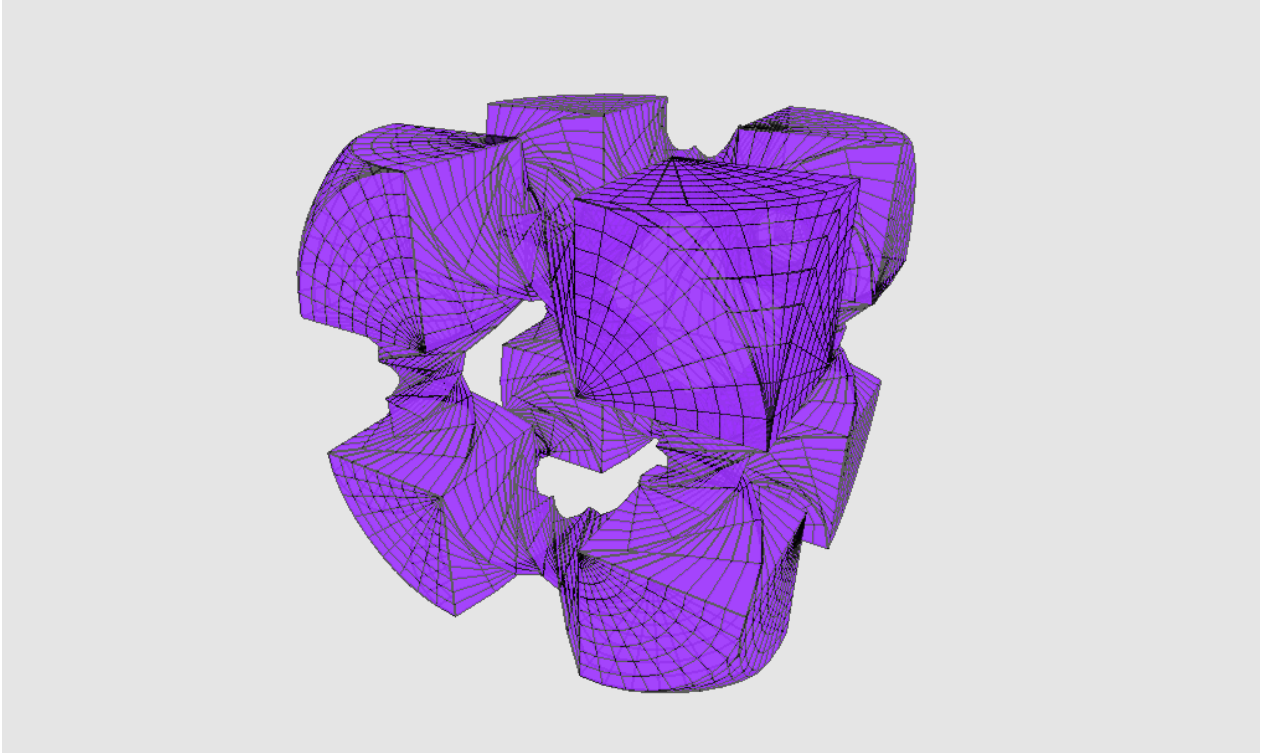
G를 눌러 B-Spline Surface 표시



다수의 도형(그리드 2개)을 불러온 모습

키보드 / 마우스 / 콘솔 입력을 통한 상세한 프로그램 실행 결과는 동봉된 Demo.mp4 파일을
확인해주시기 바랍니다. (Participate Track 2)

Model for Participating Track 1



Track 1 에 참가하기 위해 제작한 모델(SurfaceMeshWSplineWswirl_cube.obj)입니다.

80 개의 Bezier Spline Surface를 이용하여 제작하였습니다. F와 Z를 눌러 Bezier 표시 및 그리드 비표시 모드로 나타낸 결과입니다.

4. 참고 사이트

pyglet 공식 문서 - <https://pyglet.readthedocs.io/en/latest/>