MASTER ELECTRONIC DESIGN

Homework 4( )

Miguel Tlapa Juárez

19/05/2014

This document describes the system architecture and design about the body controller module, it's have block diagram and flowchart to describe software and hardware architecture.

Revision History

Date	Revision Number	Author/Editor	Modifications
June2014	0.1	Miguel Tlapa	Created file

Disclaimers

EXERCISES

1. **Send a random color for each vertex and show the result when no rasterizer is used.**

Modify the file ModelApp.cpp*

a) Add element XMFLOAT3 Normal to Struct Vertex

```
struct VERTEX
{
    XMFLOAT3 Pos;
    XMFLOAT3 Normal;
    XMFLOAT3 Color;

};
```

b) Add element COLOR to D3D11_INPUT_ELEMENT_DESC

```
D3D11_INPUT_ELEMENT_DESC ied[] =
{
    { "POSITION", 0, DXGI_FORMAT_R32G32B32_FLOAT, 0, D3D11_APPEND_ALIGNED_ELEMENT, D3D11_INPUT_PER_VERTEX_DATA, 0 },
    { "NORMAL", 0, DXGI_FORMAT_R32G32B32_FLOAT, 0, D3D11_APPEND_ALIGNED_ELEMENT, D3D11_INPUT_PER_VERTEX_DATA, 0 },
    { "COLOR", 0, DXGI_FORMAT_R32G32B32_FLOAT, 0, D3D11_APPEND_ALIGNED_ELEMENT, D3D11_INPUT_PER_VERTEX_DATA, 0 },
};
```

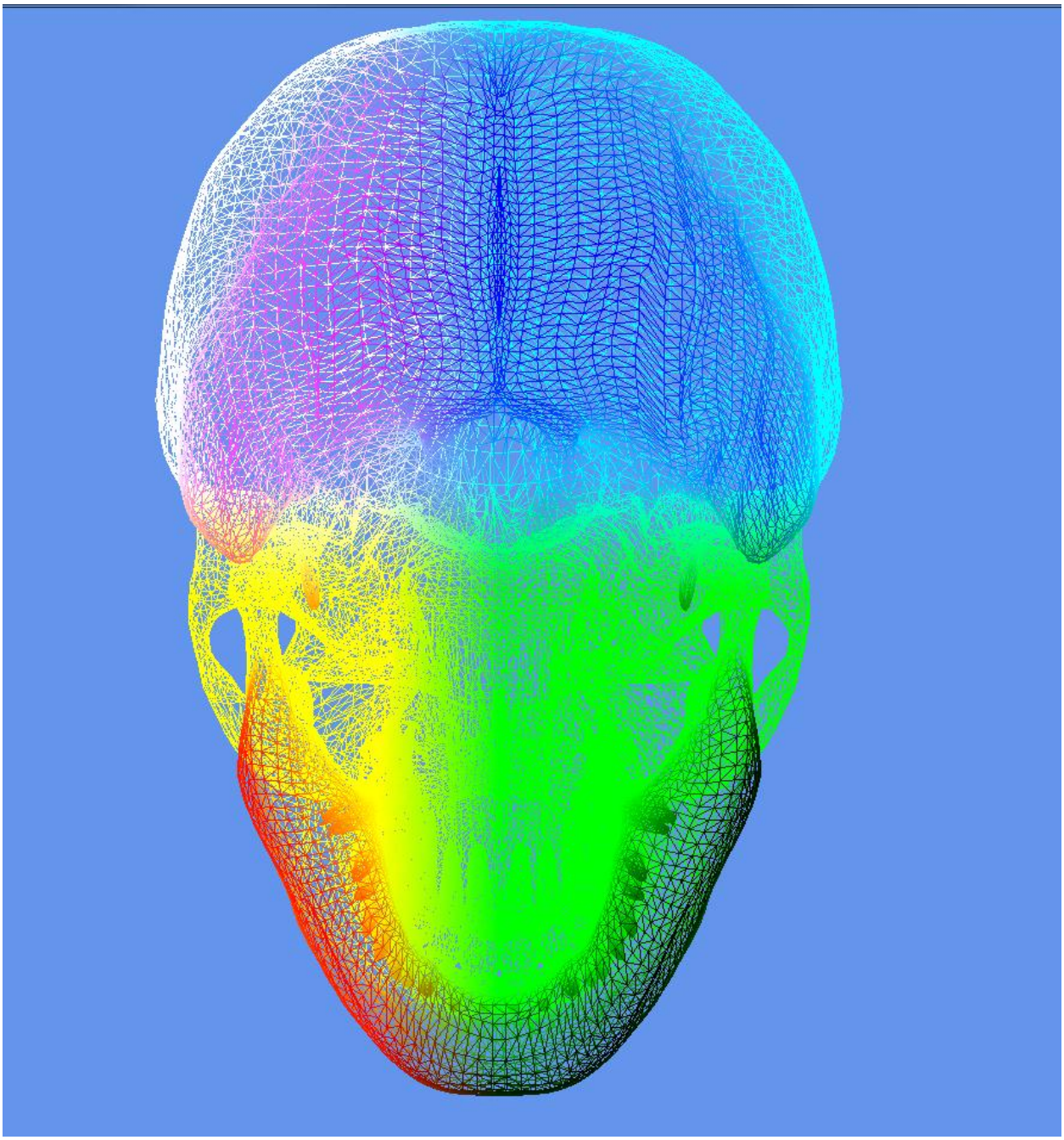
c) Add element color to AppData in VShaderModel

```
struct AppData
{
    float3 position : POSITION;
    float3 normal: NORMAL;
    float3 color : COLOR;

};
```

