

Texture Splatting

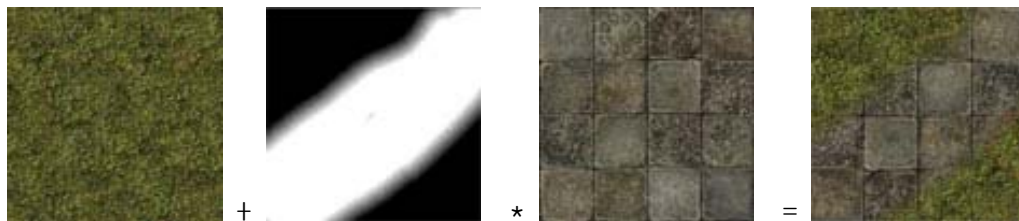
작 성 일 : 2007년 4월 2일

작 성 자 : 김보성

텍스처 스프레팅이란..

Charles Bloom이란 아저씨가 머라머라 설명한다. 무슨말이나 이게..난하나도모르겠다. 간단히 말하면 알파텍스처를 실시간으로 만들어서 일반텍스처와 멀티텍스처로 혼합하여 나타내는 기법이다. 이렇게 간단한걸 어렵게 설명해주고계신다.

그림으로 보면이해가 빠르다



기본 텍스처 + 실시간알파텍스처 * 일반텍스처 = 혼합된 텍스처

이런식으로 만드는데...

단계를 알아보면

1. 기본텍스처와 일반텍스처를 리소스에서 읽어와 생성.
2. 빈 실시간알파용 텍스처를 생성.
3. 기본텍스처렌더링
4. 실시간알파처리
5. 알파와 혼합된텍스처 기본텍스처 위에 렌더링.

이렇게 있다.

그럼 단계별로 알아보자.

1. 기본텍스처와 일반텍스처를 리소스에서 읽어와 생성.
 - 기본으로 찍혀주어야할 텍스처와 와 그위에 찍혀야할 일반텍스처를 리소스에서 불러들여 생성한다.

Ex)

```
D3DXCreateTextureFromFile(g_pd3dDevice, "baseTexture.jpg", &g_BaseTexture);
```

//기본

```
D3DXCreateTextureFromFile(g_pd3dDevice, "tile.jpg", &g_Texture) // 일반
```

코드에서 **LoadTextures()** 함수참고

2. 빈 실시간알파용 텍스처를 생성.

- 이제 실질적으로 사용할 알파텍스처를 생성한다.

Ex)

```
g_pd3dDevice->CreateTexture( TEXALPHASIZE, TEXALPHASIZE, 1, 0,  
                             D3DFMT_A8R8G8B8, D3DPPOOL_MANAGED,  
                             &g_Alphamap, NULL )
```

그리고 텍스처에 Lock,Unlock를 걸어 텍스처의 모든색상을 0으로 초기화한다.

여기서는 D3DFMT_A8R8G8B8 타워이지만 D3DFMT_A8로 바꾸어서 알파만가지는 텍스처를 생성하여 용량을 줄일수있다. 코드에 **CreateAlphaMaps()** 함수참고

3. 기본텍스처렌더링

그냥 바닥에 말그대로 기본적으로 찍어주면되겠다.

코드 **BaseTextureRender()** 참고.

4. 실시간알파처리

이부분이 제일중요하다. 이부분이 다다. 실시간으로 알파를 그려내기.

- 우선 마우스에 브러쉬를달아 알파를 그려야 하므로 피킹처리가 필요하겠다.

마우스를 3d좌표로 변화하여 모두다 알고있는 dx예제샘플의 **IntersectTriangle()** 이용해서 피킹된지점을 구한다. 이부분은 코드의 **PickingProcess()**를 참고하기 바란다.

이제 실제적으로 알파를 그리는 **DrawAlphamaps()**에대해서 알아본다.

그다음 알파 텍스처의 한픽셀에대한 현재 월드의 상의 한점의 크기를 구한다.

현재는 가로세로 1*1 짜리 하나다

```
PixSize = 1 / TEXALPHASIZE;
```

월드의 상의 한점의 크기를 구했으니 이제 브러쉬의 크기를 구해보자.

코드를 예로들면 텍스처의 UV값은 0.0~1.0이라는 사실을 알고있을것이다.

지금 계산을 편하게 하기위해 예제에서도 실제 월드좌표가 가로세로 0.0~1.0의 값을 갖는다. 이로서 텍스처의 UV값과 월드좌표는 같다.

이제 브러쉬를 만들어야겠다. 브러쉬란 포토샵에서와 같이 원모양의 중심으로부터

점차적으로 밖으로 부드럽게 알파가먹는 그런 것을 말한다 브러쉬를 크게 2단계로 나누자 명확히 찍혀야되는부분과 부드럽게 나타내야하는 부분으로 나눈다.

