

툰슈에이딩 구현

유영천

<https://megayuchi.com>

tw@dgtman

툰쉐이딩

- 디테일이 높은 것이 아름답다는 뜻은 아니다.
- 멀쩡하게 잘 나온 면상 사진을 뽀뽀 문질러서 **디테일을 떨어뜨리고**는 이쁘다고 좋아하는 행태를 떠올려보자
- 만화풍 = 일부러 디테일을 떨어뜨린다 -> 이뻐보인다.
- 따라서 실사풍보다 만화풍을 좋아하는 유저들도 많다.
- 특히 게임에선 그래픽 리소스가 그다지 정교하지 못할 때 훌륭하게 눈속임이 된다!



f:0 obj:7 hfo:0 spr:1 font:4 P:4191 V:2979 W:0 FL:31.28MB-Fail:0
Tex:16 VB:28 IB:32 CB:29 VL:5 A.Map:0 font:0 font-rect-cache:5/5(100.0%)

Vertex간 normal 보간
N dot L Lighting in Pixel Shader

Edit

Character Hierarchy

MK_000_Head.MOD

☒ Per-Vertex Lighting
☐ Per-Pixel Lighting
☐ Toon Shading
☐ Flat Shading
☐ Block Rendering

Detach

☐ Use OutLine
☐ Use PostEffect
☒ Show Grid
☐ Tessellation
☒ Shadow Bias

Bias 0
Bias 1
Bias 2
Bias 3 Apply

☐ Collision Mesh
W D H



Megayuchi
Create Mod
miku_mode
MK_000_Fa

N_DATA_WORK\app\data\character\W



f:1 obj:7 hfo:0 spr:1 font:4 P:4191 V:2979 W:0 FL:31.28MB-Fail:0
Tex:16 VB:28 IB:32 CB:29 VL:5 A.Map:0 font:0 font-rect-cache:8/9(88.9%)

Vertex간 normal 보간
N dot L Lighting ->
Toon Table Texturing in Pixel Shader

Edit

Character Hierarchy

MK_000_Head.MOD

☐ Per-Vertex Lighting
☐ Per-Pixel Lighting
☒ Toon Shading
☐ Flat Shading
☐ Block Rendering

Detach

☐ Use OutLine
☐ Use PostEffect
☒ Show Grid
☐ Tessellation
☒ Shadow Bias

Bias 0
Bias 1
Bias 2
Bias 3 Apply

☐ Collision Mesh
W D H



Megayuchi
Create Mod
miku_mode
MK_000_Fa

N_DATA_WORK\app\data\character\W

OneDrive



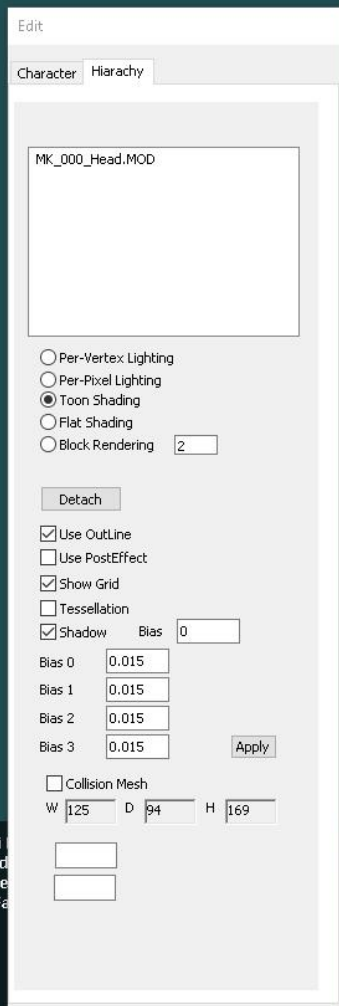
Screenshot saved

The screenshot was added to your OneDrive.



f:1 obj:7 hfo:0 spr:1 font:4 P:4191 V:2979 W:0 FL:31.28MB-Fail:0
Tex:16 VB:28 IB:32 CB:29 VL:5 A.Map:0 font:0 font-rect-cache:8/9(88.9%)

Vertex간 normal 보간
N dot L Lighting ->
Toon Table Texturing in Pixel Shader
+ Outlines with post-processing