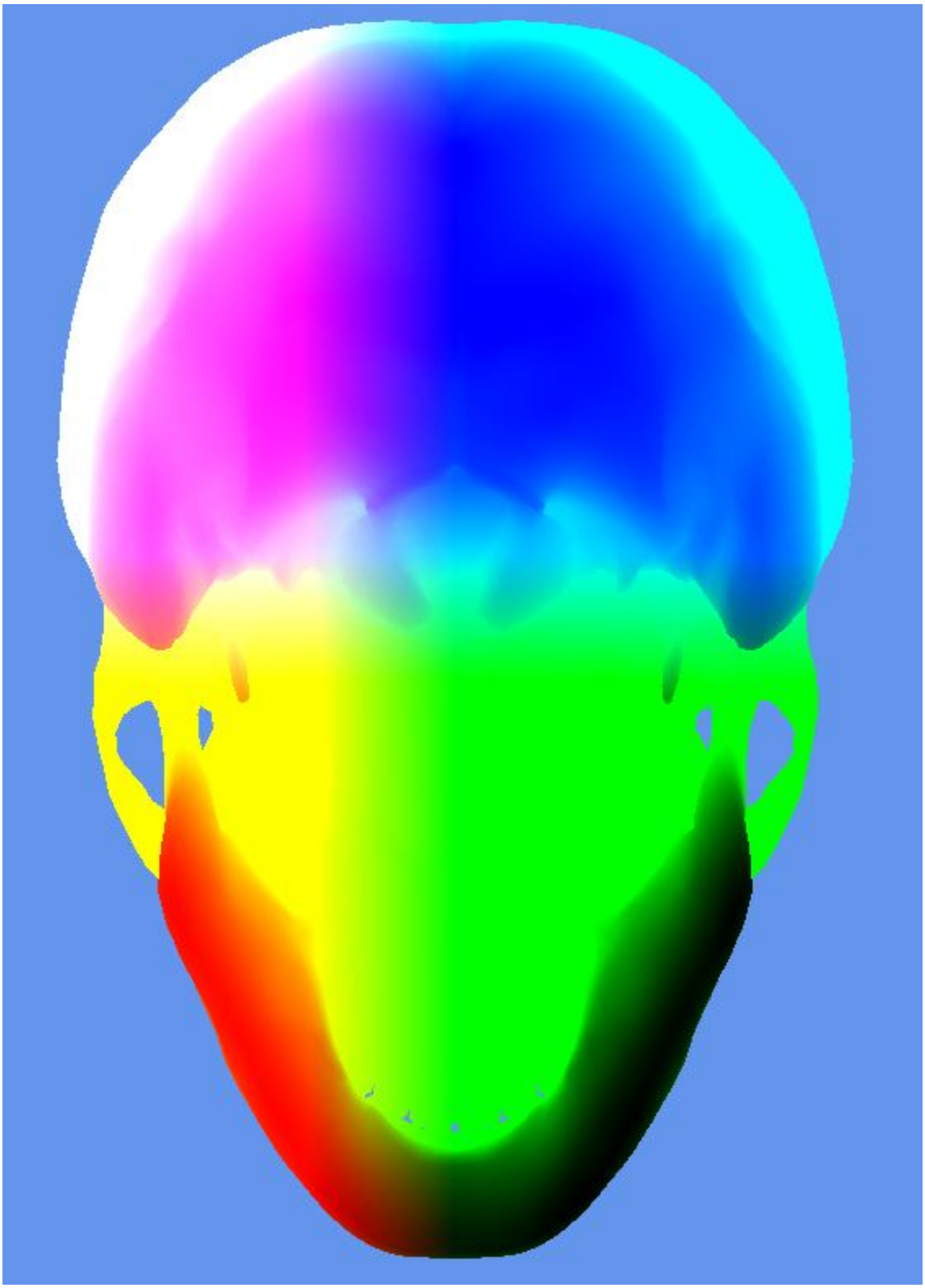
d) Add IN.Color to OUT.color

```
VOut VShader(AppData IN)
{
    VOut OUT;

    matrix wvp = mul(projectionMatrix, mul(viewMatrix, worldMatrix));
    OUT.position = mul(wvp, float4(IN.position, 1.0f));
    //IN.color = 0.1f;
    OUT.color = float4(IN.color,1.0f);
    //OUT.color = float4(0.0f,1.0f,0.0f, 1.0f);

    return OUT;
}
```