부드럽게나타내야하느부분은 최대브러쉬의 크기에서 명확시 찍혀야되는 부분을 빼면 된다



가운데 부분의 흰색이 진한부분은 명확히 찍혀야되는부분이고나머지는 부드럽게 찍혀야하는 부분이다.

마우스가 피킹된지점이 브러쉬원의 한가운데이며 점차적으로 부드럽게 나간다 물론 이부분은 단계수정이 가능하다. 우선 브러시의 최대크기와 명확히 찍혀야할부분을 구하자 최대크기와 명확히찍힐부분의 반지름값만 있으면 구할수있다.

$\text{HalfBrushSize} = \text{BrushSize}/2;$ <-최대브러쉬의 반지름

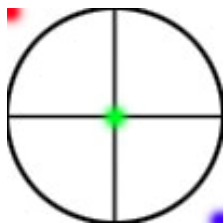
$\text{HalfSmoothSize} = \text{SmoothSize}/2;$ <-명확히찍힐부분의 반지름

최대브러쉬의 반지름과 월드의 상의 한점의 크기를 이용해서 알파텍스처에 에디트 할 크기를 구할수있다.

$\text{EditSize} = \text{fHalfBrushSize}/\text{PixSize};$

그럼 에디트할 픽셀의 시작점과 끝점을 얻어야한다.

TexPosU , TexPosV 는 현재 마우스위치에서 피킹한점을 텍스처의 uv좌표로 바꾸어준 값이다.



위그림에서 빨간점이 에디트할 시작UV, 파란점이 에디트가끝나는 UV, 연두색점이 현재 피킹한 월드좌표에대한 텍스처의 UV좌표가된다

$\text{StartPosU} = \text{TexPosU} - \text{EditSize}$

$\text{StartPosV} = \text{TexPosV} - \text{EditSize}$

$\text{EndPosU} = \text{TexPosU} + \text{EditSize}$

$\text{EndPosV} = \text{TexPosV} + \text{EditSize}$

이런식으로 시작점과 끝점을 구해냈다 에디트할 부분을 설정한것이다..

이제 시작점좌표부터 끝점좌표까지 픽셀을 읽어들여 수정만하면되겠다.

전에 위에서 최대브러쉬의 반지름과 명확히찍힐부분의 반지름을 구했다.

이것을 이용해서 명확히찍힐부분의 픽셀좌표와 부드럽게 찍힐부분의 픽셀 좌표를 얻어낼수있다.

먼저 가운데 점에서 명확히찍힐부분의 길이 즉 아까 반지름값을 범위안에잇다면 모든픽셀의 알파값을 알파값의 맥스값 255(0xff)로 만들어주면 되는것이다.

일일이 순차적인 픽셀과 가운데 중심좌표의 길이가 명확히찍힐부분의 길이 보다작다면 모두 255값으로 만들어주는것이다.

그후에 명확히찍힐부분의 의끝길이부터 브러쉬의 최대크기까지의 범위에 있는 픽셀 들은 점차적으로 알파값을 줄여주면 되는것이다.

// 최대사이즈에 포함될 우 최대사이즈까지의

// 선정보간한 알파값을 구한다.

Lenth -= fHalfSmoothSize;

Smooth = fHalfBrushSize - fHalfSmoothSize;

dwChangeColor= DWORD((Smooth - Lenth) / Smooth * 0xFF000000);

여기까지 알파를 그리는 부분이 끝났다. 중요한부분은 거의 다지나왔다.

코드에서 DrawAlphamaps()을 참고..

이제 알파 텍스처와 일반텍스처를 혼합하여 뿌려주기만하면 끝이다..

5.알파와 혼합된텍스처 기본텍스처 위에 렌더링.

이제 혼합해보자

여기서는 텍스처 스테이지가 2장이 필요하다 그래서 Geforce2MX이상의 그래픽카드가 필요하다.(ps2도 2장까지 지원)

0번스테이지에 알파텍스처를 설정하고

1번스테이지에 일반텍스처를 설정한다.

알파블렌딩이 쓰이므로 당연히 알파는 TRUE가되겠다.

g_pd3dDevice->SetRenderState(D3DRS_ALPHABLENDENABLE, TRUE);

//0번스테이지의 ALPHAOP에 알파텍스처를 설정한다

g_pd3dDevice->SetTextureStageState(0, D3DTSS_ALPHAOP, D3DTOP_SELECTARG1);

g_pd3dDevice->SetTextureStageState(0, D3DTSS_ALPHAARG1, D3DTA_TEXTURE);

//1번스테이지의 COLOROP 에 알파텍스처를 설정한후에 ALPHAOP에
0번 스테이지에있는 알파텍스처를 설정한다.

