*REPORT*

Computer Graphics mid-term

[벽면 고르고, 설문조사 하기]



|  |  |
| --- | --- |
| **담당 교수** | **송인식 교수님** |
| **학과** | **소프트웨어학과** |
| **팀원/학번** | **이지언\_32173478** |
| **제출일** | **2020년 6월 19일**  **Git:**<https://github.com/easyearn77/computer_graphics_2020_1_project> |

목차

1. 프로젝트 설명
2. 코드 분석
3. 결과
4. 소감
5. 프로젝트 설명

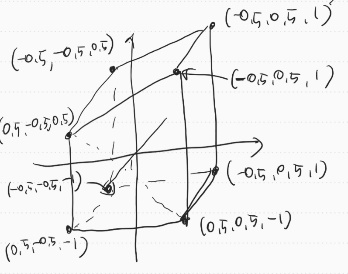
[프로젝트 배경&기대효과]

* 건축 프로젝트의 경우, 투자자, 조합원, 분양예정자 등 여러 이해 관계자들이 존재한다. 다수의 관계자들에게 건물 시뮬레이션을 보여주고, 원하는 건물 색상, 재료에 대해 피드백을 받아서 적용해야 한다. 기존의 피드백 방법은 건축 설계사나 건축사들이 시뮬레이션을 발표하면 피드백을 받는 형태이다. 하지만, 요즘과 같은 전염병이 유행하는 상황에서 다수의 관계자들을 한 곳에 모아 피드백을 받는 것은 매우 어렵다. 따라서 가장 활용성이 좋은 웹에서 건물 시뮬레이션을 동작하고, 사용자의 시선(camera)에 따라서 건물의 모양을 자세히 보고, 다양한 색상과 텍스쳐를 적용하는 방안을 생각해 보았다.
* 이 프로젝트로 인해, 건축가들은 정확한 피드백을 받을 수 있으며, 사용자 또한 자신이 원하는 질감을 다양하게 적용해 볼 수 있다면 비전문가들도 자재를 잘 몰라도 보기에서 고르면 되기 때문에 쉽게 선택할 수 있다는 장점이 있다. 따라서 웹페이지에서 모델링 결과를 편리하고 자세하게 볼 수 있다면, 결과적으로 건축가와 건축사에 대한 좋은 평가와 높은 신뢰도를 얻을 수 있다고 생각한다.

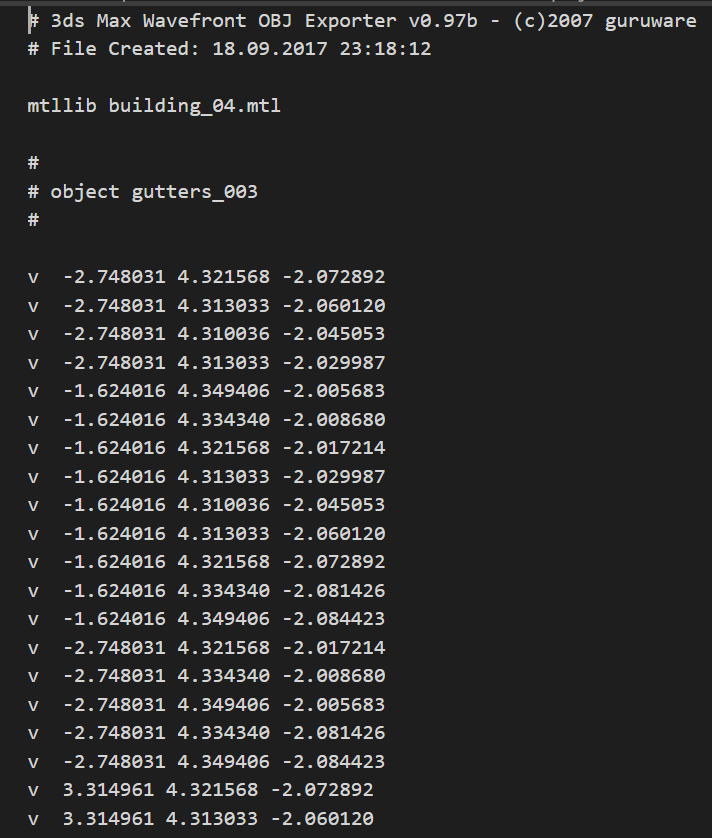
[제공기능]