e) Applying Rasterizer State



f) Without Rasterizer State



ModelApp.cpp*

```
bool ModelApp::Init()
{
    if (!DXApp::Init())
        return false;


    InitBuffers();
    //InitShaders();
    InitPrecompiledShaders();
    InitConstantBuffers();
    //InitRasterizerState();

    OnResize();

    XMVECTOR focusPoint = XMVectorSet(0, 0, 0, 1);
    XMVECTOR upDirection = XMVectorSet(0, 1, 0, 0);

    XMVECTOR pos = XMLoadFloat3(&mCam.GetPosition());
    mCam.LookAt(pos, focusPoint, upDirection);

    return true;
}
```

g) I added 3 because now I have 3 elements in the struct.

```
void ModelApp::InitShaders(){

m_pDevice->CreateInputLayout(ied, 3  pVS->GetBufferPointer(), pVS->GetBufferSize(), &mpLayout);
```

2. **Try to send the world view projection matrix directly to the shader so that you can directly use in your shader.**

ModelApp.cpp

```
class ModelApp : public DXApp{
```

a) **I added World View Project = WVP**

```
enum ConstanBuffer
{
    CB_Application,
    CB_Frame,
    CB_Object,
    CB_WVP,
    NumConstantBuffers   //must be the last
};
```

b) **I added XMFLOAT4x4 mWVP**

```
//Matrices
XMFLOAT4X4 mWorld;
XMFLOAT4X4 mView;
XMFLOAT4X4 mProjection;
XMFLOAT4X4 mWVP;
```

c) **Safe Release CW_WP**

```
ModelApp::~ModelApp()
{
    Memory::SafeRelease(mpLayout);

    Memory::SafeRelease(mpVS);
    Memory::SafeRelease(mpPS);

    Memory::SafeRelease(mpBoxVB);
    Memory::SafeRelease(mpBoxIB);

    Memory::SafeRelease(mpConstantBuffers[CB_Application]);
    Memory::SafeRelease(mpConstantBuffers[CB_Frame]);
    Memory::SafeRelease(mpConstantBuffers[CB_Object]);
    Memory::SafeRelease(mpConstantBuffers[CB_WVP]);
}
```

d) Commented this Line

```
void ModelApp::OnResize(){
    DXApp::OnResize();

    mCam.SetLens(0.25f*MathHelper::Pi, AspectRatio(), 1.0f, 1000.0f);

    XMMATRIX P = mCam.Proj();
    //XMMATRIX P = mCam.get XMMatrixPerspectiveFovLH(XMConvertToRadians(45.0f), m_ClientWidth / m_ClientHeight, 0.1f, 100.0f);
    XMStoreFloat4x4(&mProjection, P);
    //@W
    //m_pImmediateContext->UpdateSubresource(mpConstantBuffers[CB_Application], 0, nullptr, &mProjection, 0, 0);
}
```

e) Commented this Line

```
void ModelApp::Update(float dt)
{
    XMMATRIX V = mCam.View();
    //XMMATRIX V = XMMatrixLookAtLH(eyePosition, focusPoint, upDirection);
    XMStoreFloat4x4(&mView, V);
    //@W
    //m_pImmediateContext->UpdateSubresource(mpConstantBuffers[CB_Frame], 0, nullptr, &mView, 0, 0);

    //mAngle += 90.0f * dt;
    XMVECTOR rotationAxis = XMVectorSet(0, 1, 1, 0);
```

f) Commented this line

```
XMMATRIX W = XMMatrixRotationAxis(rotationAxis, XMConvertToRadians(mAngle)) + XMMatrixTranslation(mesh_x,mesh_y,mesh_z);

XMStoreFloat4x4(&mWorld, W);
//@W
//m_pImmediateContext->UpdateSubresource(mpConstantBuffers[CB_Object], 0, nullptr, &mWorld, 0, 0);
```

g) Load the Value of mProjection

Make the operation WVP

Store the value of Matrix mWVP

Pointer Inmediate Context of CB_WP

```
XMMATRIX P = XMLoadFloat4x4(&mProjection);
XMMATRIX WVP = W * V * P;
XMStoreFloat4x4(&mWVP,WVP);

m_pImmediateContext->UpdateSubresource(mpConstantBuffers[CB_WVP], 0, nullptr, &mWVP, 0, 0);
```