이렇게되면 이제 그려주기만하면된다.

```
g_pd3dDevice->SetTextureStageState(1, D3DTSS_COLOROP, D3DTOP_SELECTARG1);  
g_pd3dDevice->SetTextureStageState(1, D3DTSS_COLORARG1, D3DTA_TEXTURE);  
g_pd3dDevice->SetTextureStageState(1, D3DTSS_ALPHAOP, D3DTOP_SELECTARG1);  
g_pd3dDevice->SetTextureStageState(1, D3DTSS_ALPHAARG1, D3DTA_CURRENT);
```

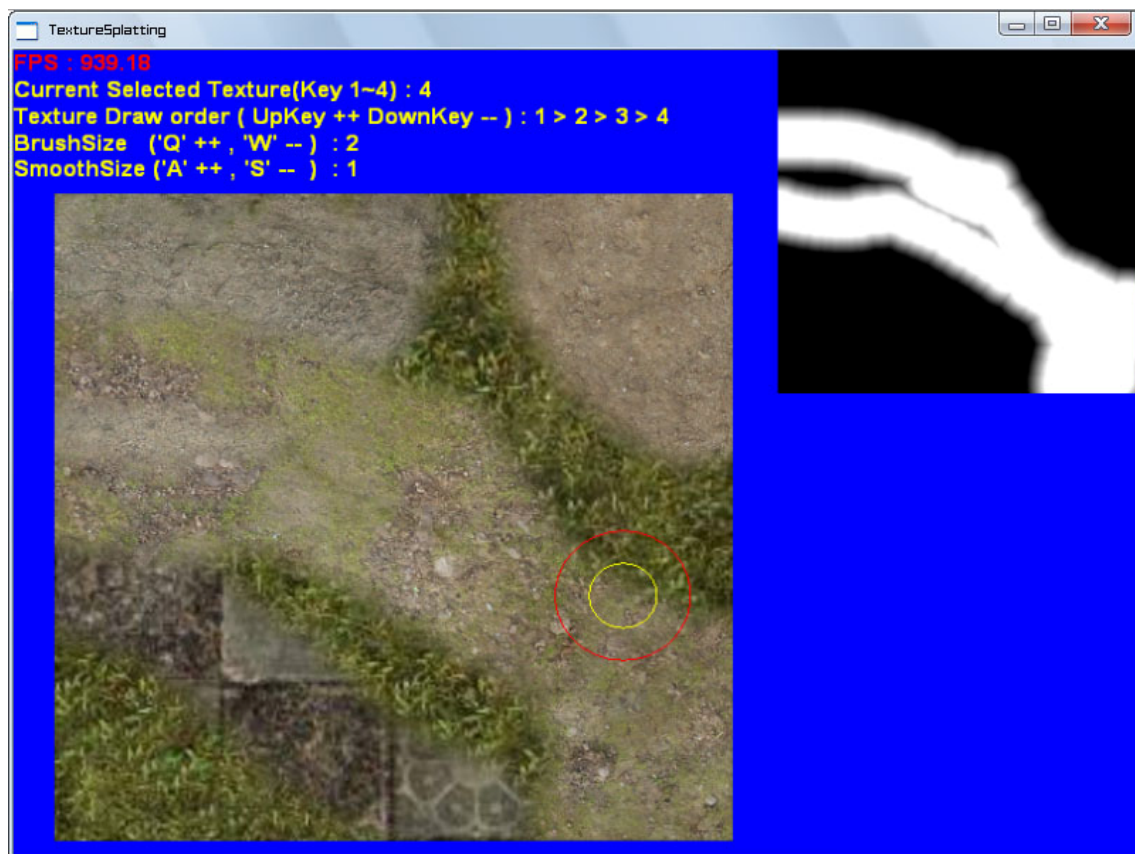
```
g_pd3dDevice->SetTexture(0, Alphamap);  
g_pd3dDevice->SetTexture(1, Texture);  
g_pd3dDevice->SetStreamSource( 0, g_pVB, 0, sizeof(CUSTOMVERTEX) );  
g_pd3dDevice->DrawPrimitive(D3DPT_TRIANGLESTRIP, 0, 2);
```

그리고꼭 뒤에그려질놈들의 스테이지에 영향을 주지않기위해 멀티텍스처 1번 셋팅
을 사용을 중지 한다.

```
g_pd3dDevice->SetTextureStageState( 1, D3DTSS_COLOROP, D3DTOP_DISABLE );  
g_pd3dDevice->SetTextureStageState( 1, D3DTSS_ALPHAOP, D3DTOP_DISABLE );
```

코드에서 `AlphaTextureRender()`를 참고

최종결과화면



현재 1장의 베이스텍스처와 4장의 혼합텍스처를 그린것이다. 오른쪽 상단에는 맨마지막의 4번째 혼합텍스처의 알파텍스처가 그려지고있다.

***** End of Document *****