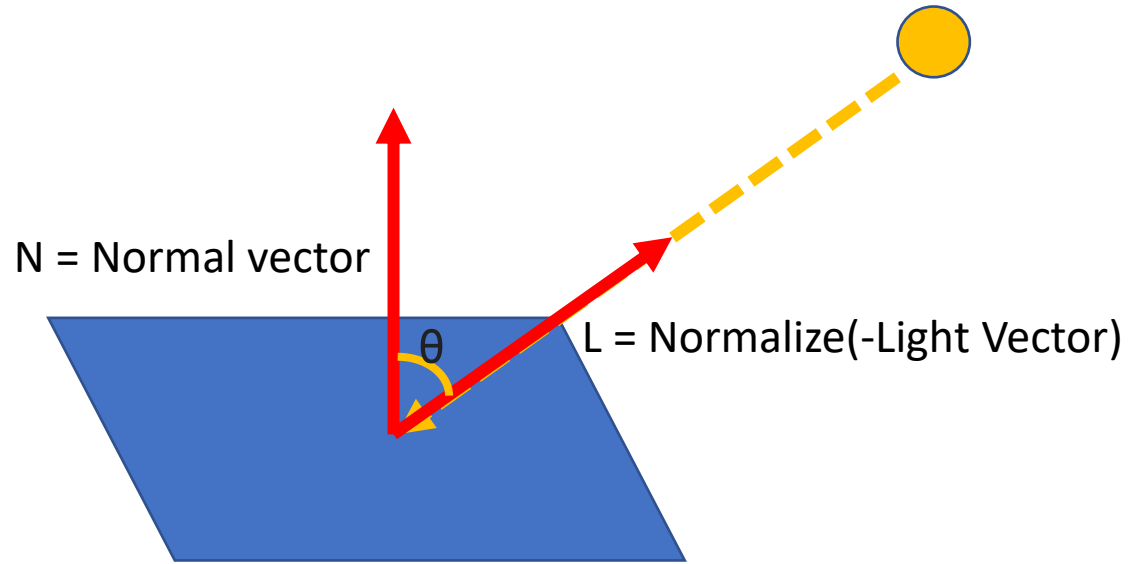
Megayuchi
Create Mod
miku_mode
MK_000_Fa

N_DATA_WORK\Wapp\Data\Character\W

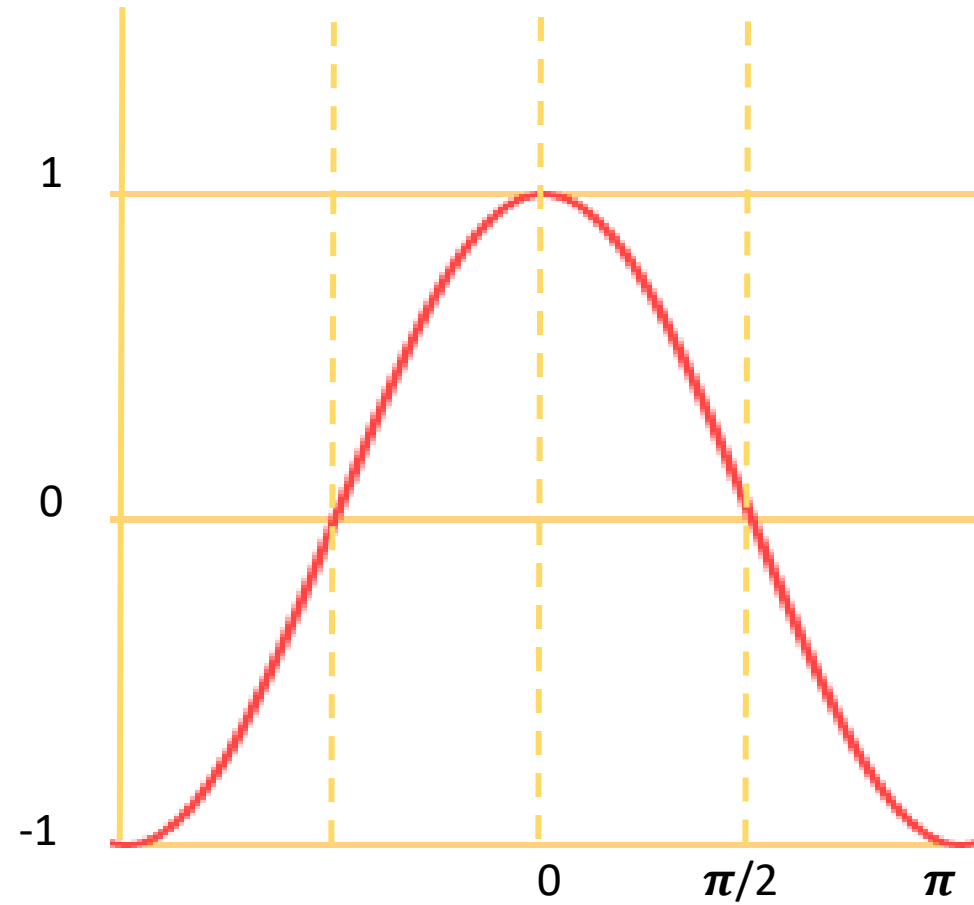
툰쉐이딩 기법

1. Toon Table Texture를 참조해서 색상 단순화해서 렌더링
2. 외곽선 추가
 - normal / depth를 함께 렌더링(Gbuffer를 사용한다면 어차피 기본 작업).
 - normal / depth 성분을 이용해서 외곽선 텍스처 생성.
3. GBuffer를 최종 렌더링 할 때 외곽선 텍스처 적용.
4. Gbuffer를 사용하지 않을 경우 마지막 패스에서 외곽선 텍스처를 덮어서 그림.

