

TerrainClass.hApp.cppBox.cpp\*Box(Graphics & gfx, float x, float y, float z)

3DX

1#include "Box.h"

2#include "additional\_headers.h"

3#include "Texture.h"

4#include "Surface.h"

5

6

7Box::Box(Graphics & gfx, float x, float y, float z) :

8x(x), y(y), z(z)

9{

10

11if (!isStaticallyBound())

12{

13struct Vertex

14{

15

16DirectX::XMFLOAT3 pos;

17

18};

19struct PixelShaderConstants

20{

21struct

22{

23float r;

24float g;

25float b;

26float a;

27} face\_colors[8];

28};

29const PixelShaderConstants cb2 =

30{

31{

32{ 1.0f,1.0f,1.0f },

33{ 1.0f,0.0f,0.0f },

34{ 0.0f,1.0f,0.0f },

35{ 1.0f,1.0f,0.0f },

36{ 0.0f,0.0f,1.0f },

37{ 1.0f,0.0f,1.0f },

38{ 0.0f,1.0f,1.0f },

39{ 0.0f,0.0f,0.0f },

40}

41};

42

43std::vector<Vertex> vertices;

44vertices.resize(8);

45

46vertices[0].pos = DirectX::XMFLOAT3(-1.0f, -1.0f, -1.0f);

47vertices[1].pos = DirectX::XMFLOAT3(1.0f, -1.0f, -1.0f);

48vertices[2].pos = DirectX::XMFLOAT3(-1.0f, 1.0f, -1.0f);

49vertices[3].pos = DirectX::XMFLOAT3(1.0f, 1.0f, -1.0f);

50vertices[4].pos = DirectX::XMFLOAT3(-1.0, -1.0f, 1.0f);

51vertices[5].pos = DirectX::XMFLOAT3(1.0f, -1.0f, 1.0f);

52vertices[6].pos = DirectX::XMFLOAT3(-1.0f, 1.0f, 1.0f);

53vertices[7].pos = DirectX::XMFLOAT3(1.0f, 1.0f, 1.0f);

54

55

56

57AddStaticBind(std::make\_unique<VertexBuffer>(gfx, vertices));

58AddStaticBind(std::make\_unique<PSConstBuff<PixelShaderConstants>>(gfx, cb2,1w));

59AddStaticBind(std::make\_unique<PixelShader>(gfx, L"ColorBlenderPS.cso"));

60

61

62

63auto vs = std::make\_unique<VertexShader>(gfx, L"ColorBlenderVS.cso");

64auto vsBlob = vs->GetVBlob();

65AddStaticBind(std::move(vs));

66std::vector<WORD> indices =

67{

680,2,1, 2,3,1,

691,3,5, 3,7,5,

702,6,3, 3,6,7,

714,5,7, 4,7,6,

720,4,2, 2,4,6,

730,1,4, 1,5,4

74

75};

76

77AddStaticBind(std::make\_unique<IndexBuff>(gfx, indices));

78

79const std::vector<D3D11\_INPUT\_ELEMENT\_DESC> ied =

80{

81{ "POS",0,DXGI\_FORMAT\_R32G32B32\_FLOAT,0,0,D3D11\_INPUT\_PER\_VERTEX\_DATA,0 }

82

83};

84AddStaticBind(std::make\_unique<PrimitiveTopology>(D3D11\_PRIMITIVE\_TOPOLOGY\_TRIANGLELIST));

85

86} else

87{

88SetIndexBufferFromStatic();

89

90}

91//struct VSMaterialConstant

92//{

93// DirectX::XMATRIX model;

94// DirectX::XMATRIX worldviewProj;

95// DirectX::XMFLOAT3 eyePos;

96// //float padding;

97//};

98//VSConst;

99//VSConst.model = DirectX::XMMatrixTranspose(GetTransformation());

100//VSConst.worldviewProj = DirectX::XMMatrixTranspose(GetTransformation() \* gfx.GetCamera());