* 웹페이지를 처음 열게 되면 건물을 정면으로 보도록 사용자 뷰를 제공한다
* 처음에 객체를 회전하는 방법을 고안했지만, 그것보다 three.js에서 제공하는 사용자 시선(camera)를 변경시키는 방안이 더욱 적합하다고 생각하여 이를 적용했다.
* 위의 색상 메뉴 버튼(바)를 보고 원하는 색상을 클릭하면, 건물의 색상을 변화할 수 있다.
* 처음에 건물의 질감도 버튼으로 변화시키려고 했으나, 색상에 적합한 질감이 있기 때문에 이를 적용하기 위해, 질감은 색상에 맞는 것으로 변화하도록 적용했다.

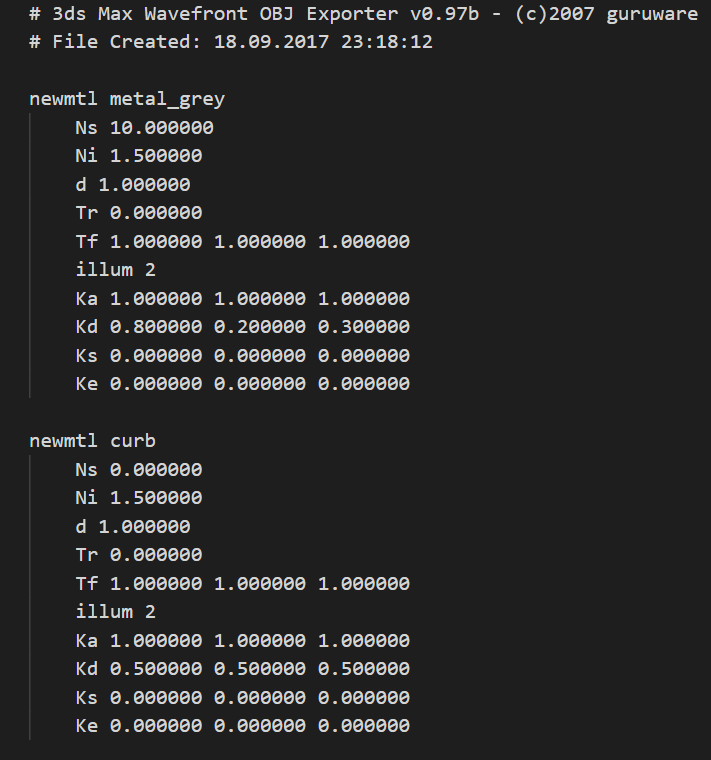
[프로젝트 진행방안]

-처음에 빌딩들을 모두 정점으로 만들어서 webgl로만 렌더링하려고 했다. 하지만 건물에는 창문, 몰드, 출입문 등 다양한 부속품들이 존재해서 모든 버텍스를 지정하기 힘들었고, 건물과 창문만 버텍스로 만들었을 때,  이런식으로 만들었는데, 너무 볼품없었다. 그래서 프로젝트가 진행되지 않을 것 같아서 기존의 3d모델링을 찾아 three.js로 진행하게 되었다.

- three.js로 진행하던 중, 3d모델 뼈대는 obj파일로 불러오면 되는 것으로 파악했다. Obj파일을 분석한 결과,

->building\_04.obj

material부분은 mtl파일로 불러오고, vertex지정 위주로 진행함을 알게 되었다. 따라서 이는 우리가 그동안 배웠던 vertex buffer로 추가되어 렌더링 됨을 알게 되었고, texture 지정을 위해서는 mtl파일을 수정하면서 프로젝트를 진행했다.

--> mtl 파일

- mtl 분석결과, fragment들을 그룹으로 만들어서 같은 material의 그룹들을 한번에 색상과 텍스처에 대한 지정을 하고 있었다.

-세부적인 attribute를 보면,

🡪 Ns: 광택의 척도. 값이 높을수록 거칠기가 낮아짐

🡪Ni: material에 대한 exponent를 지정합니다.

