

# Tutorial 11: Alpha Channel Transparency

by Victor Saar

## Content

This is another nice tutorial about alpha blending. It covers alpha blending with the alpha channel of a texture loaded from a bitmap. We just load a texture, which includes an alpha channel, after we have enabled alpha blending. We use the same texture as in the last tutorial, but this time we have the alpha channel included. On the image below you can see the alpha channel of this texture.

## application.cpp

Enabling alpha blending is the same like in the last tutorial. For more information take a look at it or tutorial 7. The following code enables everything we need.

```
//alpha blending enabled
m_pDirect3DDevice->SetRenderState(D3DRS_ALPHABLENDENABLE, true);
//source blend factor
m_pDirect3DDevice->SetRenderState(D3DRS_SRCBLEND, D3DBLEND_SRCALPHA);
//destination blend factor
m_pDirect3DDevice->SetRenderState(D3DRS_DESTBLEND, D3DBLEND_INVSRCALPHA);

//alpha from texture
m_pDirect3DDevice->SetTextureStageState(0, D3DTSS_ALPHAARG1, D3DTA_TEXTURE);
```

## main.cpp

This file isn't much changed. The most interesting change is done in the loading function of the texture. Again we use `D3DXCreateTextureFromFileExA()`. We only change two parameters. The first is the file name, which is now `leaf.png`. The second is the color key value. We set this parameter to `0`, which deactivates the color key and automatically creates the alpha channel from the file.

```
D3DXCreateTextureFromFileExA(g_App.GetDevice(), //device
    "leaf.png", //file name
    D3DX_DEFAULT, //width
    D3DX_DEFAULT, //height
    D3DX_DEFAULT, //mip levels
    NULL, //usage
    D3DFMT_UNKNOWN, //texture color format
    D3DPPOOL_MANAGED, //memory class
    D3DX_DEFAULT, //filter
    D3DX_DEFAULT, //mip filter
    0, //no color key (NEW)
    NULL, //source info
    NULL, //palette
    &pLeafTexture); //texture object
```

That's all we have to do to use alpha channel transparency.