**오브젝트를 구성하는 모든 Vertex는 고유의 색을 가지고 있는데 이것을 Vertex Color라고 한다**

**3dsMax에서 기본값은 흰색을 가지고 있으며**,**Editable Poly안에서 Vertex Color를 확인해 볼 수 있다**

텍스트, 공, 스크린샷, 구체이(가) 표시된 사진

자동 생성된 설명

**Vertex color는 여러곳에 응용 할 수 있는데 일반적으로 게임에서는 Texture대신 해서 사용 할 수 있다.** **게임 개발에서 Texture대신 Vertex color를 사용하면 최적화에 도움이 많이 된다.** 직사각형, 사각형, 스크린샷, 흑백이(가) 표시된 사진

자동 생성된 설명 직사각형, 사각형, 스크린샷, 흑백이(가) 표시된 사진

자동 생성된 설명

텍스쳐 버텍스 컬러

**AddMod = VertexPaint ()**

**addmodifier $ AddMod**

Modifier에 VertexPaint를 추가

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**$.modifiers[#VertexPaint].lightingModel = 2**

VertexPaint에서 LightModel안에 설정을 Diffuse Only로 변경

텍스트, 스크린샷, 찻주전자, 주전자이(가) 표시된 사진

자동 생성된 설명

**$.modifiers[#VertexPaint].colorBy = 1**

VertexPaint에서 Color assignmen안에 설정을 Color by Vertex로 설정

텍스트, 스크린샷, 찻주전자, 주전자이(가) 표시된 사진

자동 생성된 설명

**$.modifiers[#VertexPaint].useMaps = on**

VertexPaint에서 Rendering Options안에 있는 Mapping체크박스를 활성화

텍스트, 스크린샷, 찻주전자, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**modPanel.setCurrentObject $.modifiers[#VertexPaint]**

Modifier에 있는 VertexPaint를 선택

텍스트, 스크린샷, 찻주전자, 주전자이(가) 표시된 사진

자동 생성된 설명

**$.showVertexColors = on**

3dsMax 화면에서 바로 Vertex Color를 확인

텍스트, 스크린샷, 찻주전자, 디자인이(가) 표시된 사진

자동 생성된 설명

**modPanel.setCurrentObject $.modifiers[#Edit\_Poly]**

Modifier에 Edit Poly가 적용이 되어있으면 제일 위에 있는 Edit Poly 탭을 선택할 수 있는 명령어

**$.modifiers[#Edit\_Poly].SetSelection #Face #{}**

선택한 면이 하나도 없게 만들어 주는 명령어

**$.modifiers[#Edit\_Poly].selectByMaterialID = 숫자(Material ID)**

선택할 영역으로 Material ID를 설정, 입력할 숫자는 본인이 입력하고자 하는 숫자에 -1를 해주어야 정확히 원하는 수가 입력

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**$.modifiers[#Edit\_Poly].ButtonOp #SelectByMaterial**

설정한 Material ID를 가진 모든면을 선택해주는 명령어 (Edit Poly안에 있는 select ID버튼과 동일한 기능)

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**$.modifiers[#Edit\_Poly].DetachToObject “문자열”**

선택한 면을 지정한 문자열 이름으로 Detach 해주는 기능(Edit Poly 안에 있는 Detach기능과 동일한 효과)

**$.modifiers[#Edit\_Poly].Attach $오브젝트 editPolyNode:$**

명령어는 지정한 오브젝트를 Attach 해주는 기능 (Edit Poly안에 있는 Attach기능과 동일한 효과)

**Polyop.getElementsUsingFace $ #(i)**

지정한 면(i)을 가지고 있는 Element의 모든 면의 정보를 알 수 있다

**makeUniqueArray #(1, 4, 56, 7, 4, 9, 12, 4, 1, 10)**

makeUniqueArray명령를 사용하면 Array안에 중복되는 값을 한개만 남겨 두고 모두 삭제

**$.modifiers[1].unwrap6.selectFacesByNode 지정한 면 넘버 $**

지정한 면의 넘버부분을 Unwrap UVW안에서 선택

**$.modifiers[#unwrap\_uvw].scaleSelectedVerticesCenter a b**

선택한 오브젝트의 모디파이어의 unwrap uvw에서 선택한 vertex의 x scale(a), y scale(b)을 변수 지정한 수치 만큼으로 조절

**newMap = Gradient\_Ramp ()**

Material/Map Browser안에 있는 Graadient Ramp를 변수안에 넣어준 것과 동일한 효과

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**newMap.coordinates.W\_Angle = 90**

변수newMap의 coordimates.W\_Angle수치에 90을 넣어줌

텍스트, 스크린샷, 멀티미디어 소프트웨어, 전자제품이(가) 표시된 사진

자동 생성된 설명

**meditMaterials[24] = Standardmaterial()**

Materials[24]를 standard로 설정

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**meditMaterials[24].diffuseMap = newMap**

Materials[24]의 diffuseMap부분에 설정했던 newMap을 적용

텍스트, 스크린샷, 멀티미디어 소프트웨어, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**$.material = meditMaterials[24]**