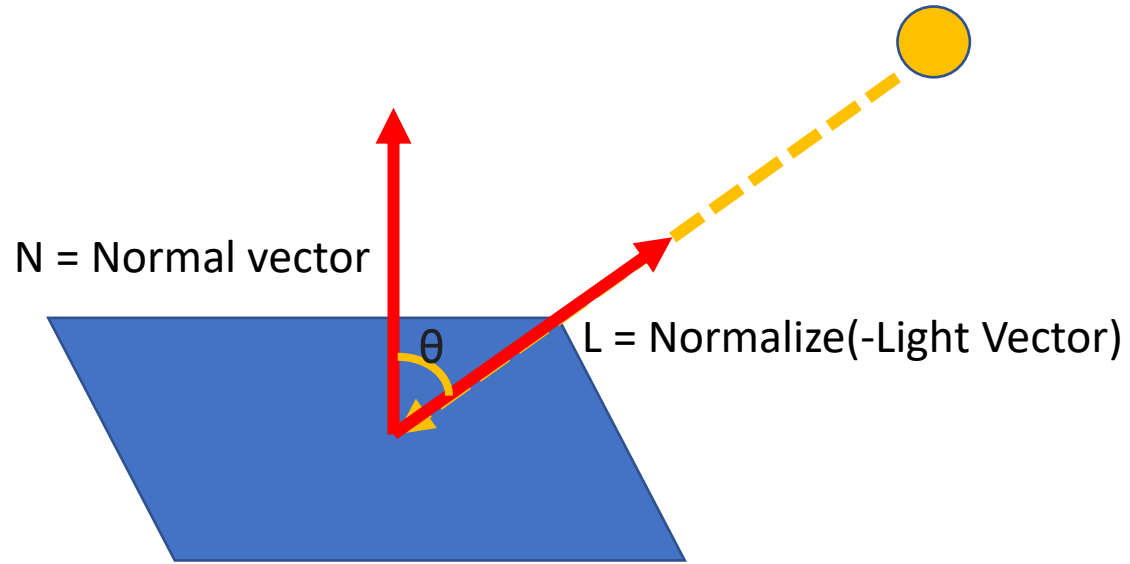
기본적인 라이팅



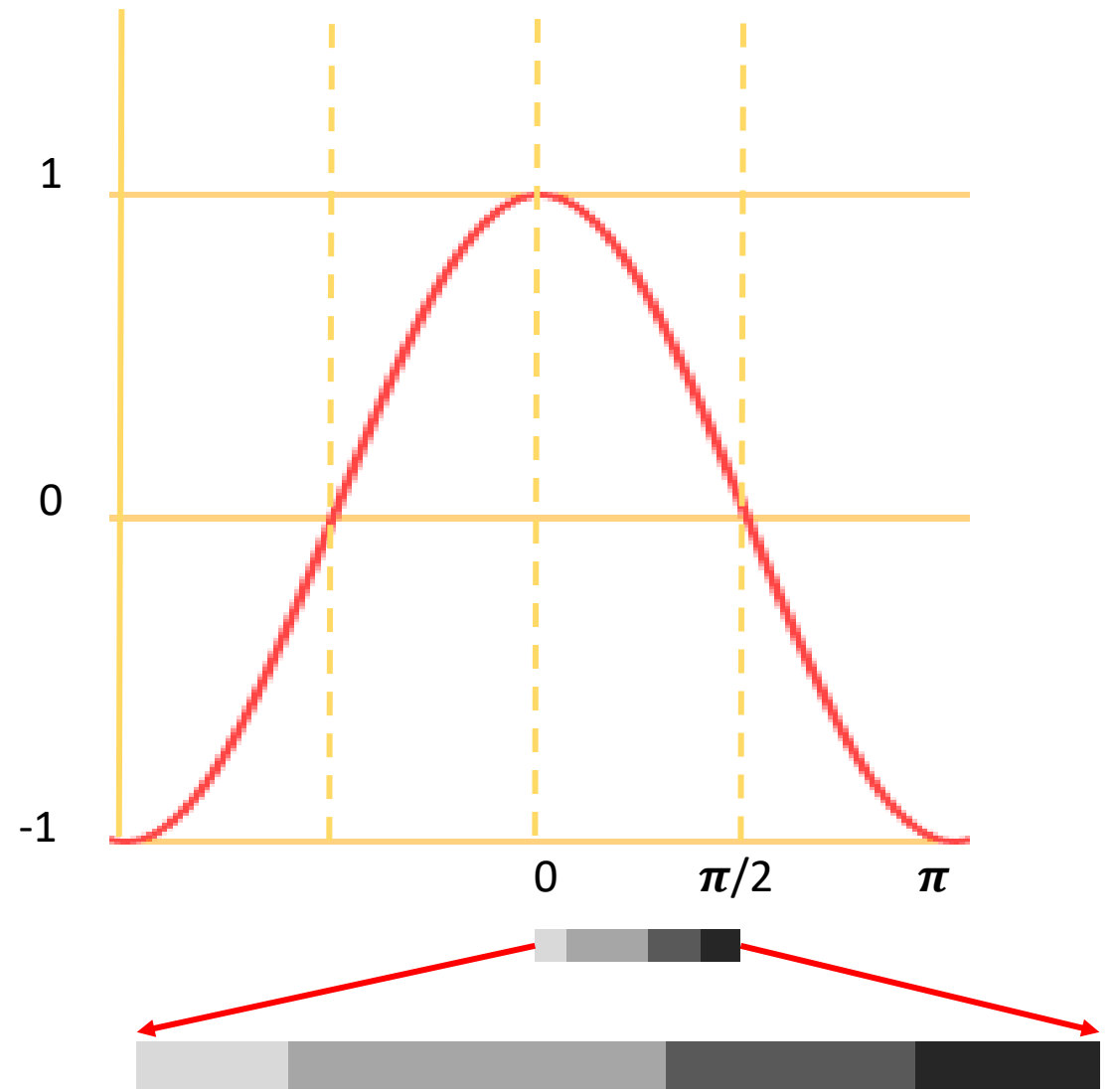
$N \cdot L = \cos \theta \rightarrow$ 밝기로 사용 (0 - 1)



