Patrick Valk 500704269

Advanced Graphics Programming

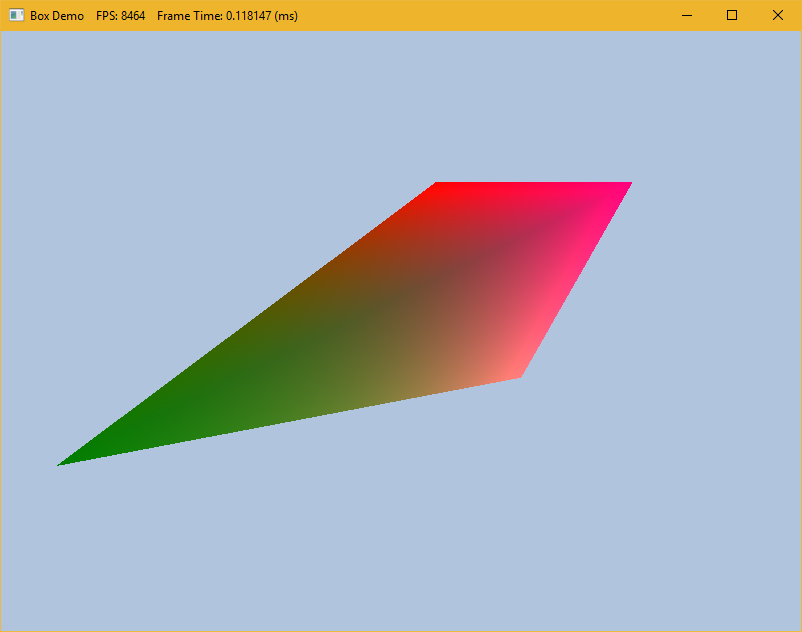
Assignment, part 1: Primitives

## Basic:

I chose to render a tetrahedron.

|  |
| --- |
| // Create vertex buffer  Vertex vertices[] =  {  { XMFLOAT3(-0.0f, -1.0f, -1.0f), (const float\*)&Colors::White },  { XMFLOAT3(-1.0f, +1.0f, -1.0f), (const float\*)&Colors::Magenta },  { XMFLOAT3(+1.0f, +1.0f, -1.0f), (const float\*)&Colors::Red },  { XMFLOAT3(+0.0f, -0.0f, +1.0f), (const float\*)&Colors::Green }  }; |

## Intermediate:

I decided to let the user stretch the model by holding middle mouse button and dragging the mouse. The CPU calculates the new mesh coordinates.

|  |
| --- |
| // Create vertex buffer  Vertex vertices[] =  {  { XMFLOAT3(-0.0f, -1.0f, -1.0f \* temp1), (const float\*)&Colors::White },  { XMFLOAT3(-1.0f, +1.0f, -1.0f \* temp1), (const float\*)&Colors::Magenta },  { XMFLOAT3(+1.0f, +1.0f, -1.0f \* temp1), (const float\*)&Colors::Red },  { XMFLOAT3(+0.0f, -0.0f, +1.0f \* temp1), (const float\*)&Colors::Green }  }; |

# Assignment, part 2: Texturing

## Basic:

I wrapped an Samsung s7 texture around a model I made with the geometrygGenerator. I manually set the texture points from the model.

## C:\Users\TheCore\AppData\Local\Microsoft\Windows\INetCache\Content.Word\3.png



|  |
| --- |
| XMFLOAT2 a[] = { XMFLOAT2(0.25f, 0.3f),  XMFLOAT2(0.25f, 0.6f),  XMFLOAT2(0.5f, 0.6f),  XMFLOAT2(0.5f, 0.3f),    XMFLOAT2(0.75f, 0.3f),  XMFLOAT2(1, 0.3f),  XMFLOAT2(1, 0.6f),  XMFLOAT2(0.75f, 0.6f),    XMFLOAT2(0.25f, 0),  XMFLOAT2(0.25f, 0.3f),  XMFLOAT2(0.5f, 0.3f),  XMFLOAT2(0.5f, 0),    XMFLOAT2(0.25f, 0.6f),  XMFLOAT2(0.5f, 0.6f),  XMFLOAT2(0.5f, 1),  XMFLOAT2(0.25f, 1),    XMFLOAT2(0, 0.6f),  XMFLOAT2(0.25f, 0.6f),  XMFLOAT2(0.25f, 0.3f),  XMFLOAT2(0, 0.3f),    XMFLOAT2(0.5f, 0.3f),  XMFLOAT2(0.75f, 0.3f),  XMFLOAT2(0.75f, 0.6f),  XMFLOAT2(0.5f, 0.6f),  }; |

