

Airplane SideWing Texture Report

2021204045 이성민



전경 사진

해당 보고서는 sideWing mesh에 texture를 적용하고, 이미지 비율을 유지하면서 날개 윗면과 아랫면에 Texture를 mapping하는 방법에 대해 설명한다. 또한 날개의 옆면은 임의의 색상이 나오도록 처리하는 방법과 minFilter 설정에 따른 효과를 고찰한다.

SideWing mesh에 Texture 적용

-Texture loading

Texture는 Three.js의 'TextureKoader'를 사용하여 로드된다. Texture가 로드된 후 콜백 함수가 실행된다.

```
const textureLoader = new THREE.TextureLoader();
const texture = textureLoader.load('./Cat.jpeg', () => {
    // Texture 로드 후 실행되는 콜백 함수
});
```

-이미지 비율 계산

이미지의 가로세로 비율을 유지하기 위해 이미지의 가로와 세로 크기를 이용하여 비율을 계산한다.

```
let imageAspectRatio = 761 / 760; // 가로 / 세로
```

-Texture 반복 값 설정

Texture가 윗면과 아랫면에 올바르게 mapping 되도록, 반복 값을 설정한다.

```
let repeatX = 2; //Texture를 x 방향으로 2번 반복
let repeatY = 1 * imageAspectRatio; //Texture를 y 방향으로 1번 반복 (비율 유지)
texture.repeat.set(repeatX, repeatY);
```

-Texture UV 초기화 및 설정

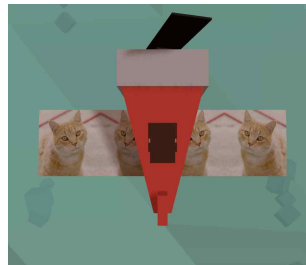
날개의 윗면과 아랫면에 Texture를 mapping하기 위해 UV 좌표를 조정한다.

```
let uv = geomSideWing.attributes.uv.array;
```

```
// Texture 회전을 위한 UV 좌표 조정
```

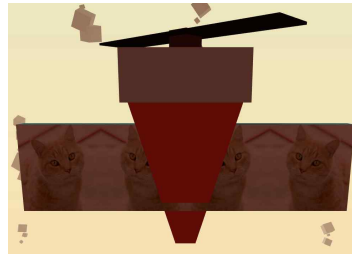
```
// 윗면
```

```
uv[16] = 0; uv[17] = 0;
uv[18] = 0; uv[19] = repeatY;
uv[20] = repeatX; uv[21] = 0;
uv[22] = repeatX; uv[23] = repeatY;
```



```
// 아랫면
```

```
uv[30] = 0; uv[31] = repeatY;
uv[28] = 0; uv[29] = 0;
uv[26] = repeatX; uv[27] = repeatY;
uv[24] = repeatX; uv[25] = 0;
```



```
geomSideWing.attributes.uv.needsUpdate = true;
```

-Texture 설정

'THREE.MirroredRepeatWrapping'를 사용하여 geomSideWing의 정가운데를 기준으로 좌우대칭을 형성(Mirror)한다.

```
texture.wrapS = THREE.MirroredRepeatWrapping;
texture.wrapT = THREE.MirroredRepeatWrapping;
```