기본적인 라이팅 -> 색상 단순화



$N \cdot L = \cos \theta \rightarrow$
Toon Texture Table의 텍스처 좌표로 사용 (0 - 1)



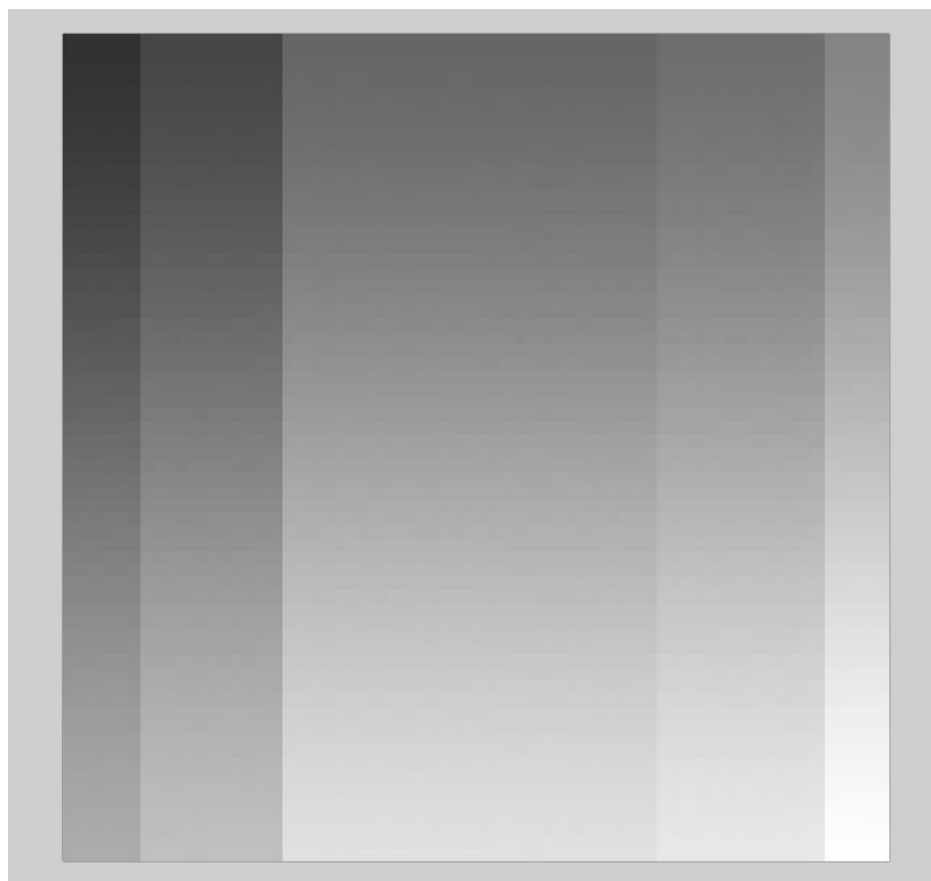
1D Toon Table Texture

Toon Table Texture

1D



2D

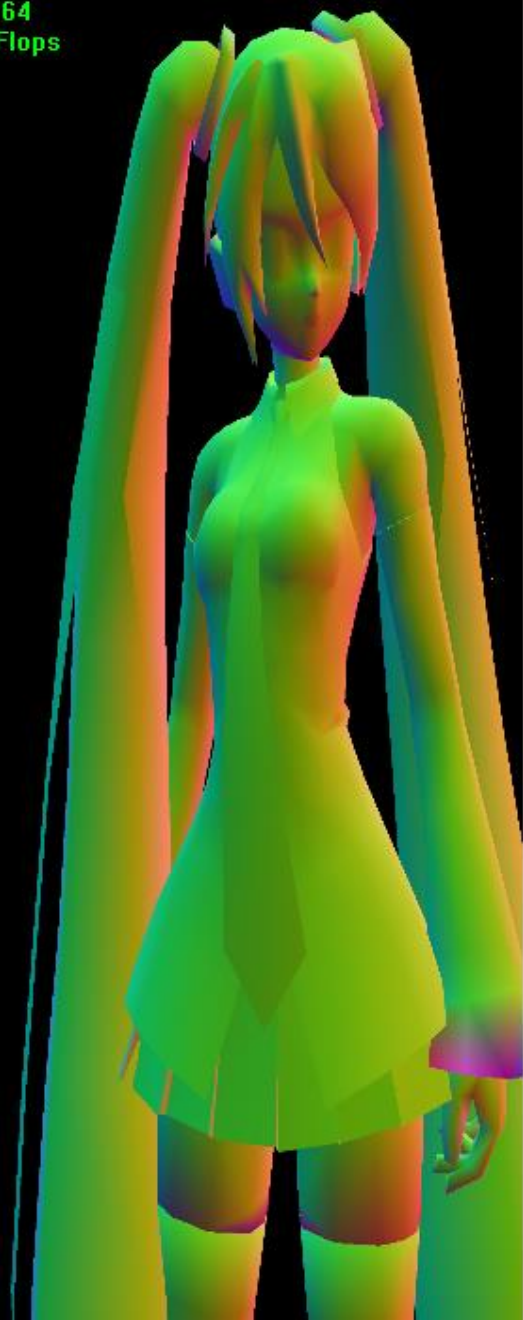


외곽선 텍스처 생성

- 임의의 텍스처에 normal을 그린다.
- Laplacian 필터를 적용해서 엣지 성분을 추출.
- 엣지 성분의 r,g,b를 합산하여(BW) inverse
- 인접한 픽셀들의 컬러 값이 급격하게 변할수록 진한 외곽선이 나온다.

Device Name:NVIDIA GeForce GTX 1660 Ti
GPU ClockRate:1.77GHz
MultiProcessors:24
SM Per MultiProcesso:64
Max GFlops:5437.44GFlops

f:0 obj:7 hfo:0 sp
Tex:19 VB:31 IB:3



Device Name:NVIDIA GeForce GTX 1660 Ti
GPU ClockRate:1.77GHz
MultiProcessors:24
SM Per MultiProcesso:64
Max GFlops:5437.44GFlops

f:0 obj:7 hfo:0 sp
Tex:19 VB:31 IB:3

[CUDA, 0 Threads] 3.915300ms Elapsed.