101//VSConst.eyePos = DirectX::XMFLOAT3(1.0f,1.0f,1.0f);

102

103//AddBind(std::make\_unique<VSConstBuff<VSMaterialConstant>>(gfx, VSConst));

104AddBind(std::make\_unique<TransformationBuffer>(gfx,\*this));

105

106}

107

108void Box::Update(float ft)

109{

110pitch += ft;

111yaw += ft;

112roll += ft;

113

Solution Explorer

Search Solution Explorer (Ctrl+)

Timer.h

Window.h

WinHeader.h

Resource Files

Shaders

ColorBlenderPS.hlsl

ColorBlenderVS.hlsl

ConstantColorPS.hlsl

PhongLightingPS.hlsl

PhongLightingVS.hlsl

TextureMappingPS.hlsl

TextureMappingVS.hlsl

Source Files

Bindables

Bindables.cpp

IndexBuff.cpp

InputLayout.cpp

PixelShader.cpp

PrimitiveTopology.cpp

SamplerState.cpp

Texture.cpp

VertexBuffer.cpp

VertexShader.cpp

App.cpp

Box.cpp

Camera.cpp

CustomException.cpp

DirectionalLight.cpp

DrawableBase.cpp

Graphics.cpp

Keyboard.cpp

Model.cpp

Mouse.cpp

PointLight.cpp

SpotLight.cpp

Surface.cpp

TerrainClass.cpp

Team Explorer - Synchronization

Search Work Item

Synchronization | Direct3D

Successfully pushed branch BufferProblems to origin.

Branch: BufferProblems

Sync | Fetch | Pull | Push | Actions

Incoming Commits

Fetch | Pull

There are no incoming commits.