선택한 오브젝트의 Material안에 만들었던 Materials[24]를 적용

스크린샷, 텍스트, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

**meditMaterials[24].showInViewport = true**

Materials[24]에 showInViewport옵션에 true를 넣어서 활성화

텍스트, 스크린샷, 멀티미디어 소프트웨어, 그래픽 소프트웨어이(가) 표시된 사진

자동 생성된 설명

try(destroydialog HairVertex)catch()

Rollout HairVertex "Hair\_VertexColor"

(

Group "VertexColor\_Bake"

(

pickbutton PM01 "PickMesh" width:170 across:2

Spinner SE01 "Hair\_ID:" type:#integer Offset:[-10,0] Range:[1,100,1]

button T01 "Hair\_Gradient" width:170 across:2

Button BV "Bake\_VertexColor" width:170

progressBar PB01 Color:red

)

fn MoveUvCenter =

(

Cen = $.modifiers[1].getSelCenter()

MovOffset = [0.5,0.5] - Cen

$.modifiers[1].unwrap2.MoveSelected [MovOffset[1],MovOffset[2],0]

)

on T01 pressed do

(

try

(

if $Hair == undefined then

(

if Getnodebyname PM01.text != undefined and PM01.text != "PickMesh" do

(

Parsel = Getnodebyname PM01.text

Select Parsel

max modify mode

addMod = Edit\_Poly ()

addModifier $ addMod

subobjectLevel = 4

$.modifiers[#Edit\_Poly].SetSelection #Face #{}

MatID = (SE01.value-1)

$.modifiers[#Edit\_Poly].selectByMaterialID = MatID

$.modifiers[#Edit\_Poly].ButtonOp #SelectByMaterial

NewName = "Hair"

$.modifiers[#Edit\_Poly].DetachToObject NewName

select (getnodebyname NewName)

)

ElementArray = #()

for i in 1 to $.Faces.count do

(

TEST = Polyop.getElementsUsingFace $ #(i)

TESTS = TEST as String

Append ElementArray TESTS

PB01.value = (100\*i/$.Faces.count)/2

)

ElementArrayB = makeUniqueArray ElementArray

addMod = Unwrap\_UVW ()

addModifier $ addMod

for i in 1 to ElementArrayB.count do

(

Xarray = #()

Yarray = #()

ElementB = execute ElementArrayB[i]

$.modifiers[1].unwrap6.selectFacesByNode ElementB $

MoveUvCenter()

$.modifiers[1].faceToVertSelect()

UVSelVer = $.modifiers[1].getSelectedVertices()

UVSelVerA = UVSelVer as array

for a in UVSelVerA do

(

oldPos = $.modifiers[1].getVertexPosition 0 a

append Xarray oldPos[1]

append Yarray oldPos[2]

)

UVXMax = (amax Xarray)-0.5

UVYMax = (amax Yarray)-0.5

XMul = (0.95/UVXMax)/2

YMul = (0.95/UVYMax)/2

$.modifiers[#unwrap\_uvw].scaleSelectedVerticesCenter XMul 1

$.modifiers[#unwrap\_uvw].scaleSelectedVerticesCenter YMul 2

MoveUvCenter()

PB01.value +=((100\*i/ElementArrayB.count)/2)

)

newMap = Gradient\_Ramp ()

newMap.coordinates.W\_Angle = 90

meditMaterials[24] = Standardmaterial()

meditMaterials[24].diffuseMap = newMap

$.material = meditMaterials[24]

meditMaterials[24].showInViewport = true

max create mode

clearSelection()

select $hair

PB01.value = 0

)

else

(

messagebox "There is hairMesh, check it"

)

)

catch()

)

on PM01 picked obj do

(

try

(

if obj != undefined do

(

PM01.text = obj.name

)

)

catch()

)

on BV pressed do

(

try

(

select $hair

selhairMesh = selection as array

if selhairMesh.Count == 1 do

(

AddMod = VertexPaint ()

addmodifier $ AddMod

$.modifiers[#VertexPaint].lightingModel = 2

$.modifiers[#VertexPaint].colorBy = 1

$.modifiers[#VertexPaint].useMaps = on

max modify mode

modPanel.setCurrentObject $.modifiers[#VertexPaint]

maxHWND = windows.getMAXHWND()

UINumArray= (UIAccessor.GetChildWindows maxHWND)

cmdHWND = for d in 1 to UINumArray.count where UIAccessor.GetWindowText UINumArray[d] == "Assign" do exit with UINumArray[d]

UIAccessor.PressButton cmdHWND

$.showVertexColors = on

for i in ($.modifiers.count) to 1 by -1 do

(

if classof ($.modifiers[i])== Unwrap\_UVW do

(

deleteModifier $ i

)

)

converttopoly $

PB01.value =50

)

hairMesh = $

Parsel = Getnodebyname PM01.text

backMat = Parsel.material

select Parsel

PB01.value =70

$.modifiers[#Edit\_Poly].Attach hairMesh editPolyNode:$

$.showVertexColors = on

$.material = backMat

PB01.value =90

converttopoly $

PB01.value =100

PB01.value = 0

)

catch()

)

)

Createdialog HairVertex width:400