normal -> Outline

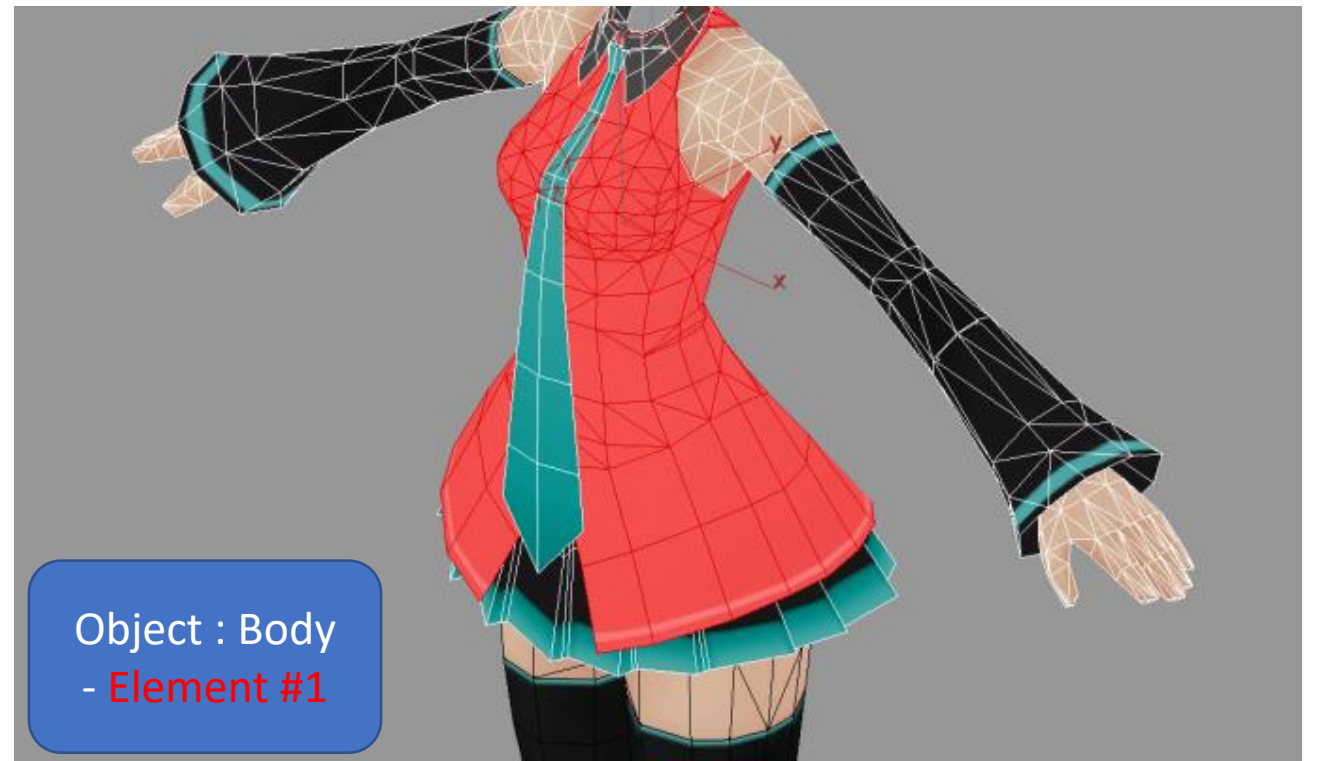
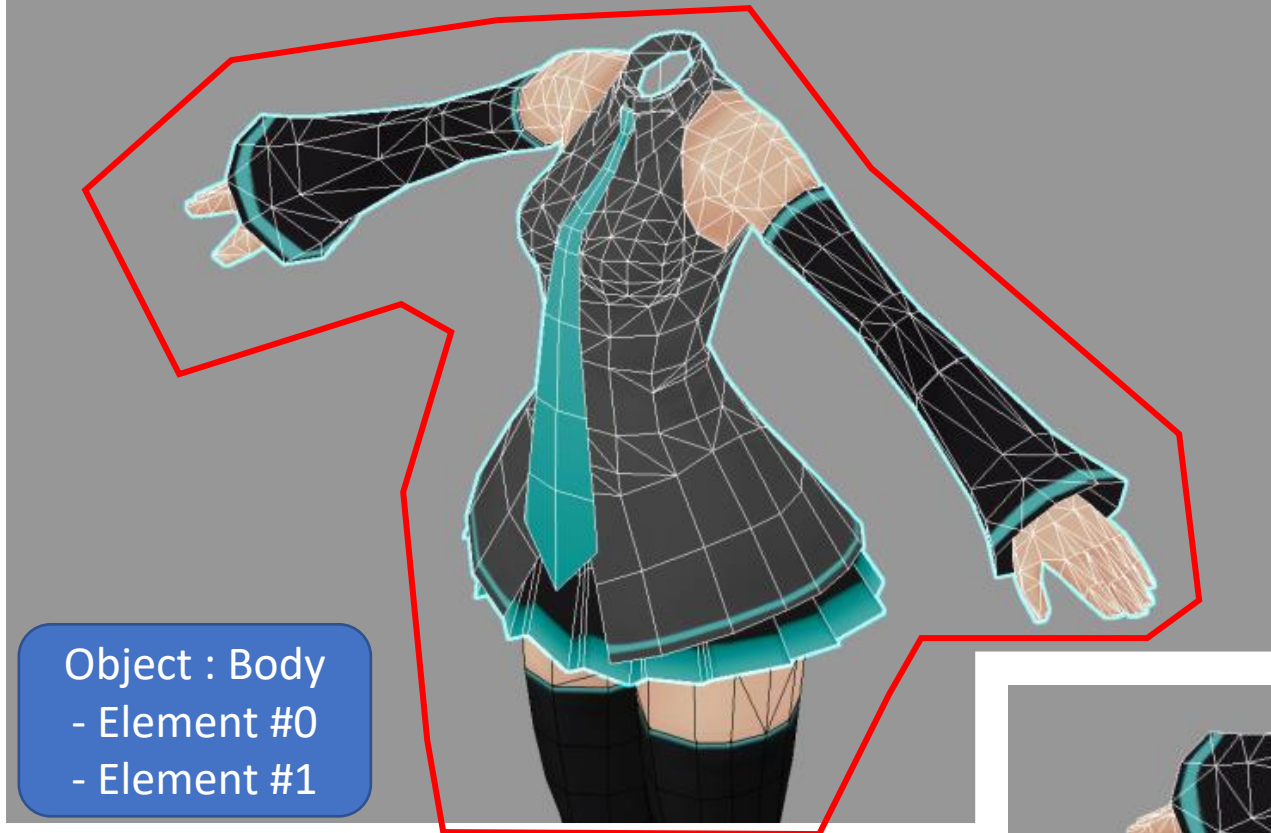
문제점 - 명백히 분리된 매시 사이에서도 외곽선이 생기지 않는다.

- normal 방향이 비슷하고 거리값(z값)이 가까운 매시들 사이에서(명백히 분리되어 있더라도) 외곽선이 생기지 않는다.
- RGBA a성분에 거리값(z값)을 넣어도 정밀도가 8bits에 불과해서 거리가 가까우면 의미있는 색상차를 만들어낼 수 없다.



normal 성분 필터링의 보완

- 분리된 매시(3dsmax에서의 element)에 각각 ID를 부여한다.
- Export할 때 매시데이터에 element ID를 포함한다.
- GBuffer를 만들 때 normal외에 element ID도 렌더링 한다.
- 외곽선 추출 시 ID값의 차이를 반영한다.



