

Szakmai kérdések

Írta: konzolcowboy - 2009.04.21 14:30

Ennek a témának az a célja, hogy szakmai kérdéseket tegyünk fel itt Nyisztor Károly úrnak. Itt feltehetünk minden olyan kérdést, amelyet a könyvek elolvasása ellenére sem értünk, vagy magyarázatra szorulnak.

=====

Vá: Szakmai kérdések

Írta: konzolcowboy - 2009.04.21 15:32

Nekem rögtön lenne egy kérdésem, a kórdinátákkal kapcsolatban.

A 2D-s programozás során tudtam azt hogy mit jelent az x, és y kórdináta.(x pixel a képernyő bal szélétől, y pixel a képernyő tetejétől.)

A direct3d azonban már vektorokkal dolgozik. Tudom hogy a következő sor létrehoz egy helyvektort egy 3D-s kórdináta rendszerben:

```
D3DXVECTOR3( -0.5f, 0.0f, 1.0f )
```

De mit jelent pl. a -0.5f?

Azt tudom, hogy negatív irányban kell az x tengelyen haladni -0.5 (valamilyen egységet), de pontosan milyen egységet takar? Pixel? mm? És hol van ez az x tengely? A képernyő melyik részén jön létre a kordináta rendszer?

Csak azért merült fel bennem ez a kérdés, mert a DXAlap nevű program megjelenít egy háromszöget a képernyőn. Én átírtam a háromszög pontjainak a kórdinátáit, de nem jelent meg a háromszög, ezért visszaírtam az eredeti értékeket, és úgy már ismét működött, és nem értem, hogy miért csak az eredeti értékekkel működik?

A világekordináta fogalmával is van némi gondom.

Például van egy olyan kórdináta rendszerem egy kockás lapon, amelyben minden egység egy kockának felel meg, és abban elkészíték egy háromszöget. Majd a háromszög pontjainak az xyz kórdinátáit (amelyek a lapon levő kórdináta rendszerben vannak értelmezve) beviszem egy direct3d programba. Ha a programban elvégzem a világ nézet transzformációkat, akkor a háromszög meg fog jelenni a képernyőn úgy, ahogy a kockás lapon én megrajzoltam?

Előre is köszönöm a válaszokat, és elnézést kérek a buta kérdésekért :blush:

=====

Vá: Szakmai kérdések

Írta: Swinkx - 2009.04.21 20:44

Megpróbálok én válaszolni remélem nem gond:)

Nah szóval koordináta rendszer.Ezt így elég nehéz elmagyarázni, inkább megmutatom.

http://www.tankonyvtar.hu/site/upload/2008/10/images_page_99.1.png

Az ábra ne érdekeljen, nézd csak a nyilakban végződő x,y,z-t. Ezeket hívjuk tengelyeknek, ahogy láthatod mind1ik merőleges a másik 2 által meghatározott síkra.

Ez mondjuk pont 1 jobbsodrásu rendszer, a d3d alpból balsodrásuval dolgozik, ezekről leírás a shaderkönyvben a 14, a másikban a 33. oldalon találsz leírást:)(még annyi, hogy a d3d ennek a koordináta rendszernek az elforgatását használja, szal ha szemből nézed, vízszintesen van az X tengely és jobbra növekszik, függőlegesen a y az felfele növekszik, és a z pedig pontosan befele néz a monitor közepébe és arra is növekszik.)

Itt szerintem nincs semmi konkrét mértékegység, ez egy viszonyszám.

Valószínűleg azért nem láttad, mert túl agy koordinátákat adtál meg neki, és a 3szög kilógott a képből.

Inkább azt mondanám, hogy hasonló lesz a 2 3szög.

Remélem segitettem, de majd jön Carlos, és elmagyarázza neked érthetően:D

=====

Vá: Szakmai kérdések

Írta: admin - 2009.04.24 11:23

Kösz, Swinkx! :-)

=====

Vá: Szakmai kérdések

Írta: Quantum - 2009.04.26 21:09

Sziasztok!

Nem rég kezdtem el a dx-et és mivel nagyon sokat foglalkoztam/foglalkozok grafikával (inkább 3D-vel), dimenziók szerint haladok. Tehát most még a sprite-oknál tartok.

Végig néztem a könyvet és hiányolom belőle az egérkezelést, de inkább úgy fogalmaznék, hogy nem találtam benne. Kutakodtam a neten, és találtam egy dx7-es magyar leírást, ami azt írja, hogy a directinput egere csak az elmozdulást és a sebességet figyeli, de pontos helyzetet nem tudunk lekérdezni (ami nekem kicsit fura, mivel hozzá vagyok szokva a C-n belüli bgi könyvtárhoz). Nekem viszont szükségem lenne rá, mivel menüt szeretnék készíteni.

