

## Direct3D - 2. diel

Ostatný diel bol zameraný na nastavenie vývojového prostredia pre DirectX. V dnešnom diele sa zameriam na inicializáciu Direct3D a dozviete sa množstvo nových informácií.

### Terminológia

Device (zariadenie):

Pod týmto pojmom sa v Direct3D rozumie či už fyzická, alebo logická grafická karta. Fyzická je taká, ktorú máte nainštalovanú v počítači. Aj ako logická sa tvári vtedy, keď má dva výstupy na monitor, resp. televízor a podobne. Fyzická a logická sa od seba líšia funkcionalitov. K tomu sa však dostaneme asi až niekedy neskôr. Page flipping (Page swapping)

Je jedna z metod, ktorou sa obraz dostáva na obrazovku. Funguje to tak, že sa vytvoria dva buffery. Obraz sa zapisuje do jedného z nich. Keď sa má renderovať, tak sa obsahy bufferov prehodí. Na obrazovku sa tak dostane buffer, v ktorom sú zapísané obrazové dáta, čiže jednotlivé pixely. Druhý buffer sa vyčistí a je pripravený na zápis. Takto sa to neustále opakuje. Je to jednoducho ako princíp otáčania strán v knihe.

Keď sú k dispozícii dva buffery nazývame to double buffering. Keď sú buffery tri, ide o triple buffering. Buffer, ktorého obsah sa vykresľuje na obrazovku sa nazýva front buffer, ostatné sú back buffery.

Keby neexistovala táto technika, tak by všetky objekty za sebou nechávali "šmuhy" na tom mieste, kde sa naposledy nachádzali. Game loop

Game loop je programový cyklus, ktorý sa opakuje, až kým sa hra nemá ukončiť. V ňom prebieha všetko vykresľovanie, spracúvanie fyziky, zvukov a ostatné programové operácie. Matrices (matice)

N krát M rozmerné pole. Matriciam venujem jeden samostatný článok, pretože na pochopenie sú trochu zložitejšie, ale ak chcete vytvoriť nejaký poriadnejší 3D program, budete ich musieť ovládať.

Inicializácia

Na inicializáciu Direct3D slúži funkcia Direct3DCreate8. LPDIRECT3D8 Direct3DCreate8( UINT verzia\_SDK )

verzia\_SDK:

Vždy nastavte na D3D\_SDK\_VERSION.

Keď si ho inicializujete Direct3D, tak treba inicializovať device. Na to slúži funkcia CreateDevice. HRESULT

LPDIRECT3D8::CreateDevice(

UINT Adapter,

D3DDEVTYPE DevType,

HWND Okno,

DWORD Parametre,

\*D3DPRESENT\_PARAMETERS Parametre\_obrazu,

\*LPDIRECT3DDEVICE8 Device

)

Adapter:

Ako parameter zatiaľ zadávajte D3DADAPTER\_DEFAULT

DevType:

Typ devicu. Máte dve možnosti: D3DDEVTYPE\_HAL a REF. Keď zadáte HAL, tak budete mať k dispozícii plnú akceleráciu grafiky grafickou kartou. Keď zadáte REF vezíu, tak sa všetko bude vykonávať softvérovou. REF verziu sa oplatí používať vtedy, keď potrebujete v rámci kompatibility napísať nejaké funkcie na spracovanie grafiky.

Okno:

Okno, do ktorého sa bude vykresľovať.

Parametre:

Sem budeme zadávať D3DCREATE\_SOFTWARE, resp. HARDWARE\_VERTEXPROCESSING.

Keď zadáte SOFTWARE, tak sa budú vertexy spracúvať softvérovou, v paenom prípade ich bude spracúvať grafická karta. Existuje ešte aj možnosť MIXED, o tej si však povieme niekedy neskôr (ak vôbec).

Keď inicializujete device, tak sa oplatí otestovať, či grafická karta podporuje vami vybranú verziu vertex processingu.

Parametre\_obrazu:

Dátová štruktúra, ktorá obsahuje informácie o tom, ako sa má device inicializovať.

