

Fehér Krisztián

Direct2D

DirectX

programozás

egyszerűen

BBS-INFO Kiadó, 2018.

Minden jog fenntartva! A könyv vagy annak oldalainak másolása, sokszorosítása csak a kiadó írásbeli hozzájárulásával történhet.

A könyv nagyobb mennyiségben megrendelhető a kiadónál:
BBS-INFO Kiadó, info@bbs.hu Tel.: 407-17-07

A könyv megírásakor a szerző és a kiadó a lehető legnagyobb gondossággal járt el. Ennek ellenére, mint minden könyvben, ebben is előfordulhatnak hibák. Az ezen hibákból eredő esetleges károkért sem a szerző, sem a kiadó semmiféle felelősséggel nem tartozik, de a kiadó szívesen fogadja, ha ezen hibákra felhívják figyelmét.

ISBN 978-615-5477-60-7
E-book ISBN 978-615-5477-62-1
Kiadja a BBS-INFO Kft., Budapest
Felelős kiadó: a BBS-INFO Kft. ügyvezetője
Nyomdai munkák: Biró Family Nyomda
Felelős vezető: Biró Krisztián

TARTALOMJEGYZÉK

TARTALOMJEGYZÉK	3
1. Előszó	5
2. A Direct2D madártávlathól	8
2.1. Mi az a Direct2D?.....	8
2.2. Miért Direct2D?.....	9
2.3. Mit csinál a Direct2D?.....	10
2.4. Mire használhatjuk a Direct2D-t?.....	10
2.5. Milyen gyors a Direct2D?.....	10
3. A Direct2D és a Visual Studio	12
3.1. A fejlesztőeszköz kipróbálása.....	14
3.2. Régi típusú függvények - figyelmeztetés kikapcsolása.....	21
3.3. Külső könyvtárak hozzáadása projektekhez.....	21
3.4. Hibakeresés Visual Studioban.....	22
4. Direct2D alapok	24
4.1. Fejlécállományok.....	24
4.2. A rajzolás folyamata.....	25
4.2.1. Direct2D Koordinátarendszer.....	28
4.2.2. Erőforrások kezelése.....	28
4.3. Egy egyszerű Direct2D alkalmazás.....	29
5. A rajzolás alapjai Direct2D-ben	34
5.1. A rajzterület.....	35
5.2. A képernyő törlése.....	35
5.3. Alapvető képernyőadatok kezelése.....	36
5.3.1. Bitképmérethatár lekérdezése.....	36
5.3.2. Pixelformátum lekérdezése.....	36
5.3.3. Pixelméret lekérdezése.....	37
5.3.4. RenderTarget méretének lekérdezése.....	38
5.3.5. DPI lekérdezése.....	38
5.3.6. DPI megváltoztatása.....	38
5.3.7. Élsimítás beállítása.....	39
5.3.8. Szöveg élsimításának beállítása.....	39
5.4. Színek megadása.....	40
5.5. Ecset (Brush) használata.....	40

5.5.1. Egyszerű ecset.....	41
5.5.2. Átmenetes ecset	42
5.5.3. Bitkép alapú ecset.....	47
5.5.4. Ecset átlátszósága	53
5.6. Vonaltulajdonságok beállítása	53
5.7. Flush	55
5.8. Kivágás megadása	56
5.9. Transzformációk alkalmazása.....	56
5.10. Pontok rajzolása	59
5.11. Vonalrajzolás	59
5.12. Ellipszis rajzolása.....	60
5.13. Négyzet rajzolása.....	61
5.14. Lekerekített sarkú négyzet rajzolása	63
5.15. Szöveg kirajzolása.....	65
5.15.1. További formázási metódusok	68
5.16. Bitképek kezelése	69
6. Haladó rajzolási technikák	74
6.1. Összetett geometria rajzolása.....	74
6.2. Meshek kezelése és tesszelláció	78
6.2.1. Szoftveres tesszelláció.....	82
6.2.2. Vonalrajzolás háromszögekkel.....	84
6.3. GeometrySink vagy Mesh?.....	84
6.3.1. Különböző színű háromszögek	84
6.3.2. Szupergyors rajzolás háromszögekkel	85
6.4. Rétegek használata	86
6.4.1. Rétegek transzformációja	89
6.4.2. Speciális esetek, effektezés	89
6.5. Képkivágás gyorsabban.....	90
6.6. Off-screen renderelés megvalósítása.....	92
6.7. A Direct3D és a Direct2D összekapcsolása	94
6.8. Teljesítményjavítás	98
6.8.1. Alapértelmezett képernyőtörlés felülírása.....	98
6.8.2. BeginDraw-EndDraw hívások minimalizálása	99
6.8.3. Off-screen renderelés	99
7. Háromdimenzió és a Direct2D	100
7.1. Hátsólap eldobás.....	104
8. Záró gondolatok.....	107
9. Függelék	108
10. Ajánlott irodalom	111