Szóval a kérdés: hogyan lehet dx-ben menüt készíteni? Vagy erre van egy nem dx megoldás?

=====

Vá: Szakmai kérdések

Írta: admin - 2009.04.26 23:19

Szia Quantum!

Sajnos nem tudom, melyik könyvről beszélsz. Az előző könyvemben is foglalkoztam a DirectInput-tal, a

mostaniban pedig még részletesebben foglalkozom.

Javaslom, nézd át a 11.1.6. "Beviteli eszközök - a billentyűzet és az egér kezelése" című fejezetet a 287. oldalon, a Shaderprogramozás könyvben.

A DX7-es dokumentációt pedig csak akkor tanulmányozd, ha történész vagy, vagy nosztalgiázni akarsz. :)

A másik félreértés esetedben az egérkezelés C-ben. A C és a C++ platformfüggetlen nyelvek, épp ezért az egérkezeléssel nem foglalkoznak - azonban valóban léteznek oprendszer függő függvénykönyvtárak.

Az természetes, hogy nem az egér pontos helyzetét, hanem a görgők relatív elmozdulását adja vissza a DInput API. A fejlesztőn múlik, hogyan interpretálja az elmozdulást, a felbontást-, és egyéb beállításokat figyelembe véve. Nem véletlen az sem, hogy a legtöbb játékban beállítható az egér érzékenysége...

Az utolsó félreértés a menü készítés és az egérkezelés keverése. Menüt többféleképpen készíthetsz, a legegyszerűbb módszer az, ha téglalap alakú területeket definiálsz, és az adott téglalapon belül megfelelően kezeled az egér eseményeket. Ennyi. Persze animálhatod a menüt, lehet 3D, beúszhat balról, vagy éppenséggel viccesen pattoghat, ha az ember rááll az egérrel. A lehetőségek száma majdhogynem korlátlan itt is, de a szükséges eszközöket a DX biztosítja (2D grafika, DInput).

Azonban a DirectX és a C++ alapú fejlesztés nem ugyanaz, mintha valaki Flash-ben összedob egy menüt.

Utóbbi egy gyors eszköz vizuális megoldásokra, a C++ meg a DirectX pedig valósídejű, 3D-PROGRAMOZÁSHOZ(!) kell.

Tanács: A neten számos, nyílt forráskódú motor fellelhető (pl. Irrlicht, Neoengine), amelyekben kész megoldásokat találhatsz, akár menükezelésre is.

Azonban józan paraszti ésszel is belátható, hogy alapjában véve ortografikus vetítésre (sprite-ok), valamint egérkezelésre van szükség a menü megjelenítésére és a menükezelésre.

Üdv,
Carlos

Vá: Szakmai kérdések

Írta: Quantum - 2009.04.27 10:44

Heló Carlos!

Köszönöm válaszod. Látom, nem voltam egyértelmű, ezen most javítok. :P

A Grafika és Játékprogramozás DirectX-szel c. könyvedről beszéltem, a Shader-est még nem tudtam beszerezni. DX7 meg nem érdekel, bár szeretem a régmúlt történelmet. :D

Amit a C-vel és a Cpp-vel kapcs. írtál, azokkal tisztában vagyok. Több grafikai könyvtár, header létezik ezen nyelvekhez, a legtöbben az egérkezelés az egér valódi koordinátaival történik. Egy ilyen könyvtárat tanultam (egyetemen ez volt az elvárás). Ha érdekel, belinkelem érdekesség képpen:

http://www.inf.unideb.hu/grafika/schwarz_hallg.htm

Tehát akkor, mint írtad, a DX adta lehetőségekből kell gazdálkodjak.

"a C++ meg a DirectX pedig valósídejű, 3D-PROGRAMOZÁSHOZ(!) kell." Ezzel tisztában vagyok. :)

Tehát akkor sprite és DI.

Sejtettem, hogy ebbe az irányba kell elmozduljak, de a kérdésem leginkább az egér miatt merült fel.

Köszí szépen a választ!

=====

Vá: Szakmai kérdések

Írta: admin - 2009.04.27 11:16

Helló Quantum!

Részemről is félreértés történt, mivel első üzeneted alapján arra gondoltam, hogy olyan környezetben, illetve olyan fejlesztőeszközökkel dolgoztál, ahol például a menükezelés is adott. Sajnos itt mindezt magadnak kell felépíteni, azonban óriási előnye a dolognak, hogy gyakorlatilag nincs megkötés.