Element in 3ds max

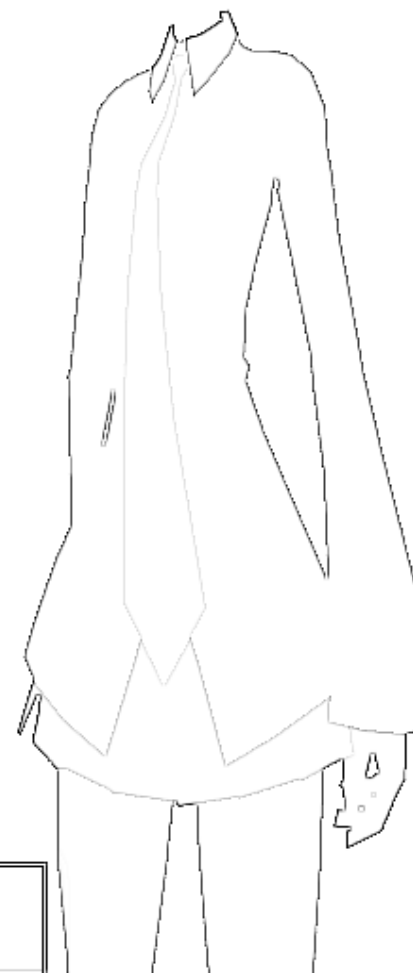
Device Name:NVIDIA GeForce GTX 1660 Ti
GPU ClockRate:1.77GHz
MultiProcessors:24
SM Per MultiProcesso:64
Max GFlops:5437.44GFlops