## Intermediate:

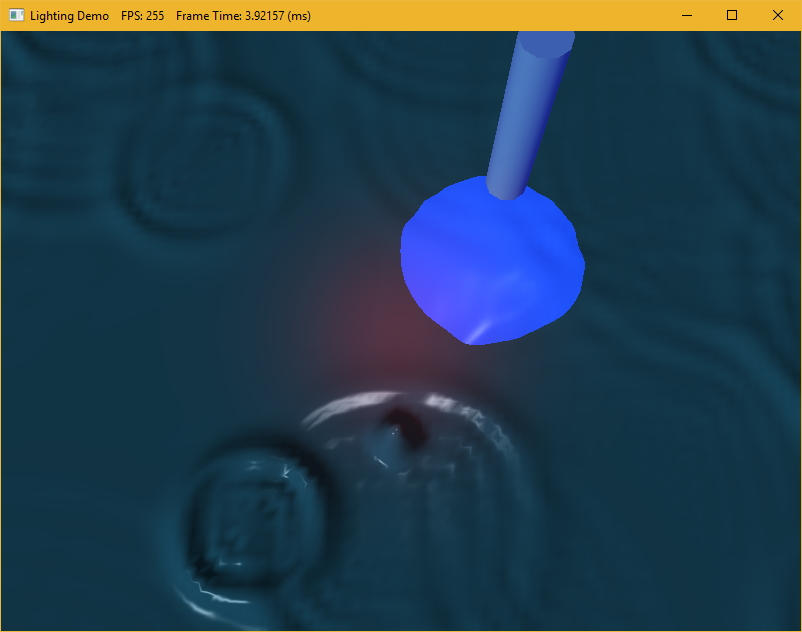
## 

|  |
| --- |
| HR(D3DX11CreateShaderResourceViewFromFile(md3dDevice,  L"Textures/Texture2.bmp", 0, 0, &mDiffuseMapSRV, 0));  HR(D3DX11CreateShaderResourceViewFromFile(md3dDevice, L"Textures/Texture3.png", 0, 0, &mDiffuseMapSRV2, 0)); if (gUseTexure)  {  float4 result = 0;  float4 resultColor = gDiffuseMap2.Sample(samAnisotropic, pin.Tex);  float resultAlpha = gDiffuseMap2.Sample(samAnisotropic, pin.Tex).a;  if (resultAlpha > 0)  resultColor = gDiffuseMap2.Sample(samAnisotropic, pin.Tex);  else  resultColor = gDiffuseMap.Sample(samAnisotropic, pin.Tex);  result = (resultColor);  texColor = result;  } |

# Assignment, part 3: Lighting

## Basic:

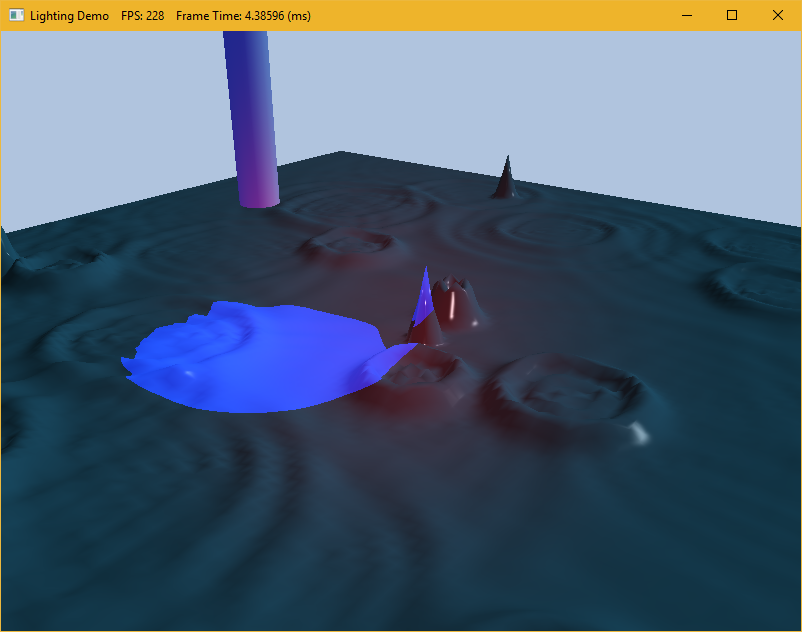
For basic I made a blue want with a blue light, I chose to have a harsh border around it because I thought it should look like a flash light.