1. Előszó

Olvasóink közül biztosan sokan gondoltak már arra, hogy milyen jó lenne grafikus programozási ismereteiket, megoldásaikat a számítógépekben megtalálható videokártyák (GPU) számítási kapacitásának bevonásával használni.

Erre napjainkban számos lehetőség van programozási szempontból, ám a legtöbbjük megtanulása tetemes időt és energiát igényel.

Felmerül a kérdés: nem lehetne egyszerűbben?

Nos, a válasz: igen!

A Microsoft Direct2D programozási felülete pontosan ezt az igényt hivatott és képes is lefedni. Könyvünk segítségével az ehhez szükséges tudást lehet megszerezni.

Kinek szól a könyv?

Könyvünk a C/C++ programozásban alapvető jártassággal rendelkező olvasókhoz szól elsősorban. A Microsoft Visual Studio Community 2017 fejlesztőeszközt fogjuk használni a könyv példáinak elkészítéséhez, ezért ennek az eszköznek a minimális ismerete is elengedhetetlen lesz.

Amennyiben a fenti területek nem hangzanak ismerősen, akkor sincsen semmi baj, ugyanis a könyv érdemben történő gyakorlati alkalmazásához szükséges alapismereteket megszerezhetjük akár a szintén a BBS-Info kiadó által megjelentetett "**Grafikus és játékkalkalmazások fejlesztése**" című könyvünkből is, mely napjainkban jelenleg az egyetlen olyan magyar nyelven elérhető könyv, mely naprakész információkat tartalmaz, átfogó módon. Ebből a szempontból jelen könyvünk az imént említett mű folytatásának is tekinthető, de teljesértékű könyv is, önállóan is használható.

A Direct2D programozását alacsony szintű programozási eszközökkel, WIN32API-n keresztül fogjuk végezni, C/C++ nyelvre támaszkodva, erre készülünk.

Mit tanulhatunk meg a könyvből?

A könyv ismeretanyagának elsajátításával az olvasó képes lesz önállóan olyan, grafikus megjelenítést használó alkalmazásokat készíteni, melyek kihasználják a modern GPU-kban rejlő óriási teljesítmény-potenciált.

Noha a Direct2D alapvetően kétdimenziós megjelenítésről szól, előljáróban eláruljuk, hogy a háromdimenziós megjelenítés használatának mikéntjét is bemutatjuk Direct2D segítségével.

Mivel a Direct2D a Direct3D API-ra épül, ezért elkerülhetetlenül is meg fogunk szerezni olyan ismereteket, melyek birtokában a későbbiekben a Direct3D programozása sem lesz olyan nehéz, legalábbis, ha erre adja valaki a fejét.

Kinek nem szól a könyv?

Könyvünk nem a Direct3D alapú megjelenítés programozásának bemutatását tűzte ki célul, így erre könyvünk nem tér ki. Nem tárgyaljuk továbbá a GPU alapokon végezhető párhuzamosítási technikákat sem.

Mindazonáltal arra biztatunk minden kedves olvasót, hogy bátran ismerje meg ezeket a könyvünk szempontjából rokon szakterületeket is.

A modern grafikus programozás nem triviális, nem egyszerű feladat. Minden kezdet nehéz, ám az első lépés megtétele után lehetőségek egész kincsestára nyílik meg előttünk. Tegyük egy próbát, megéri!

A könyv jelölésmódja, illusztrációk

Az egyes függvények, adatszerkezetek leírásánál dőlt betűvel megadjuk a paramétereket és azok típusait is.

Ahol a megértést és persze a gyakorlati alkalmazást segíti, az összetettebb adatszerkezeteket, konstansokat is megadjuk. Például így:

```
void SetAntialiasMode(  
    D2D1_ANTIALIAS_MODE antialiasMode  
) = 0;
```

```
typedef enum {  
    D2D1_ANTI_ALIAS_MODE_PER_PRIMITIVE = 0,  
    D2D1_ANTI_ALIAS_MODE_ALIASED      = 1  
} D2D1_ANTI_ALIAS_MODE;
```

Könyvünk a forráskódokat szürke háttérrel kiemelve mutatja:

```
D2D1_RECT_F negyzet;
```

A könyv számos illusztrációt tartalmaz, melyek elsősorban szemléltető jellegűek és adott esetben eltérhetnek attól, amit az olvasó a saját számítógépén fog majd látni.