Egy ilyen könyvtárat tanultam (egyetemen ez volt az elvárás). Ha érdekel, belinkelem érdekesség képpen: www.inf.unideb.hu/grafika/schwarcz_hallg.htm

Hú, a régi jó BGI és a VGA üzemmódok programozása! :-) Anno én is próbálkoztam vele, azonban gyorsan leváltottam saját, alacsony szintű (Assembly) kódra. Van is pár demóm, amelyek az időközben megszűnt portálon fenn voltak, majd újra feltöltöm őket, érdekességgként.

Remélem ma már nem erőltetik az egyetemen a bgi könyvtárat (max. érdekességgként említeném meg széljegyzetben).

=====

Vá: Szakmai kérdések

Írta: Quantum - 2009.04.27 11:29

Nem erőltették, de az óra anyaga ez alapján volt. Vagányságból megkérdeztem, hogy dx-ben be lehet-e adni a házikat :D, azt mondták lehetséges és el is fogadják. De akkor még csak tanultam az oop-t, nem mertem belekezdeni DirectX-be, így maradtam a C-nél. :)

Valamire nagyon jó volt: megtanultuk a 2D-, a 3D- és a vektorgrafika alapjait. Pl. nem tudom, hogy van-e egyszerűsített megoldás approximációs és interpolációs görbék létrehozására dx-ben, de ha nincs, könnyebb dolgom van. :P

Ha megtalálom, én is feltöltöm a demóimat (beadandóimat).

=====

Vá: Szakmai kérdések

Írta: konzolcowboy - 2009.04.27 11:42

Carlos megfogadtam a tanácsodat, és egyszerűbb projecttel kezdem a játék fejlesztést. Egy 2D-s falbontó játékot készítek első körben. Elkészítettem a koncepciót, és a grafikát, viszont egy valamire nem tudok megoldást találni. Ez pedig a labda csavarása az ütővel.

Nincs valami jó ötleted, hogy honnan kellene elindulnom?

Előre is köszönöm a választ.

=====

Vá: Szakmai kérdések

Írta: admin - 2009.04.27 13:44

A DX lehetőséget ad szakaszok és különálló pontok megjelenítésére, amelyeket tetszőleges algoritmusok alapján állítasz elő.

Ajánlom figyelmedbe továbbá a D3DX D3DXVec3CatmullRom() segédfüggvényét, amely paraméteres ívek számíthatók ki.

=====

Vá: Szakmai kérdések

Írta: admin - 2009.04.27 14:47

konzolcowboy írta::

Ez pedig a labda csavarása az ütővel. Nincs valami jó ötleted, hogy honnan kellene elindulnom?

Egy egyszerű koncepciót vázolok fel.

A labda visszaverődési szöge a beesési szög és az ütő sebességének függvénye. Ha az ütő épp áll, akkor a labda ugyanolyan szögben hagyja el az ütőt, mint amivel becsapódott. Persze ez unalmas lenne, ezért érdemes megfűszerezni egy csipetnyi random értékkel. A lényeg: nem kell túlbonyolítani, itt úgysem a valós fizikán van a hangsúly.

=====

Vá: Szakmai kérdések

Írta: axen - 2009.06.08 07:37

Hello!

Nekem egy kezdő kérdésem lenne. Most járok az 1. könyvben az index puffernél és megjegyzésként le is van írva, hogy a triangle strip-el is ugyanúgy meg lehet spórolni a vertexeket. Ezek szerint, ha jól értem, akkor vagy index puffer, vagy triangle strip, a kettőnek együtt nincs értelme? :P

=====

Vá: Szakmai kérdések

Írta: admin - 2009.06.08 16:24

Szia!

Úgy gondoltam, hogy a téma megérdemel egy cikket:

http://www.nyisztorkaroly.org/index.php?option=com_content&view=article&id=65:3d-optimalizalas-trian

glelist-trianglestrip-index-puffer&catid=36:programozas-tervezes&Itemid=60

Remélem ez megválaszolja a kérdésetet! Jó felvetés volt, köszönöm!

Üdv,
Carlos

Vá: Szakmai kérdések

Írta: axen - 2009.06.08 16:56

Hú, köszi, így már világosabb. :)

Vá: Szakmai kérdések

Írta: YAQ - 2009.07.17 15:34

Üdv!