🡪Tr: 투명도

🡪illum: 조명

🡪Ka: ambient 반사도

🡪Kd: diffuse 반사도

🡪Ks: specular반사도

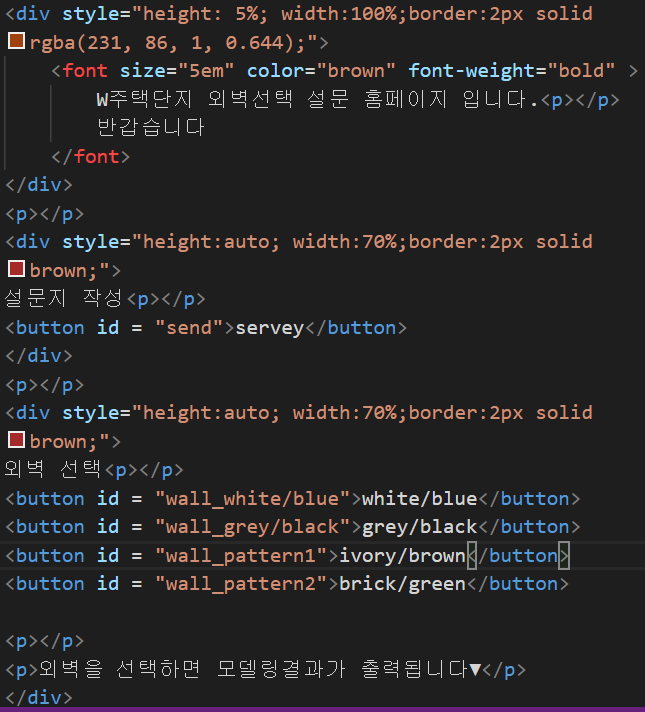
-->이러한 반사도에 대한 정의는 phong reflection을 배우면서 같이 배웟었다.



* 매핑할 텍스쳐는 이런 코드를 해당 material밑에 작성하여 매핑할 수 있다. 이때, 가로 세로 비율이 같아야 한다.

1. 코드 분석

**[ch\_building.html]**

* head: 화면의 background color를 지정하여 좀 더 실제 홈페이지와 비슷하게 제작했다.
* 모델링 화면이 들어갈 canvas는 가로세로 70%로 반응형으로 제작했다.
* Body:
* 
* 홈페이지에 들어갈 문구들과 외벽선택에 대응하는 버튼, 설문지 작성 페이지로 넘어 갈 수 있는 하이퍼링크 버튼을 만들었다.
* Three.js를 이용하기 위해 필요한 js파일들과 orbitControl을 위한 js파일, 3d모델을 위한 objLoader, 텍스쳐를 매핑하기 위한 MTLLoader의 js파일들을 포함했으며, 마지막에 실제로 동작이 일어나는 building.js를 스크립트로 포함시켰다.

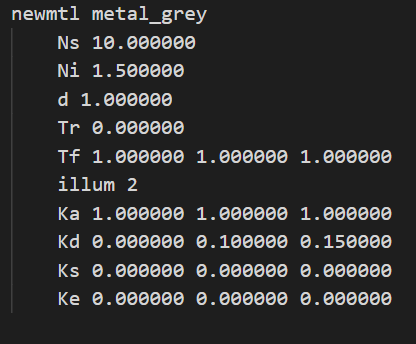
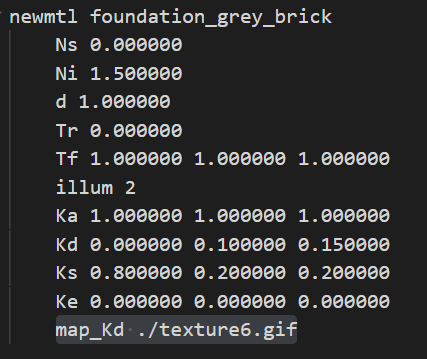
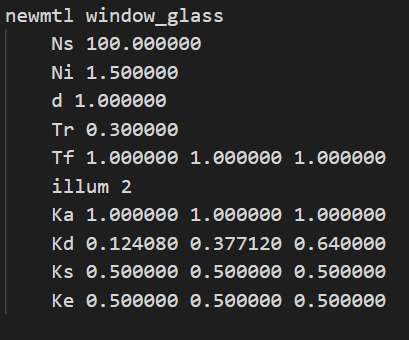
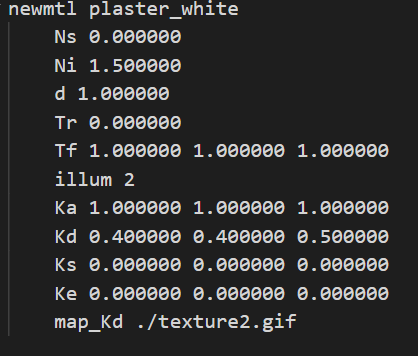
**[building.js]**