Letölthető mellékletek

A kiadó weboldaláról minden bemutatott alkalmazás teljes forráskódja letölthető. Ezek nem komplett projektek, melyeket a Visual Studioval meg lehet nyitni, hanem egy létrehozott alapértelmezett projekt kódjaként használhatóak, egyszerű szöveges másolás-beillesztés módszerrel. Ez egyszerűbb, átláthatóbb és helytakarékosabb megoldás, mintha nagyméretű, bonyolult szerkezetű, teljes Visual Studio projekteket tanulmányoznánk.

A szerzőről

A szerző hivatásos szoftvertesztelő, minőségbiztosítási tanácsadó, diplomás német irodalmár, a Magyar Térinformatikai Társaság (HUNAGI) egyéni szakértője.

Gyerekkorában autodidakta módon tanult meg programozni, az évek során számos programozási nyelvvel megismerkedett. megszerzett tudását előszeretettel használja alternatív, kísérleti alkalmazások készítésére, melyek egy része ingyenesen elérhető, sőt vannak köztük nyílt forráskódúak is. A szerző fejleszt Windows desktop, Android és webes környezetekre.

Szakterülete a digitális grafika programozása, valamint digitális térképpalkalmazások készítése. Sok időt fordít saját térinformatikai keretrendszerének fejlesztésére, a ZEUSZ-ra, melyet a NASA-nál is ismernek.

Tudását igyekszik minél szélesebb körben megosztani másokkal is. Ennek folyományaként több könyve is megjelent már a hazai könyvesboltokban az elmúlt években, nem egy közülük sikerlisták élére is került. Munkáiról bővebben a <http://feherkrisztian.atw.hu/> weboldalon is lehet olvasni.

2. A Direct2D madártávlatból

Könyvünk a Microsoft Direct2D grafikus programozási felületének használatához nyújt bevezetőt. Mielőtt belevágnánk fejszénket a téma gyakorlati részébe, először nem árt tisztában lenni azzal, hogy egyáltalán mi is az az eszköz, aminek a használatára készülünk és az hol helyezkedik el a grafikus megjelenítési technológiák világában.

Ennek a fejezetnek a végére fel fogjuk tudni ismerni, hogy saját, esetleges már meglévő grafikus programjaink hogyan profitálhatnak a Direct2D technológiából.

2.1. Mi az a Direct2D?

A programozás iránt érdeklődő olvasók előtt bizonyára ismerős a Microsoft DirectX technológiája. Ez a multimédiás programozási felület napjaink de facto eszközévé vált például a játékfejlesztések terén. Ennek oka a rendkívül nagy teljesítmény, mely a videokártyák feldolgozóegységeinek (magjainak) a munkába állításából is fakad.

A DirectX maga mára egy komplex eszközkészletté vált, mely nem csupán a háromdimenziós megjelenítésért felelős Direct3D komponenst tartalmazza, hanem például audio eszközök programozását, vagy a játékvezérlőekét is. A DirectX tehát lényegében egy API gyűjtemény.

Számunkra a grafika a lényeg, ezért DirectX-en belül a Direct3D és a Direct2D jön szóba. 2D és 3D külön-külön is programozható? Igen is és nem is. A Direct2D (könyvünk fókusza) ugyanis önálló programozási felületet kínál, mégis a Direct3D technológián alapul. Leegyszerűsítve, a Direct2D a Direct3D funkcióit hívja, egy közbeiktatott rétegen keresztül.

Szerencsére ez a közbeiktatás nem jelent kompromisszumot az elérhető teljesítmény tekintetében, ami számunkra nagyon jó hír. Sőt, eláruljuk, hogy a Direct2D bizonyos területeken, például a vonalraj-

zolás terén, túl is szárnyalja a Direct3D-t. Ebből azt kell észrevennünk, hogy a Direct2D nem szolgai módon csak a Direct3D funkcióira nyúl vissza, hanem valódi hozzáférése is van a grafikus hardver lehetőségeihez. Ez egy újabb jó hír.

