

obj Viewer & Drawing a Hierarchical Model

Computer Graphics Project 2 - 컴퓨터소프트웨어학부 2017029343 김기환

Which requirements you implemented

1. Manipulate the camera in the same way and draw the reference grid plane.

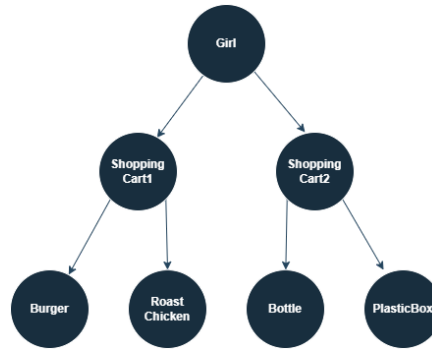
- Project1의 코드를 사용하여 동일하게 카메라를 이동시키고 Reference grid plane을 그렸음.
- **Trouble Shooting** : 카메라의 위치가 Y축 최상단이나 최하단에 도착하여 넘어가는 과정에서 x,z 값이 반전되는데 이 과정에서 부자연스러운 모습이 연출된다.

2. Single mesh rendering mode

- ObjectInfo라는 클래스를 정의하여, obj 파일에서 읽어온 정점의 위치, 노말 값, 면을 이루는 정점의 번호 등을 객체 내부 배열에 담을 수 있도록 하였음.
- glfwSetDropCallback 함수를 이용하여 obj 파일이 Window에 drag-and-drop 되면, parsingObject 함수를 호출하여 함수 내부에서 해당 obj 파일의 내용을 Parsing하여 정점의 위치, 노말 값, 면을 이루는 정점의 번호들을 배열에 저장하고 Phong Shading, Phong Illumination을 위해 vertex normal을 계산하여 이 또한 배열에 저장하였음. 이후, infoObj 함수를 호출하여 해당 obj 파일의 정보를 terminal(stdout)에 출력하도록 하였음.
- 코드의 통일성을 위해서 Single mesh rendering mode에서도 계층 구조를 도입, 하나의 루트노드만 존재하는 계층 구조를 만들고 이를 렌더링하는 식으로 구현하였음. DrawObject 함수에 해당 노드 객체와 ObjectInfo 객체를 전달하여 이를 바탕으로 렌더링을 진행하였음.
- obj 파일마다 model의 크기가 다르게 렌더링되는데, 이를 해결하기 위해서 정점들의 x, y, z 값 중 절댓값이 가장 큰 값을 읽어와 해당 값으로 모든 정점들을 나누어 obj 파일에 따라 크기가 다르게 렌더링되는 문제를 해결하였음.

3. Animating hierarchical model rendering mode

- Node 클래스를 정의하고 이를 바탕으로 Hierarchical model rendering을 구현하였음. 3-Level, 각각의 노드는 2개의 자식노드를 갖도록 하기 위해 총 7개의 노드를 6개의 obj 파일을 통해서 구현하였음.



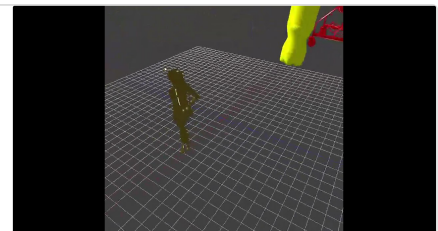
마트에 장을 보러가는 아줌마 컨셉

- 프로그램 실행 시 미리 지정해둔 obj 파일을 Parsing하여 각각 ObjectInfo 객체 내부에 obj 파일의 정보를 담고, 각각의 Node 객체와 함께 DrawObject의 인자로 전달, 해당 함수 내부에서 렌더링을 진행하여 Window에 출력하도록 하였음. → **프로그램 실행 시 Parsing과 Rendering 과정으로 인해 30초에서 1분 정도 시간이 소요될 수 있음**
- 프로그램 실행 초기에는 Reference grid plane만 출력이 되고, Window에 obj파일을 드래그 앤 드롭하게 되면 해당 obj 파일만 렌더링하며, 'H' 키를 누르게 되면 미리 지정해놓은 obj 파일들이 Hierarchical model rendering 되어 출력된다. 이후 다른 obj 파일을 드래그 앤 드롭하게 되면 동일하게 해당 obj 파일만 렌더링 되게 된다.

HYU CSE4020 - ComputerGraphics Project2

ComputerGraphics Project2 - Hierarchical Model Rendering
2017029343 김기환

 <https://youtu.be/nVgygrpWNdg>



4. Lighting & Etc

- Phong Illumination과 Phong shading을 위해서 shader 프로그램을 수정하였으며, 여러 uniform 변수를 도입하여 객체마다 서로다른 색상을 가질 수 있도록 하였으며, 광원의 위치를 y축 기준으로 회전하도록 구현하였음.
- 프로그램 실행 시 solid mode로 작동하며, 'z' 키 입력시 wireframe mode로 작동 되도록 함, (solid mode ↔ wireframe mode)