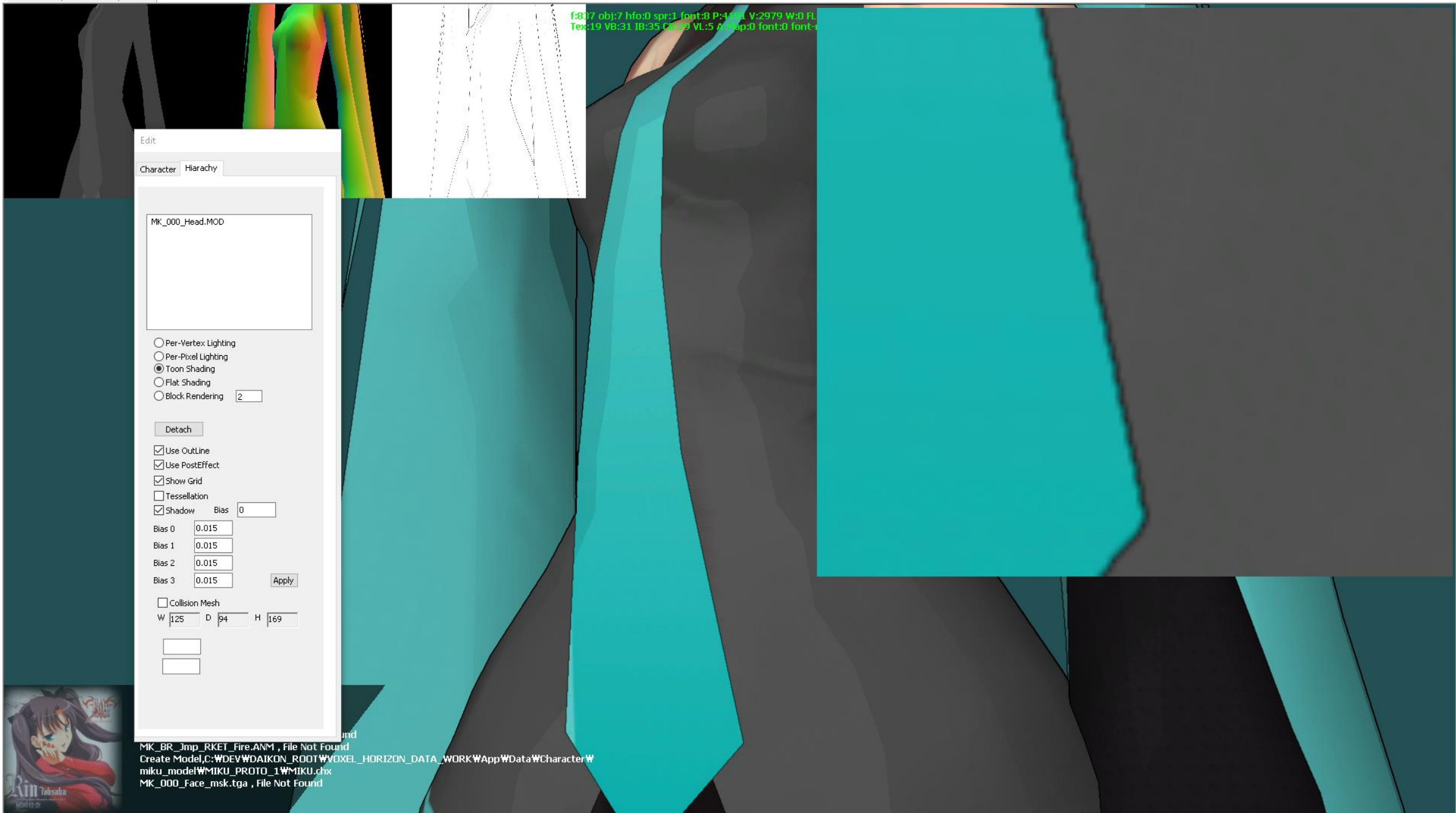


Element ID -> Outline

Device Name:NVIDIA GeForce GTX 1660 Ti
GPU ClockRate:1.77GHz
MultiProcessors:24
SM Per MultiProcesso:64
Max GFlops:5437.44GFlops

[CUDA , 0 Threads] 1.517300ms Elapsed.





Edit

Character Hierarchy

MK_000_Head.MOD

☐ Per-Vertex Lighting
☐ Per-Pixel Lighting
☒ Toon Shading
☐ Flat Shading
☐ Block Rendering

Detach

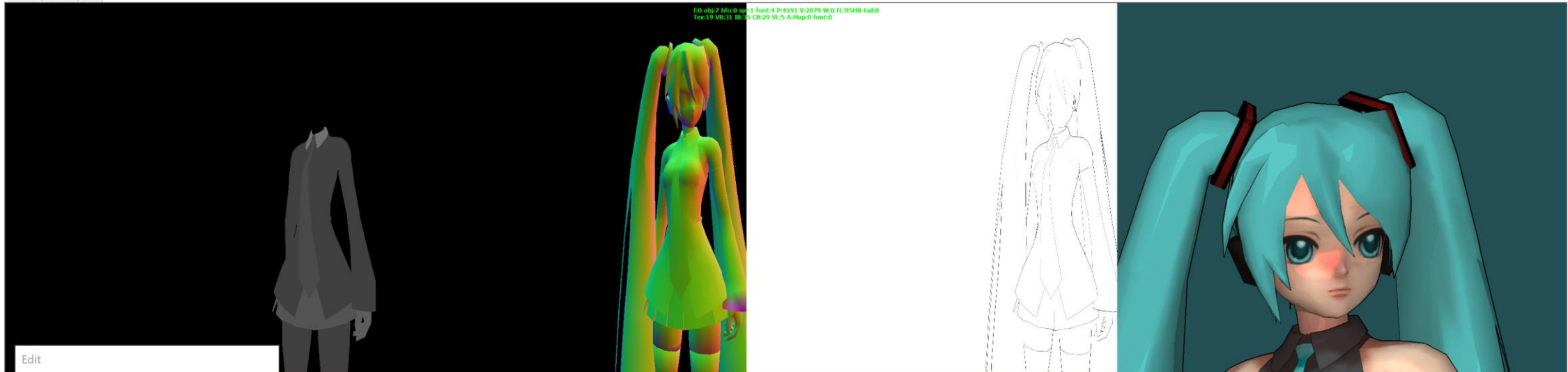
☒ Use OutLine
☒ Use PostEffect
☒ Show Grid
☐ Tessellation
☒ Shadow Bias

Bias 0
Bias 1
Bias 2
Bias 3 Apply

☐ Collision Mesh
W D H



MK_BR_Jmp_RKET_Fire.ANM , File Not Found
Create Model,C:\WDEV\WDAIKON_ROOT\WVOXEL_HORIZON_DATA\WORK\WApp\WData\WCharacter\W