Rég jártam már erre (az előző fórumodban Dave névvel futottam)

Nemrég megvettem a könyvedet és nagyon tetszik (főként a kissé gyakorlatiasabbra fogott elméleti alapozás). Még nem volt időm teljesen végig olvasni, de már nem sok hiányzik. Tehát mindent egybevetve gratulálok hozzá.

Közben sajnos le kellett állnom egy időre a saját játékom fejlesztésével (volt pár évtizedes matek lamaradásom az egyetemen, és pótolni kellett, a végére egész jól összejött:silly:), de mostmár gördülékenyebben fognak menni a dolgok.

Jah, itt tennék egy kis kitérőt: Quantum: én is a DE-re járok, szeptembertől lesz bevgafunk, ajánlhatnál valami jó gyakvezetőt:).

Az első szakmai kérdésem pedig talán inkább az FX Composerrel kapcsolatos: vannak a shaderkódok elején ilyen, és ehhez hasonló sorok:
uniform extern float4x4 gWorldViewProjection;
uniform extern float4 gLightColor;

Nah most ezeket értem is, de ha az effektet az FX Composeren belül rádobom egy anyagra, akkor semmi sem látszik. Arra nem volt nehéz rájönnöm, hogy itt nem kap a mátrix a programból értéket, és azért nem. Majd nagynehezen kilestem egy másik programból, hogy ha ezt írom: uniform extern float4x4 gWorldViewProjection :WorldViewProjection;, akkor a program belepakolja a mátrixba a szükséges adatokat, és már látszik a modell a képen (fogor, mozog, minden)

Viszont a többi változónak nem tudom a program beli megfeleltetését:(
tehát erre a célra kellene nekem valami lista, vagy forrás.
(bevallom, az fx composer teljes súgóját még nem volt időm átolvasni, de ha esetleg valakinek már van listája, akkor sok időt spórolna nekem a belinkeléssel)

Probléma megoldva. (ctrl+j)

=====

Vá: Szakmai kérdések

Írta: Swinkx - 2009.07.19 21:06

Szia,
a problémám ugyanaz, tudnál segíteni?

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.07.20 10:14

Swinkx: Én a 2.5-ös verziót használom, ott egyszerűen a kód egy üres részére kattintasz, aztán ctrl+j, erre feljön egy kis submenu "choose snippet" felíráttal, ott pl nálam a legutolsó a "World View Projection", ez az alábbi változókat szúrja be:

```
float4x4 World : WORLD;  
float4x4 View : VIEW;  
float4x4 Projection : PROJECTION;
```

```
float4x4 WVP : WORLDVIEWPROJECTION;  
float4x4 WorldInverse : WORLDINVERSE;
```

ezek már maguktól kapják az értékeket.

Nekem meg az lenne a kérdésem, hogy a futóváltozók értékeit (úgy a shader kód futása közben) ki lehet valahoggy írni? (esetleg fájlba)

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.07.20 18:48

és még egy kérdés:

van itt egy ilyen:

```
#if defined(DEBUG) | defined(_DEBUG)  
#pragma comment( lib, "d3dx9d.lib" )  
#else  
#pragma comment( lib, "d3dx9.lib" )  
#endif
```

a kérdés pedig az, hogy hogyan lehet ilyet a 32/64 bites fordításra definiálni?

=====

Vá: Szakmai kérdések

Írta: axen - 2009.07.20 19:14

Szerintem így:

VS-ben fent release/debug combóban -> configuration manager -> solution platform new -> platform x64.

Aztán a lib elérése elvileg már ott van a VS-ben (a Directx SDK beállítja neked). Ha mégse lenne meg, akkor beállítod a szokásos módon: tools -> options -> projects and solutions -> VC++ directories -> platform x64/library files ->

Microsoft DirectX SDK (March 2009)Libx64

És akkor elvileg csak fent kapcsolagatod. A libek neve meg ugyanaz.

szerk.: Ez csak a professional és a feletti edition-okban működik, express-hez nincs 64 bites C++ fordító tudtommal. ;)

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.07.20 23:04

kösz az ötletet, igazából nem lib-hez kellett, de megjegyzem (később még jól jöhet).

Én a sűgóban ezt találtam:

```
#if defined (_WIN64)
#define INT_SIGN_BIT 0x8000000000000000
#else
#define INT_SIGN_BIT 0x80000000
#endif
```

(tulajdonképpen egyedi definíciókhoz kell, de közben ezeket is lerövidítettem)

Jah, közben felmerült egy újabb probléma.