Outgoing Commits

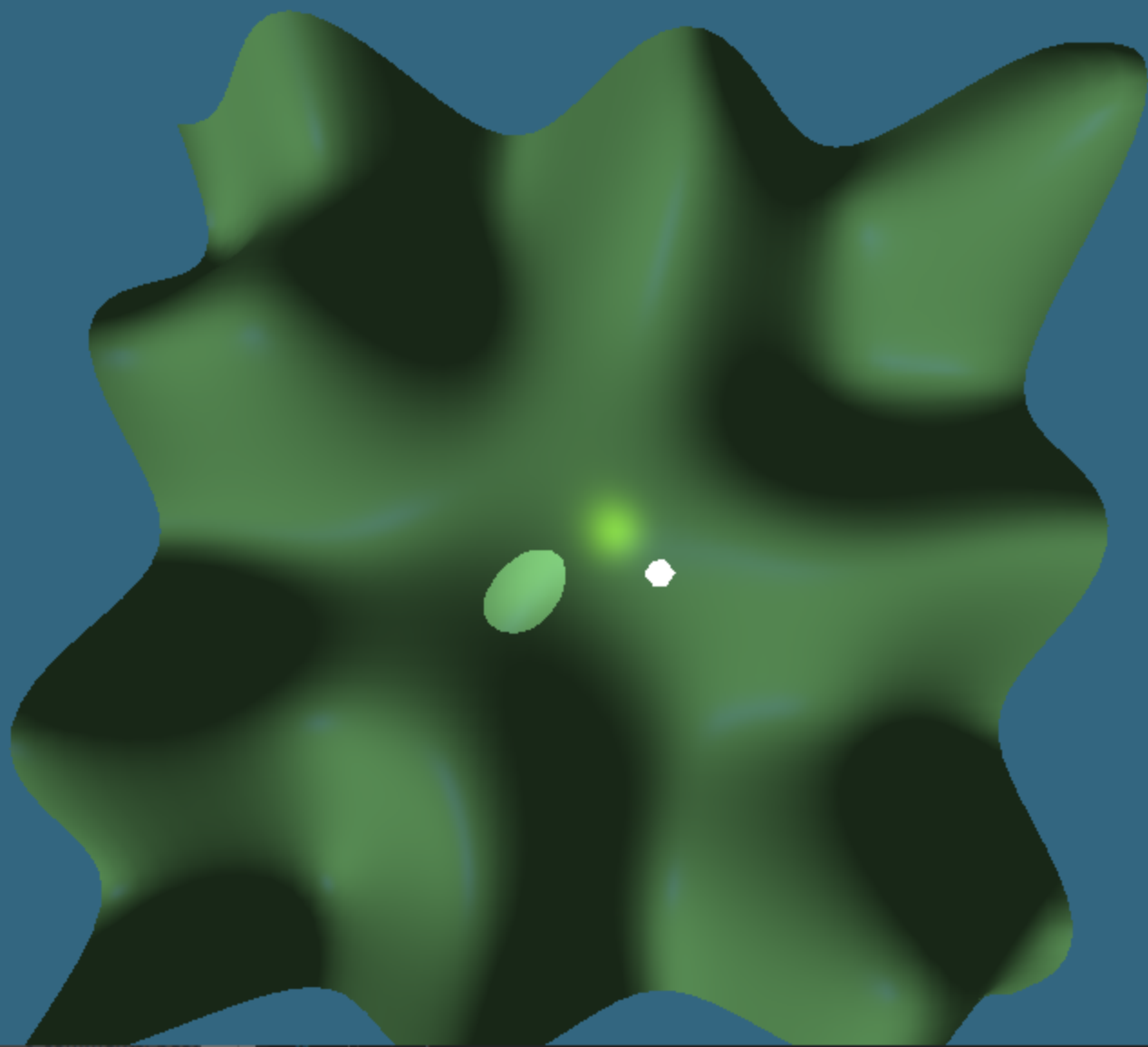
Push | View Summary

There are no outgoing commits.

```
1  #pragma once
2  #include "Window.h"
3  #include "Camera.h"
4  #include "DirectionalLight.h"
5  #include "PointLight.h"
6  #include "SpotLight.h"
7  #include "TerrainClass.h"
8  class App
9  {
10 public:
11     App();
12     ~App();
13     int Go();
14 private:
15     void Update();
16 private:
17     Window wnd;
18     Camera cam;
19     DirectionalLight dirLight;
20     PointLight pointLight;
21     SpotLight spotLight;
22     float x, y, z, dt;
23     float last_x;
24     float last_y;
25     float mPhi, mTheta, mRadius;
26     //Model m;
27     TerrainClass bl{ wnd.gfx(), cam.GetPosition(), 50, 50, 160.0f, 160.0f };
28     Box dl{ wnd.gfx(), 10.0f, 80.0f, -10.0f };
29
30
31     };
```

Search Solution Explorer (Ctrl+;)

- ▷ InputLayout.h
- ▷ MatHelper.h
- ▷ PixelShader.h
- ▷ PrimitiveTopology.h
- ▷ SamplerState.h
- ▷ Texture.h
- ▷ TransformationBuffer.h
- ▷ VertexBuffer.h
- ▷ VertexShader.h
- ▷ Drawables
  - ▷ Effects
  - ▷ objects
    - ▷ Box.h
- ▷ DrawableBase.h
- ▷ DrawableSource.h
- ▷ additional\_headers.h
- ▷ App.h
- ▷ Camera.h
- ▷ Colors.h
- ▷ CustomException.h
- ▷ DirectionalLight.h
- ▷ Graphics.h
- ▷ Keyboard.h
- ▷ Light.h
- ▷ LightSource.h
- ▷ Model.h
- ▷ Mouse.h
- ▷ PointLight.h
- ▷ resource.h
- ▷ SpotLight.h
- ▷ Surface.h
- ▷ TerrainClass.h
- ▷ Timer.h
- ▷ Window.h
- ▷ WinHeader.h
- ▷ Resource Files
- ▷ Shaders





Hello