h) Increase the number of ConstantBuffers

```
void ModelApp::Render(float dt)
{

    m_pImmediateContext->ClearRenderTargetView(m_pRenderTargetView, DirectX::Colors::CornflowerBlue);
    m_pImmediateContext->ClearDepthStencilView(m_pDepthStencilView, D3D11_CLEAR_DEPTH | D3D11_CLEAR_STENCIL, 1.0f, 0);
    //**********************
    m_pImmediateContext->IASetInputLayout(mpLayout);
    // select which primtive type we are using
    m_pImmediateContext->IASetPrimitiveTopology(D3D11_PRIMITIVE_TOPOLOGY_TRIANGLELIST);
    // select which vertex buffer to display
    UINT stride = sizeof(VERTEX);
    UINT offset = 0;
    m_pImmediateContext->IASetVertexBuffers(0, 1, &mpBoxVB, &stride, &offset);
    m_pImmediateContext->IASetIndexBuffer(mpBoxIB, DXGI_FORMAT_R32_UINT, 0);

    //shaders
    m_pImmediateContext->VSSetShader(mpVS, nullptr, 0);
    //@WT
    m_pImmediateContext->VSSetConstantBuffers(0, 4, mpConstantBuffers);

    m_pImmediateContext->PSSetShader(mpPS, nullptr, 0);

    m_pImmediateContext->RSSetState(mpWireframeRS);

    // draw the vertex buffer to the back buffer
    m_pImmediateContext->DrawIndexed(mModelIndexCount, 0, 0);

    //**********************
    HR(m_pSwapChain->Present(0, 0));
}
```

I) Added HR

```
void ModelApp::InitConstantBuffers(){
    // Create the constant buffers for the variables defined in the vertex shader.
    D3D11_BUFFER_DESC constantBufferDesc;
    ZeroMemory(&constantBufferDesc, sizeof(D3D11_BUFFER_DESC));

    constantBufferDesc.BindFlags = D3D11_BIND_CONSTANT_BUFFER;
    constantBufferDesc.ByteWidth = sizeof(XMMATRIX);
    constantBufferDesc.CPUAccessFlags = 0;
    constantBufferDesc.Usage = D3D11_USAGE_DEFAULT;

    HR(m_pDevice->CreateBuffer(&constantBufferDesc, nullptr, &mpConstantBuffers[CB_Application]));
    HR(m_pDevice->CreateBuffer(&constantBufferDesc, nullptr, &mpConstantBuffers[CB_Frame]));
    HR(m_pDevice->CreateBuffer(&constantBufferDesc, nullptr, &mpConstantBuffers[CB_Object]));
    HR(m_pDevice->CreateBuffer(&constantBufferDesc, nullptr, &mpConstantBuffers[CB_WVP]));
}
```