* 화면에 그리고자 하는 scene객체를 설정한다. Background의 색을 설정하면 원하는 색을 해당 렌더링 한 화면의 배경으로 입힐 수 있다.
* 시선에 해당하는 camera, objloade를 넣을 객체, webgl렌더링을 위한 렌더 객체를 설정합니다.
* **addLight():** light를 추가하는 함수로, light를 총 4개를 설정합니다. ->HemisphereLight: 하늘색에서 지상색으로 색상이 희미해 지는 장면 바로 위에 위치한 광원입니다.

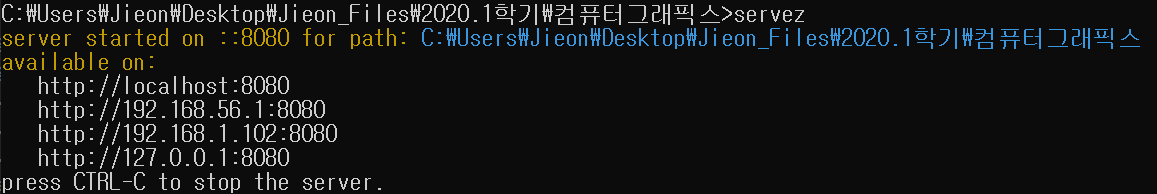
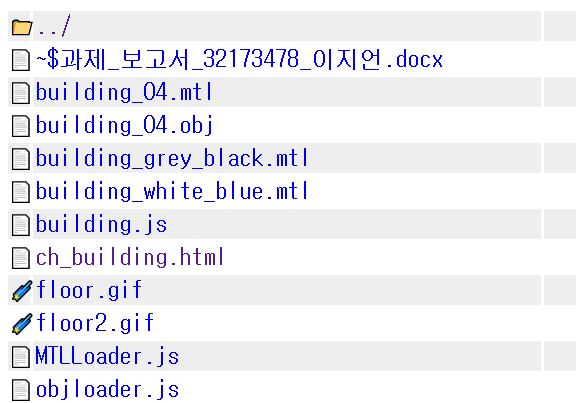
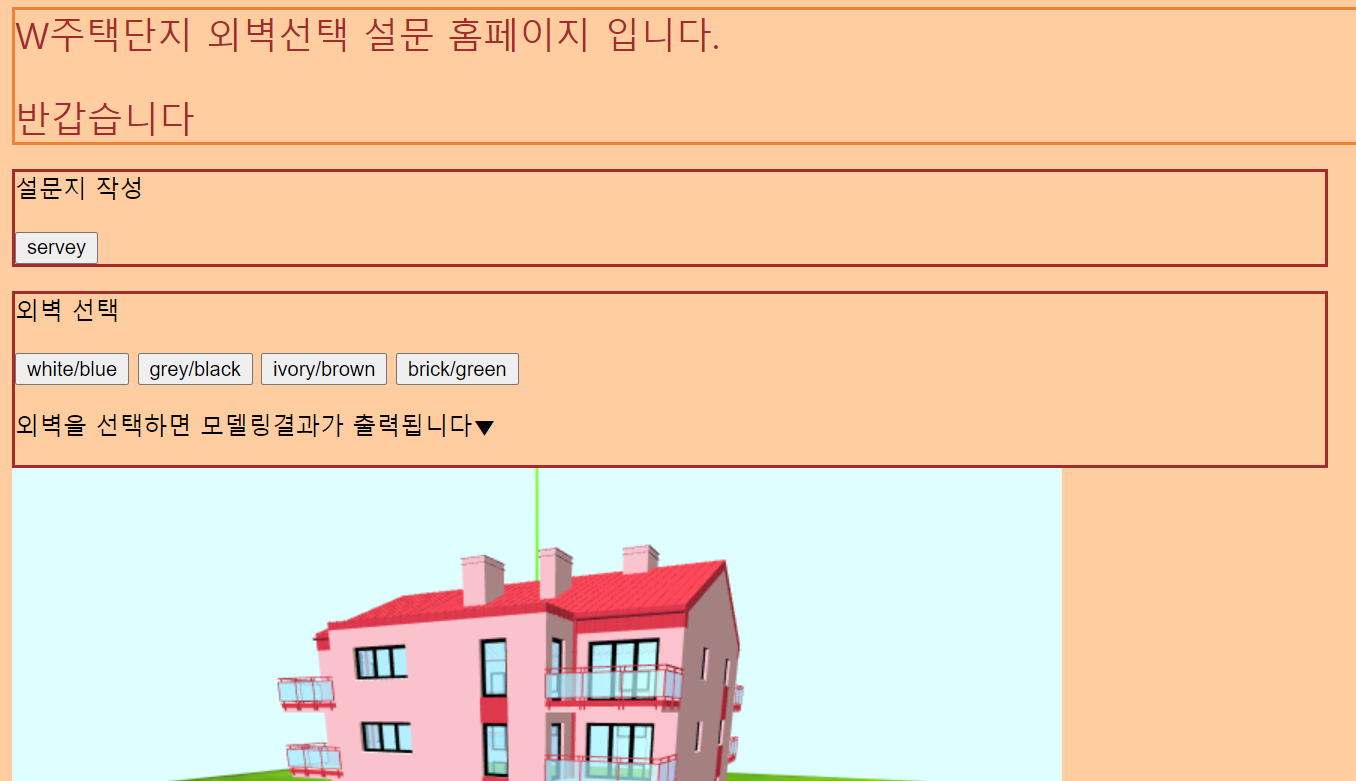
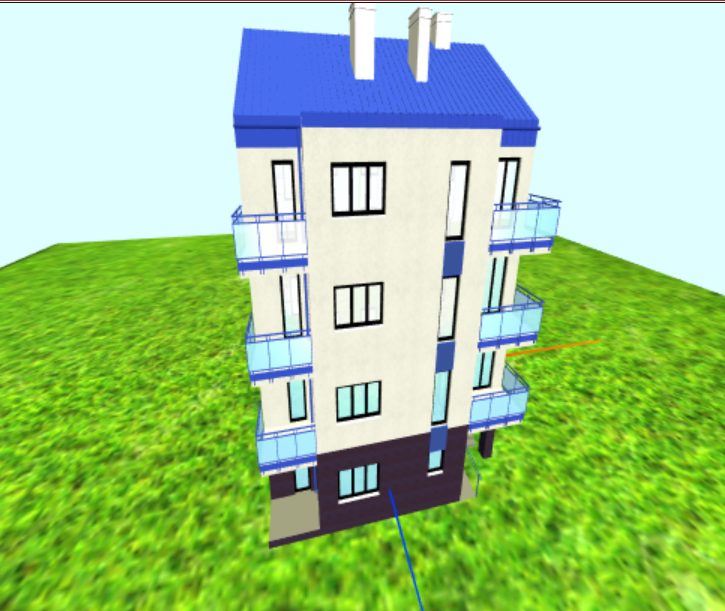
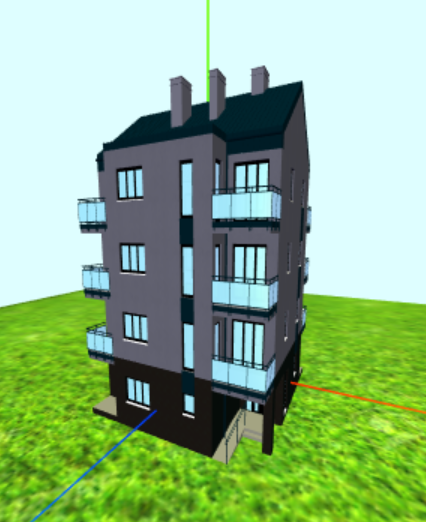
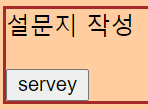
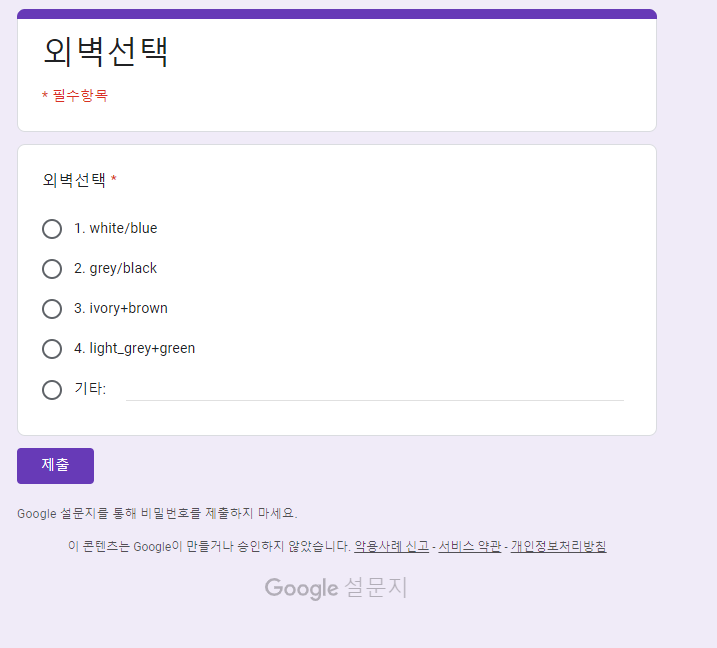
>DirectionalLight: 특정 방향으로 방출되는 빛