A Direct2D úgy lett tervezve, hogy rajta keresztül használható legyen a Microsoft DirectWrite API-ja, ami professzionális minőségű szöveges megjelenítést tesz lehetővé.

Viszont joggal kérdezhetjük: mi szükség Direct2D-re, ha lényegében ugyanazt csinálja, mint a Direct3D? Ennek oka elsősorban az egyszerűségben és az átjárhatóságban van.

1. A Direct2D sokkal egyszerűbben programozható, mint a Direct3D. Kevesebb kódsorral kell megküzdenünk, gyorsabban tanulható, jobban átlátható. Aki programozott a GDI, vagy GDI+ API-k valamelyikével, hasonló logikával, függvénynevekkel fog a Direct2D esetében is találkozni.
2. Mivel nincs szükségünk a teljes 3D "fegyverarzenálra", amit a Direct3D kínál, így "felesleges" dolgokkal sem kell kínozni agytekervényeinket.
3. A Direct2D a GDI és a GDI+ irányába is kínál átjárhatóságot. Ez azt jelenti, hogy képes például GDI eszközkontextusra is rajzolni. Mire jó ez? Amennyiben valaki GDI, vagy GDI+ alapú grafikus megjelenítő rendszerét GPU alapokra akarja helyezni, azt több lépcsőben, átmenetes formában is megteheti. Megjegyezzük, hogy a Direct2D a Direct3D irányába is kínál átjárási lehetőséget. Ennek bemutatására könyvünk végén külön alfejezetet szenteltünk.

A Direct2D első verziója 2007-ben jelent meg, a Windows 7 keretein belül. A Windows 8 óta pedig elérhető az 1.1-es verziója is, mely jelentős újdonságokat is behozott a "motorháztető alá".

2.2. Miért Direct2D?

Asztali számítógépes környezetben a grafikus alkalmazásmegjelenítés jövője a GPU alapon gyorsított megjelenítés. Microsoft Windows környezetben ez elsősorban a DirectX-et jelenti, a gyakorlatban pedig a Direct3D-t.

A modern Direct3D programkód viszont meglehetősen bonyolult és kezdők számára ijesztő, hiszen manapság már az ún. shaderprogramozás, a GPU-k alacsony szintű kezelése sem kerülhető el, ami csak tovább bonyolítja a dolgokat.

Nem igazán éri meg egy átlagos felhasználói program elkészítésekor, ahol csak például egyszerű kétdimenziós megjelenítésre van szükség (néhány egyszerű geometriai alakzat, esetleg szöveg kirajzolására), hogy a programozó bevesse a Direct3D arzenálját egy alapvetően egyszerű feladat elvégzéséhez.

Igény viszont van a gyors és jó minőségű kétdimenziós rajzolás programozhatóságára is. Pontosan ez az, amire a Direct2D tökéletes megoldás lehet.

2.3. Mit csinál a Direct2D?

A Direct2D elsődlegesen a renderelésre összpontosít, tehát egy renderelő eszközként tekinthetünk rá. Kiemelten fontos továbbá, hogy együtt tud működni más Windows API-val is, mint például a GDI, vagy teljesen banális módon magával a Direct3D-vel is.

Ennek ellenére rendelkezik egy nagyon jól átgondolt alap rajzoló függvénygyűjteménnyel is, mely szinte teljesen a GPU bevonásával működik. Ha például vonalakat rajzolunk, a háttérben a vonalak háromszögekre lesznek lebontva és teljesen a GPU segítségével történik a renderelés is, anélkül, hogy külön kódot kellene írunk erre. Jól hangzik, nemde?

Összefoglalásképpen elmondhatjuk: a Direct2D használata akkor jó, ha modern eszközkészlettel akarunk nagy teljesítménnyel rajzolni, elsősorban kétdimenzióban, de nem akarunk nagy kódbázist alkalmazni erre.

2.4. Mire használhatjuk a Direct2D-t?

Jóformán bármire: képfeldolgozásra, grafikonok megjelenítését végző alkalmazásokhoz, 2D játékok készítéséhez, stb.

Sőt, ha rendelkezünk szoftveres 3D megjelenítő motorral, akkor annak teljesítményét is "felturbózzhatjuk" a Direct2D segítségével. Könyvünkben azt is meg fogjuk ismerni, hogy hogyan alkalmazhatunk egy ilyen hibrid 3D megjelenítő motort is. De ne siessünk ennyire előre, kezdjük az elején!