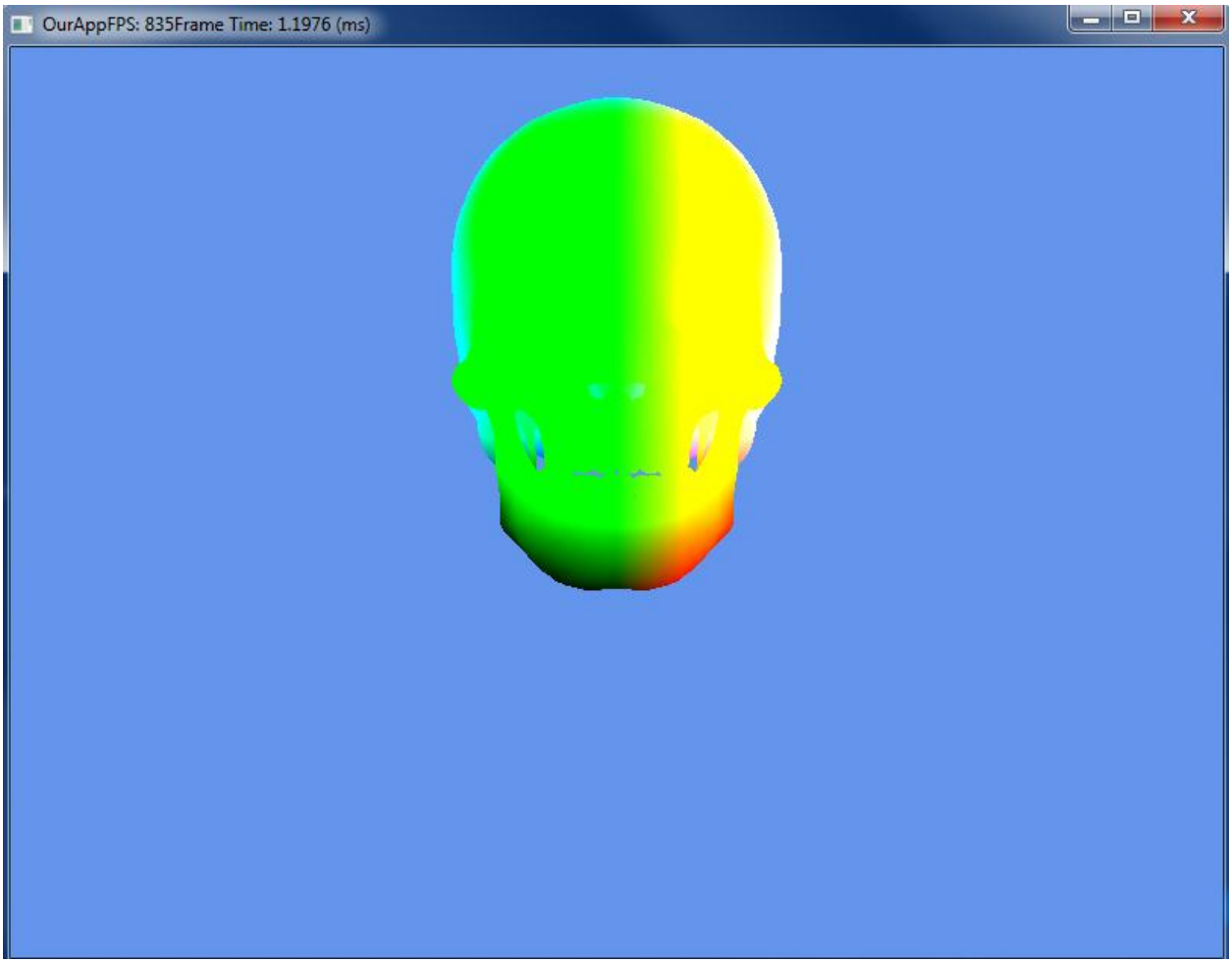
J) Added the Constant Buffer in

VShaderModel.hlsl

```
cbuffer PerWorld : register(b3){

    matrix worldViewProjectionMatrix;

}
```

K) Make the operation mul including the wordViewProjectionMatrix

```
VOut VShader(AppData IN)
{
    VOut OUT;

    //matrix wvp = mul(projectionMatrix, mul(viewMatrix, worldMatrix));

    OUT.position = mul(worldViewProjectionMatrix, float4(IN.position, 1.0f));
    //IN.color = 0.1f;
    OUT.color = float4(IN.color,1.0f);
    //OUT.color = float4(0.0f,1.0f,0.0f, 1.0f);

    return OUT;
}
```



3) Translate the mesh to a different position when you press a key.

a) Define Atrributes

ModelApp.cpp

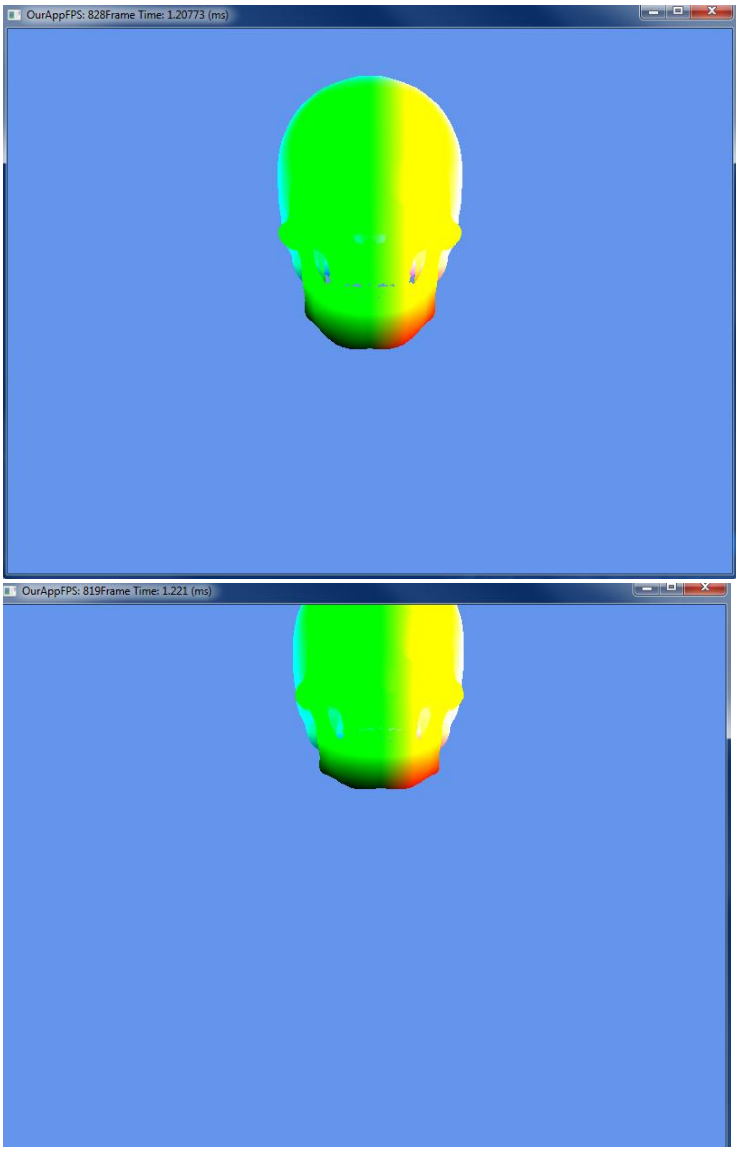
```
class ModelApp : public DXApp{
```

```
// Position Mesh

int mesh_x;
int mesh_y;
int mesh_z;
int count_pos_mesh;
```

```
void ModelApp::Update(float dt)
{
    //
```