miku_model\WMIKU_PROTO_1\WMIKU.chx
MK_000_Face_msk.tga , File Not Found



Edit

Character Hierarchy

Model List

Motion List

Material List

Dupl & Edit

Object List

Attach

PLAY

Wire Frame



결론 및 부록

- 툰 셰이딩 자체는 구현하기 쉽다.
- 잘~ 적용하려면 약간의 고민이 필요하다.
 - 오브젝트 별로 외곽선 on/off
 - 외곽선 두께 조절
 - 매시의 모양에 따른 외곽선 진하기 조절
 - Toon texture의 색상 분포 비율
 - Toon texture를 1D로 쓰지 2D로 쓰지...

결론 및 부록

- 외곽선을 그리는 더 쉬운 방법도 있다.
 1. 프레임 버퍼에 렌더링 할 때 약간 scale up(ex:1.1배)해서 렌더링 한다.
 2. 이때 모든 면을 뒤집어서 렌더링한다.(Back face cullin모드를 반대로)
 3. 그 위에 정상적으로 (scale : 1.0배) , (back face culling 모드 원래대로) 렌더링 한다.
 4. 외곽선 진하기와 두께가 일정하다(좋은 의미가 아님).
 5. 코/ 입술 등에서 어색하게 출력될 수 있음.