>AmbientLight: 조명을 추가하기 전에 장면에 이미 존재하는 조명이다. 전체적으로 장면의 모든 오브젝트를 동일하게 비춥니다.

>SpotLight: 하나의 점에서 한 방향으로 원뿔을 따라 더 큰 크기로 커지는 빛

* **Makefloor()**: 바닥을 만드는 함수이다. 바닥에 잔디 그림으로 만들고 싶어서 texture를 지정하고 해당 파일을 TextureLoader로 읽어온다. Geometry는 Three.js에서 제공하는 평평한 판인 planeGeometry를 사용하고, 크기는 50X50으로 지정한다. 만든 geometry에 적용할 material변수를 만들고, MeshPhongMaterial로 texture와 ambient를 만들어서 반사효과를 준다. Mesh함수를 통해 floor변수에 geometry와 material을 저장하고, 바닥으로 만들기 위해 x축 방향으로 회전시켜야 한다. 그림자를 받을 수 있게 만들고, scene에 만든 바닥을 추가한다.
* **loadMTLLoader(url):** 사용할 mtl파일의 위치를 받으면, mtlLoader를 불러서 변수에 넣어 편리하게 사용한다. mtlLoader를 통해 해당 파일을 로드하고, 3D모델의 vertex정보가 있는 obj파일을 불러오도록 loadOBJLoader(material)을 실행합니다. 이 때, material은 방금 불러온 mtl파일의 정보이다.
* **-**> 로드가 되는 과정을 확인하기 위해 function(xhr)를 만들어서 load되는 정도를 console로 출력하고, 오류가 생기면 알림을 전달합니다.
* **loadOBJLoader(materials):** 편리하게 사용하도록 loader에 OBJLoader객체를 만들어 놓고, 인자로 받은 materials를 setMaterials함수로 material파일을 설정합니다. // obj파일의 위치를 갖고 mtl과 동일하게 load합니다. 건물 객체의 위치를 0,0,0으로 지정하여 OrbitControl의 중앙에 위치시킵니다. Load한 건물 객체를 scene에 추가합니다.
* **initThree():** 브라우저가 WebGL을 지원하는지 체크하고, webgl로 보낼 render의 크기를 700X500으로 설정합니다. (보기 편하도록) 건물에 중요한 요소가 shadow이기 때문에, shadow를 가능하도록 shadowMap을 만들고, renderer가 그릴 곳을 자식으로 추가한다.// x,y,z축을 보여주기 위해 AxisHelper를 사용한다. // camer를 사용자 입장에서 PerspectiveCamera로 설정하고, 그 위치를 설정합니다.// OrbitControls를 사용하여 사용자시선인 카메라가 건물 객체 주위를 공전하도록 합니다. 이는 카메라가 변경될 때 마다 render해야 하므로, addEventListener로 감시합니다. 그리고 나서 공전속도나 zoom속도등 OrbitControl에 필요한 정보들을 저장합니다.
* **Animate():** 반복적인 animate를 위해 requestAnimationFrame에 animate를 한번 더 지정하고, 앞의 함수들에서 지정한 scene,camera를 같이 render합니다. 그리고 OrbitControl로 인해 바뀐 control정보들을 update합니다.