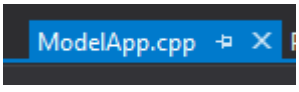
b) Asking if the key (I) was pressed, at the end I could move the object using the XMatrixTranslation with coordinates in the axis x, y, z.

```
XMVECTOR rotationAxis = XMVectorSet(0, 1, 1, 0);

if (GetAsyncKeyState('I') & 0x01)
{


    if (count_pos_mesh == 0)
    {
        mesh_x++;
    }

    if (count_pos_mesh == 1)
    {
        mesh_y++;
    }

    if (count_pos_mesh == 2)
    {
        mesh_z++;
        count_pos_mesh = 0;
    }

    count_pos_mesh++;
}



XMMATRIX W = XMMatrixRotationAxis(rotationAxis, XMConvertToRadians(mAngle)) + XMMatrixTranslation(mesh_x,mesh_y,mesh_z);
```



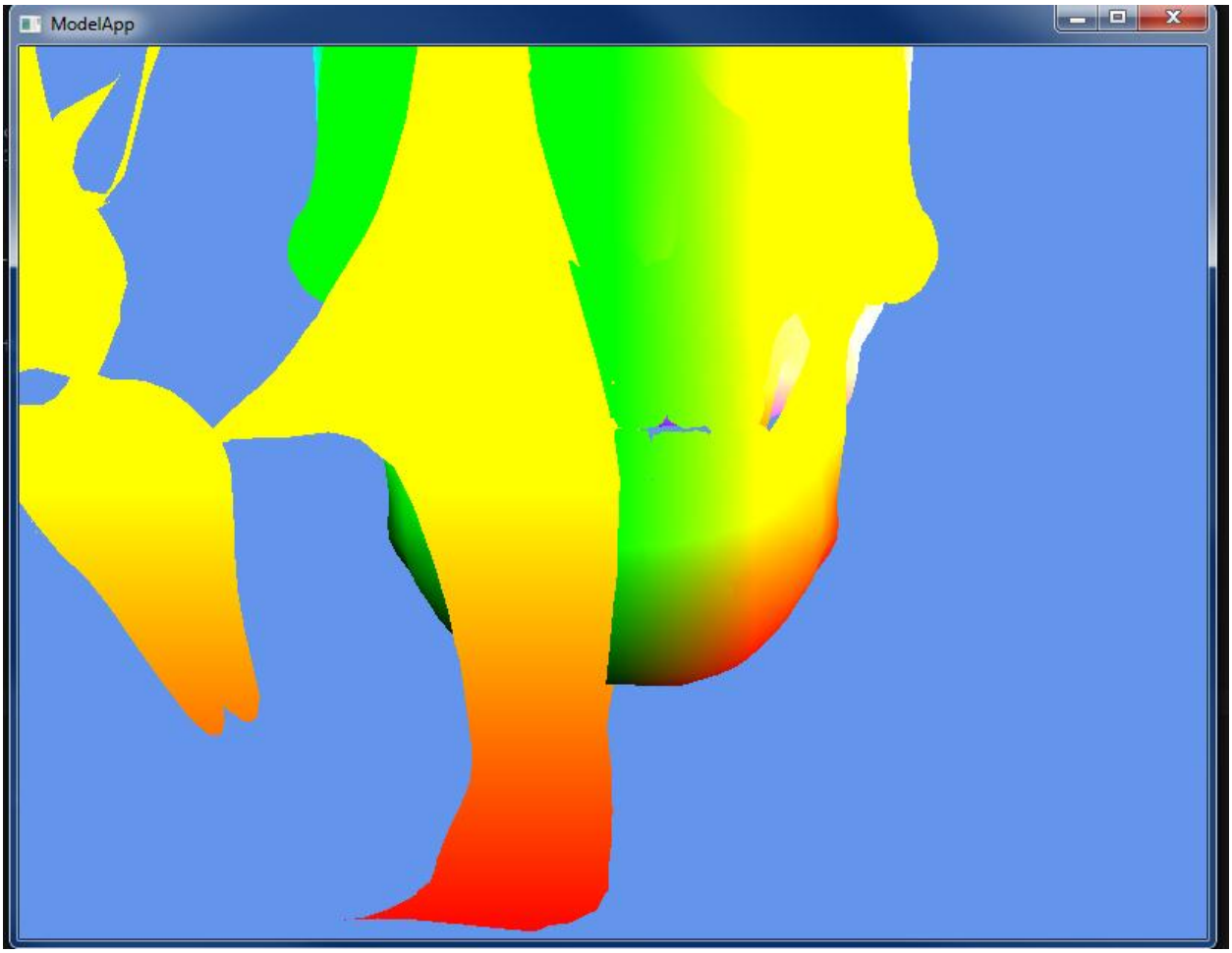
4)Show the first half of the triangles as solid and the second half as wireframe.