2.5. Milyen gyors a Direct2D?

Igen, ez egy jogos kérdés. Rögtön az elején eláruljuk a választ: a Direct2D megközelíti az OpenGL és a Direct3D renderelési sebességét.

Egy igen jó számítógéppel készítettünk egy nem reprezentatív teljesítménytesztet különböző grafikus API-k tekintetében, ami nem száz százalékosan pontos méréseken alapul, de tájékozódni lehet a segítségével.

	100.000 Point	100.000 Polyline	100.000 Poligon
GDI	63	296	1592
GDI+ aliased	1529	17285	17144
GDI+ antialiased	1560	28938	28813
Direct2D aliased	140	63	297
Direct2D antialiased	234	141	62
OpenGL aliased	281	281	172
OpenGL antialiased	296	296	172
HTML5 Canvas	584	782	894
Flash (desktop)	102	207	290

Tesztgép:

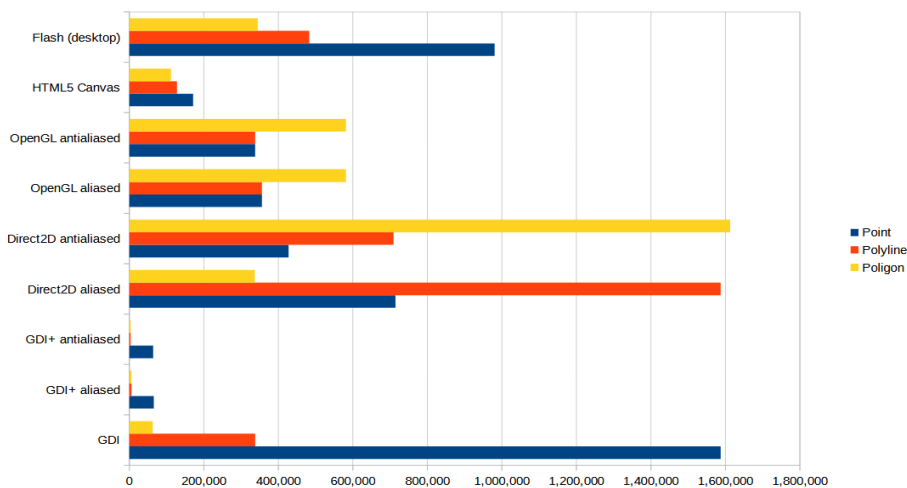
CPU: Intel Core i5 – 4 x 3.3GHz

RAM: 16 GB RAM

GPU: Quadro 600

A teszt során 100.000 pontot, vonalat és poligont rajzoltunk ki a képernyőre és megmértük az ezek rendereléséhez szükséges időt.

Grafikonnal ábrázolva még jobban látszik, hogy a Direct2D-nek igenis van keresnivalója a grafikus API-k között.

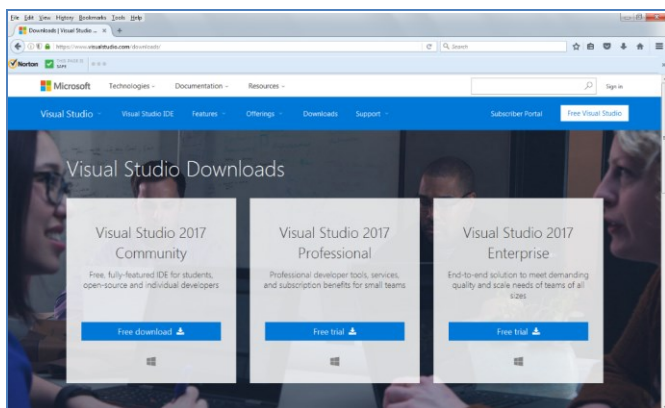


3. A Direct2D és a Visual Studio

A Visual Studio a Microsoft hivatalos fejlesztőeszköze. Professionál kiadása fizetős, ám egy ideje létezik nyílt forrású verziója is, a 'Microsoft Visual Studio Community 2017' ami teljesen ingyenes, mindössze egy ingyenes Microsoft fiókot kell megadnunk a korlátlan idejű használatához. A Community Edition a korábbi Express Edition felváltásának is tekinthető.

Az alábbi webhelyről tölthető le:

<https://www.visualstudio.com/downloads/>



A 'Free download' linkre kattintva először is le kell töltenünk egy online telepítőalkalmazást, melyet elindítva kezdhető meg a tényleges telepítés.

Először el kell fogadnunk a Microsoft liszensz feltételeit.