|  |
| --- |
| // Directional light.  mDirLight.Ambient = XMFLOAT4(0.2f, 0.2f, 0.2f, 1.0f);  mDirLight.Diffuse = XMFLOAT4(0.5f, 0.5f, 0.5f, 1.0f);  mDirLight.Specular = XMFLOAT4(0.5f, 0.5f, 0.5f, 1.0f);  mDirLight.Direction = XMFLOAT3(0.57735f, -0.57735f, 0.57735f);    // Point light--position is changed every frame to animate in UpdateScene function.  mPointLight.Ambient = XMFLOAT4(0.0f, 0.0f, 1.0f, 1.0f);  mPointLight.Diffuse = XMFLOAT4(0.7f, 0.7f, 0.7f, 1.0f);  mPointLight.Specular = XMFLOAT4(0.7f, 0.7f, 0.7f, 1.0f);  mPointLight.Att = XMFLOAT3(0.0f, 0.1f, 0.0f);  mPointLight.Range = 25.0f;  mPointLight.Position.x = 10;  mPointLight.Position.y = 20.0f;  mPointLight.Position.z = 10;  // Spot light--position and direction changed every frame to animate in UpdateScene function.  mSpotLight.Ambient = XMFLOAT4(1.0f, 0.0f, 0.0f, 1.0f);  mSpotLight.Diffuse = XMFLOAT4(1.0f, 0.0f, 0.0f, 1.0f);  mSpotLight.Specular = XMFLOAT4(1.0f, 1.0f, 1.0f, 1.0f);  mSpotLight.Att = XMFLOAT3(1.0f, 0.0f, 0.0f);  mSpotLight.Spot = 96.0f;  mSpotLight.Range = 20.0f;  mSpotLight.Position.x = -10;  mSpotLight.Position.y = 30.0f; |

## Intermediate:

For intermediate I made a laser light pointing from the camera perspective to the water, the user can change the size of the laser by holding the middle mouse button and dragging the mouse.

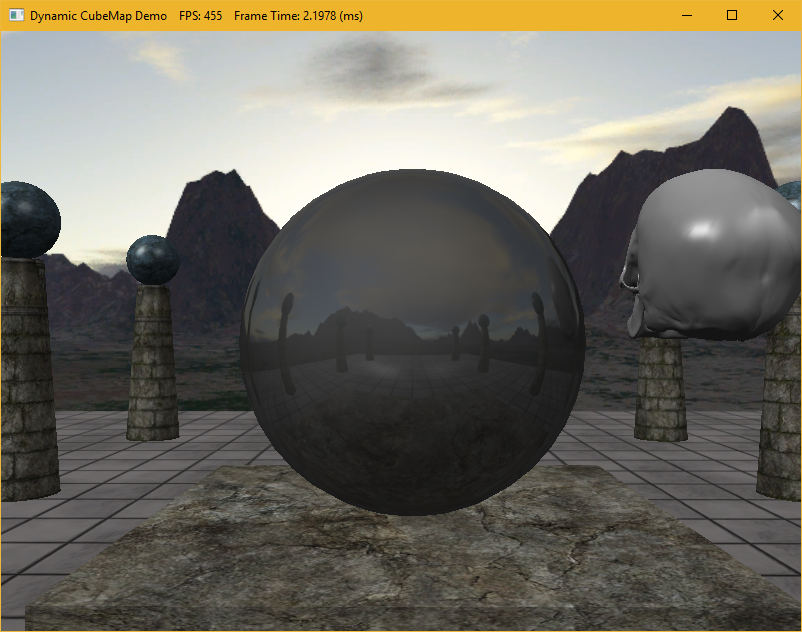


# Assignment, part 4: Shading

## Basic:

I let the user change the chrome reflection by holding the middle mouse button and dragging from left to right.





|  |
| --- |
| if ((btnState & MK\_MBUTTON) != 0)  {  debounce = true;  temp1 = (x / 600.0f);  mCenterSphereMat.Reflect = XMFLOAT4(temp1, temp1, temp1, 1.0f);  mCenterSphereMat.Ambient = XMFLOAT4(temp1, temp1, temp1, 1.0f);  mCenterSphereMat.Diffuse = XMFLOAT4(temp1, temp1, temp1, 1.0f);  mCenterSphereMat.Specular = XMFLOAT4(temp1, temp1, temp1, 1.0f);  } |