5) Show more than 1 skull

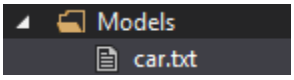
a) 

b) I sent the same skull only I translated the world moving the value of axis xyz.

```
void ModelApp::Render(float dt)
{

    m_pImmediateContext->ClearRenderTargetView(m_pRenderTargetView, DirectX::Colors::CornflowerBlue);
    m_pImmediateContext->ClearDepthStencilView(m_pDepthStencilView, D3D11_CLEAR_DEPTH | D3D11_CLEAR_STENCIL, 1.0f, 0);
    //**********************
    m_pImmediateContext->IASetInputLayout(mpLayout);
    // select which primtive type we are using
    m_pImmediateContext->IASetPrimitiveTopology(D3D11_PRIMITIVE_TOPOLOGY_TRIANGLELIST);
    // select which vertex buffer to display
    UINT stride = sizeof(VERTEX);
    UINT offset = 0;
    m_pImmediateContext->IASetVertexBuffers(0, 1, &mpBoxVB, &stride, &offset);
    m_pImmediateContext->IASetIndexBuffer(mpBoxIB, DXGI_FORMAT_R32_UINT, 0);

    //shaders
    m_pImmediateContext->VSSetShader(mpVS, nullptr, 0);
    //@WT
    m_pImmediateContext->VSSetConstantBuffers(0, 4, mpConstantBuffers);

    m_pImmediateContext->PSSetShader(mpPS, nullptr, 0);

    m_pImmediateContext->RSSetState(mpWireframeRS);

    // draw the vertex buffer to the back buffer
    m_pImmediateContext->DrawIndexed(mModelIndexCount, 0, 0);

    XMMATRIX t = XMMatrixTranslation(-2, -1, 2);

    XMStoreFloat4x4(&mWVP, t);

    m_pImmediateContext->UpdateSubresource(mpConstantBuffers[CB_WVP], 0, nullptr, &mWVP, 0, 0);
    m_pImmediateContext->DrawIndexed(mModelIndexCount, 0, 0);


    //**********************
    HR(m_pSwapChain->Present(0, 0));
}
```