* **버튼 사용:** 해당 버튼에 대한 onclick정보가 들어오면, 그에 맞는 텍스쳐로 변경해 주기 위해 loadMTLLoader를 실행합니다. 각 파일들에 대한 정보는 밑에 기재하겠습니다.
* **설문지 페이지 이동: “**send” 버튼에 대한 요청이 들어오면, google form의 링크로 화면이 변경됩니다.
* 초기화면은 15줄에서 불렀던 mtl파일로 인해, 예시 건물이 띄워집니다.
* Mtl 파일 정보(예시로 building\_grey\_black.mtl을 참고하겠습니다.)
* -> 지붕+난간 material
* 지붕 색은 Ns값을 10을 주어, metal성질에 맞게, 살짝 반사되도록 만들었다. 색은 Kd를 보면, 거의 검정색에 가깝게 나타난다.
* -> 벽면 하단부: 보통 어두운 색
* 벽면 하단부로, 주로 어두운 색을 취하고, 해당 벽면에 대한 텍스처를 매핑한다.
* -> 유리창
* 유리창 특성상 빛에 반사를 많이 하기 때문에, Ns값을 100으로 주었으며, 유리창은 투명하기 때문에, Tr을 0.3으로 살짝 불투명 하게 했다. (완전 투명하게 하면 현실성이 떨어짐), 그리고 색상은 하늘색에 가깝게 하여 유리창을 만들었다.
* -> 벽면 상단
* 벽면 상단의 gray색을 만들고, 울퉁불퉁한 벽면을 사용하기 위해 texture이미지를 입혔다.

1. 결과

* servez라는 cmd명령 툴을 가지고 서버를 만들어 실행했다.
* 
* 컴퓨터의 파일을 서버로 올려주었다. 해당 서버를 웹에서 실행하면,
* 
* 여기서 .html을 들어가면
* 
* 홈페이지가 나타나고, 초기에 렌더링한 모델이 나타난다.
* -> white/blue 
* ->grey/black 
* ->ivory/brown 
* ->brick/green 
* 해당 버튼을 누르면 새롭게 렌더링이 된다.
* 자신이 원하는 외벽이 생겼을 경우, 설문지 작성버튼을 누르면 된다.
*  

4. 소감

-처음에 vertex를 지정해서 하다가 도저히 만족스러운 결과가 나오지 않아서, obj파일을 사용하게 되었습니다. 만약 blender을 사용하여 gltf파일을 사용했다면, 더욱 활용성 있는 프로젝트가 되었을 것 같습니다. 매우 아쉽지만, obj파일을 통해 그동안 배웠던 vertex에 대한 정보를 제대로 알게 되었으며, mtl파일을 분석하면서 texture, fragment에 대한 기본 지식을 다시 복습할 수 있어서 오히려 이론을 복습하기에 좋은 기회라고 생각이 듭니다. 그리고 mtl파일을 직접 변형시켜서 모델에 매핑시키는 과정에서 물체가 어떤 방식으로 빛을 반사하고, 속성을 다양하게 주면서 texture, material에 대한 지식을 습득할 수 있었습니다. 그리 처음에 이해가 잘 되지 않았던 이론적인 부분을 실습하면서 깨우친 것 같습니다.

사실 처음에 건물의 부분적 요소가 너무 많아서 프로젝트 진행에 어려움을 겪었습니다. 하지만 이를 찾아가는 과정에서 객체에 대한 정보들을 제대로 알게 된 것 같습니다.

그동안 좋은 가르침 주셔서 감사합니다.