SideWing mesh의 옆면 색상 처리

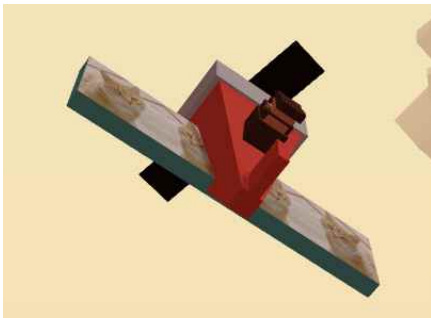
sideWing의 각 면에 대한 재질을 생성한다. 윗면과 아랫면에는 Texture를 적용하고, sideWing의 오른쪽, 왼쪽, 앞면, 뒷면에는 Texture 대신 임의의 색상을 적용한다. 이 색상은 MeshPhongMaterial의 color 속성을 이용하여 지정한다. (이 예제에선 blue로 설정했다.)

```

var matTexture = new THREE.MeshPhongMaterial({ map: texture, flatShading: true });

var materials = [
    new THREE.MeshPhongMaterial({ color: Colors.blue, flatShading: true }), // 오른쪽 면
    new THREE.MeshPhongMaterial({ color: Colors.blue, flatShading: true }), // 왼쪽 면
    matTexture, // 윗면
    matTexture, // 아랫면
    new THREE.MeshPhongMaterial({ color: Colors.blue, flatShading: true }), // 앞면
    new THREE.MeshPhongMaterial({ color: Colors.blue, flatShading: true }) // 뒷면
];

```



minFilter 설정에 따른 효과 고찰

minFilter는 texture가 축소될 때의 필터링 방법을 결정한다. 이 예제에서는 'LinearMipmapLinearFilter'를 사용하여 Texture 축소 시 높은 품질을 유지하도록 한다. 이 필터는 부드러운 축소 결과를 제공하며, 특히 작은 크기로 축소될 때 유리하다.

-안티소트로픽 필터링

안티소트로픽 필터링(Anisotropic Filtering, AF)은 Texture Mapping에서 시각적 품질을 향상시키기 위해 사용되는 기술이다. 특히, Texture가 사선 또는 멀리서 볼 때 흐릿해지는 문제를 해결하는 데 유용하다. 일반적인 Texture 필터링 방법과 비교하여 더욱 선명하고 디테일한 이미지를 제공한다.

-안티소트로픽 필터링의 원리

Texture가 뷰어의 시야에서 사선으로 배치되면 Texture의 축소 및 왜곡이 발생한다. 기본적인 이방성 필터링은 Texture의 각 픽셀이 다양한 각도에서 어떻게 보이는지 계산하여 이러한 왜곡을 최소화한다. 이를 통해 texture가 다양한 각도에서도 선명하게 보이게 된다. Texture의 anisotropy 속성을 설정하여 사용할 최대 안티소트로픽 필터링 수준을 지정할 수 있다.

```

texture.minFilter = THREE.LinearMipmapLinearFilter; // 축소 필터링
texture.anisotropy = 64; // 안티소트로픽 레벨 설정

```

-안티소트로픽 필터링의 효과

선명한 Texture: Texture가 멀리서나 사선으로 볼 때도 선명하게 보임.

시각적 품질 향상: 특히 대형 Texture나 디테일이 중요한 장면에서 품질이 크게 향상.

Texture 왜곡 감소: Texture의 왜곡을 줄여 더 자연스러운 시각적 경험을 제공.



안티소트로픽 필터링 없음



안티소트로픽 필터링 적용

-안티소트로픽 필터링 없음

안티소트로픽 필터링을 사용하지 않으면 Texture가 뷰어와의 각도에 따라 흐릿해지거나 왜곡될 수 있다. 특히 멀리 있는 Texture나 사선으로 배치된 Texture에서 이러한 현상이 두드러진다.

-안티소트로픽 필터링 적용

안티소트로픽 필터링을 적용하면 이러한 문제를 해결하여 Texture가 항상 선명하고 디테일하게 보인다. 필터링 레벨을 높이면 품질이 더욱 향상되지만, 성능에 대한 영향을 고려해야 한다.

요약

이 Report는 SideWing mesh에 texture를 적용하는 방법과 각 면에 임의의 색상을 지정하는 방법에 대해 설명하였다. 또한 minFilter 설정과 안티소트로픽 필터링을 통해 Texture 품질을 유지하는 방법을 고찰하였다. 이러한 접근법을 통해 이미지 비율을 유지하면서 Texture를 효과적으로 mapping 할 수 있었다.