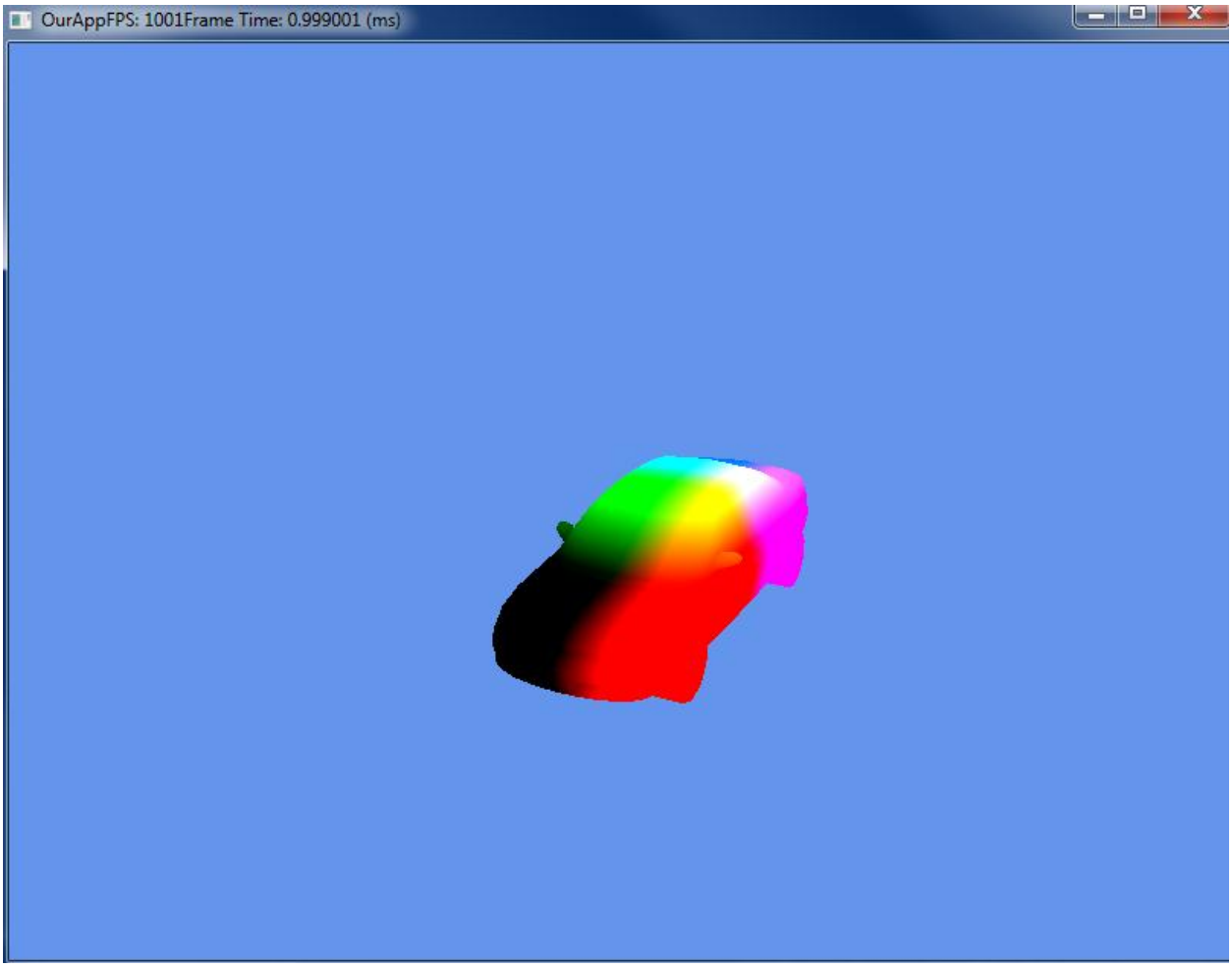


6) Including a Car




```
void ModelApp::InitBuffers(){
```

```
void ModelApp::InitBuffers(){

    std::vector<VERTEX> vertices2(5);
    //std::ifstream fin("Models/skull.txt");
    std::ifstream fin("Models/car.txt");
```



7) With one key enable and disable the rasterizer state.

a) I added another private attribute in the class ModelApp

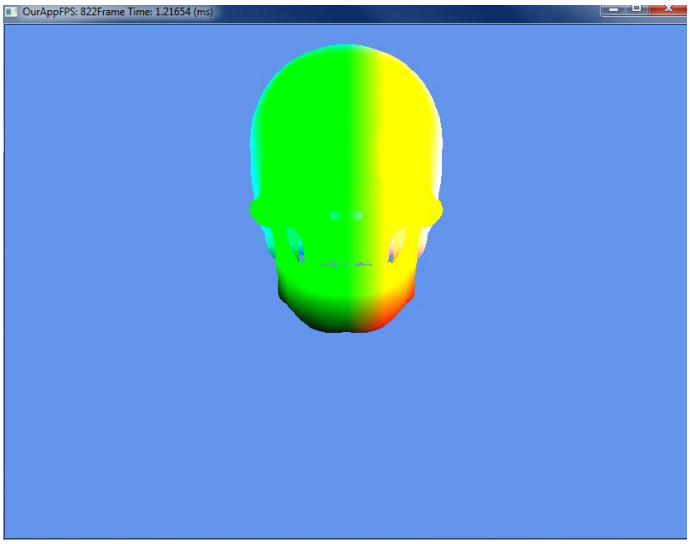
```
class ModelApp : public DXApp{
```

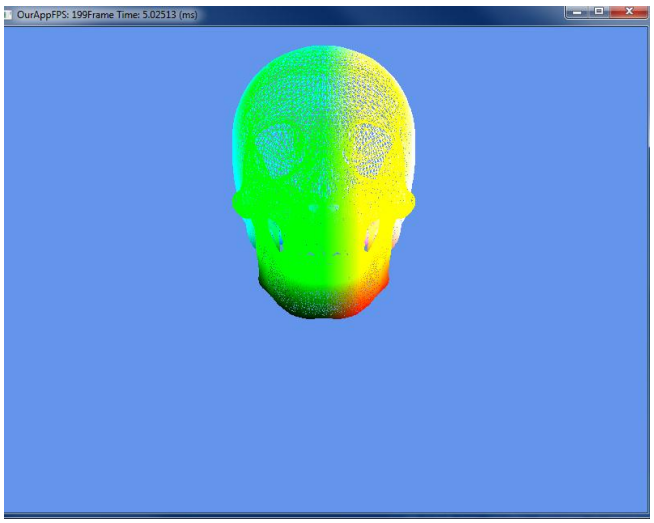
```
// Init Rasterize State
int count_rasterizer_state;
```

b) In the Update Method change the value of wireframeDesc.

```
void ModelApp::Update(float dt)
{
```

```
if (GetAsyncKeyState('K') & 0x01)
{
    if (count_rasterizer_state == 0)
    {

        D3D11_RASTERIZER_DESC wireframeDesc;
        ZeroMemory(&wireframeDesc, sizeof(D3D11_RASTERIZER_DESC));
        wireframeDesc.FillMode = D3D11_FILL_WIREFRAME; //D3D11_FILL_WIREFRAME; //
        wireframeDesc.CullMode = D3D11_CULL_NONE;
        wireframeDesc.FrontCounterClockwise = false;
        wireframeDesc.DepthClipEnable = true;
        HR(m_pDevice->CreateRasterizerState(&wireframeDesc, &mpWireframeRS));


    }

    if (count_rasterizer_state == 1)
    {

        D3D11_RASTERIZER_DESC wireframeDesc;
        ZeroMemory(&wireframeDesc, sizeof(D3D11_RASTERIZER_DESC));
        wireframeDesc.FillMode = D3D11_FILL_SOLID; //D3D11_FILL_WIREFRAME; //
        wireframeDesc.CullMode = D3D11_CULL_NONE;
        wireframeDesc.FrontCounterClockwise = false;
        wireframeDesc.DepthClipEnable = true;
        HR(m_pDevice->CreateRasterizerState(&wireframeDesc, &mpWireframeRS));



        count_rasterizer_state = -1;
    }


    count_rasterizer_state++;
}
```


OurAppFPS: 822Frame Time: 1.21654 (ms)


OurAppFPS: 199Frame Time: 5.02513 (ms)