Device:

Odkaz na device, ktorý sa má práve vytvoriť. Pozastavím sa pri D3DPRESENT\_PARAMETERS. Je to dátová štruktúra obsahujúca niekoľko členov. Mojim cieľom nie je ich všetky vysvetliť (bolo by to zbytočne zdĺhavé), preto vám odporúčam pozrieť sa do dokumentácie DirectX. Nemusíte sa však obávať, že by som nič nevysvetlil. Vysvetlím jednoducho tie parametre, s ktorými budeme pracovať, čiže časom možno aj všetky.

Vo výpise nebudem uvádzať zachytávanie chýb, pretože by zdrojový kód len

zbytočne predlžovali a komplikovali. V pribalených zdrojových kódoch ale budú všetky chyby ošetrované.

Ako by to malo vyzeráť

```
LPDIRECT3D8 lpD3D; // direct3D
```

```
LPDIRECT3DDEVICE8 lpDev; // device
```

```
...
```

```
lpD3D = Direct3DCreate8(D3D_SDK_VERSION);
```

```
D3DPRESENT_PARAMETERS d3dPP;
ZeroMemory(&d3dPP, sizeof(d3dPP));
d3dPP.Windowed = true; // true - oknový režim
d3dPP.hDeviceWindowed = hMainWindow; // okno aplikácie
d3dPP.BackBufferCount = 1; // počet backbufferov, 1, èiže double buffering
d3dPP.SwapEffect = D3DSWAPEFFECT_DISCARD;
lpD3D->CreateDevice(D3DADAPTER_DEFAULT, D3DDEVTYPE_HAL, hMainWindow,
D3DCREATE_HARDWARE_VERTEXPROCESSING, &d3dPP, &lpDev); Touto postupnosťou príkazov by sme mali
inicializované nutné minimum Direct3D8.
```

#### Rendering

Rendering je proces, pri ktorom sa na obrazovku výstupného zariadenia (v našom prípade monitora) kreslí scéna. Od verzie Direct3D8 sa dá renderovať veľmi jednoducho. Stačí vám na to štyri rýchlozapamätateľné funkcie. Sú nimi Clear(), BeginScene(), EndScene() a Present(). HRESULT LPDIRECT3DDEVICE8::Clear( DWORD count, const D3DRECT rect, DWORD flags, D3DCOLOR color, float z, DWORD stencil )

#### Count, Rect:

Tieto parametre nastavte zatiaľ na NULL, resp. 0.

#### Flags:

D3DCLEAR\_TARGET - pri vykresľovaní každého frame za vymaže obrazovú plochu

#### Color:

Farba, ktorou sa bude buffer mazať. Môžeme ju zadať buď hexadecimálnou hodnotou (napr. 0x00FFFFFF pre bielu farbu), ale makrom D3DCOLOR\_XRGB(r,g,b).

#### Z

Keďže táto hodnota je normalizovaná, môže byť od 0 do 1. 1 znamená, že sa vyčistí celý backbuffer, 0 znamená, že sa ešte nebude vôbec.

#### Stencil:

Tento parameter nastavte na 0, resp. NULL.

Ako ste už asi zistili, Clear() slúži na vyčistenie aktívneho buffera.

Funkcie BeginScene() a EndScene() slúžia na začatie, resp. ukončenie oblasti, v ktorej sa bude renderovať. Tieto funkcie nemajú žiadne parametre.

Poslednou je funkcia Present(). Má štyri parametre, ktoré môžete s kľudným svedomím nastaviť na NULL.

Základná funkcia na rendering by mohla vyzerať asi takto. lpDev->Clear(0, NULL, D3DCLEAR\_TARGET, 0x00ffffff, 1.0f, 0);

```
lpDev->BeginScene();
```

```
// v tejto oblasti môže prebiehať vykresľovanie
```

```
lpDev->EndScene();
```

```
lpDev->Present(NULL,NULL,NULL,NULL); Záver
```

V dnešnom diele sme sa dozvedeli ako Direct3D inicializovať a ako renderovať. Ďalší diel bude zameraný na kreslenie jednoduchých útvarov (primitív) ako sú trojuholníky, štvorce a podobne.

Zdrojové kódy: Direct3D2.rar