Nézegettem a programomat, és rádöbentem, hogy több helyen is át kell írni. (még a fixed pipelinet-t használta)

Az ezzel kapcsolatos kérdésem az lenne, hogy vannak ilyen 3ds-max ban elkészített animált modelleim, amit a pandasofttal exportáltam ki directx-be. Még a könyvsorozat 1. részében le volt írva, hogy hogyan lehet ezeket az animált dolgokat programban használni. Viszont nem tudom, hogy létezik-e annál egyszerűbb technika, vagy csak az van?

Jah, és továbbra sem tudom, hogy egy shader kódban lévő változó értékét hogyan lehet fájlba v akárhova kiíratni futás közben:(

Vá: Szakmai kérdések

Írta: admin - 2009.07.27 10:03

>> egy shader kódban lévő változó értékét hogyan lehet fájlba v akárhova kiíratni futás közben:(
Erre a célra ajánlom a shader debugger-t. A PIX for Windows használatáról találsz útmutatást a könyvben is.

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.07.31 12:49

Debuggerekkel nem vagyok kibékülve, de akkor majd ezzel összeismerkedek.

Lenne még C++ os kérdésem.

van pl egy ilyenem:

```
while (i
```

=====

Vá: Szakmai kérdések

Írta: admin - 2009.07.31 17:53

A postfix operátorok kénytelenek létrehozni másolatot, ugyanis nem az eredeti értéket módosítják azonnal. A postfix operátorok egy ideiglenes példányban letárolják a bejövő paraméter értékét, növelik az értéket, majd az ideiglenes értéket adják vissza.

Valahogy így képzeld el a dolgot:

```
int postfix_increment( int& i )  
{  
    int temp = i;  
    ++i;  
    return temp;  
}
```

Ami megmagyarázza a következőt:

```
int i = 1;  
cout
```

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.07.31 22:13

Oké, ez megnyugtató, és megmagyarázza az add edx,1 -et is.

Tudom, hogy nem ez fogja lassítani, csak egyrészt kíváncsi voltam, hogy hogyan lehetne rábírni az operátort ideiglenesen másféle működésre, másrészt meg már sokszor szívtam azon, hogy utólag akartam optimalizálni valamit. Általában sokáig tart rájönnöm, hogy a régi kódjaim mit csinálnak (különösen az éjszakai verziók), kommentet meg nem szeretek írni. (egyrészt lustaság, másrészt amikor írtam, akkor meg a kommentet nem értettem)

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.08.10 18:00

Van egy aprócsaka effektbetöltési problémám, aminek nem tudom, hogy járhatnék a végére.

itt egy kis részlet:

```
char effectFile;
...
ID3DXBuffer *error;
HRESULT hr;
ID3DXEffect *effect;
hr=D3DXCreateEffectFromFile(myD3DDevPtr,effectFile,0,0,0,0,&effect,&error);
```

erre a dx hibakereső a hr alapján ezt feleli: "undetermined error occurred"

a debug szerint az effectfile helyes, az effect mutató null, és az error is null

Szerintem mi lehet a probléma?

=====

Vá: Szakmai kérdések

Írta: admin - 2009.08.12 14:28

Elképzelhető, hogy nem itt van a probléma, hanem egy korábbi hiba miatt hasal el az effekt betöltés. Debug-old végig az előző részeket is, ne csak ezt a helyet ellenőrizd. Pl. érvényes a device pointer?

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.08.13 10:10

Közben kiderült a bibi. A notebookon akartam folytatni a keresést, erre ott minden probléma nélkül futott. A gépen Win7 van, gondolom ez kavarhatott be (legalábbis remélem, különben én cseszhettem el valahol)

Azt ugye meg lehet csinálni, hogy van egy tömböm az alkalmazásban, és átadni az egészet a vertexáryalós résznek?

Úgy gondoltam, h valahogy egy textúrát kellene csinálni a tömbből a memóriába, és aztán lehetne mintavételezni a texturehandler segítségével.

Szerinted mi a legegyszerűbb megoldás? (esetleg simán a vramban nem lehetne létrehozni a tömböt, és a címét átadni az árnyalónak)

=====

Vá: Szakmai kérdések

Írta: admin - 2009.08.13 12:45

A leírás alapján neked valószínűleg a 'render-to-texture' avagy 'offscreen rendering' technikára lenne inkább szükséged, amit nem taglaltam a könyvben. DirectX-ben nem nehéz megvalósítani, ugyanakkor haladóbb technikának számít. Némi támpont:

http://www.codesampler.com/dx9src/dx9src_7.htm#dx9_offscreen_rendering

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.08.13 13:26

Igen, hasonló,

A történet az lenne, hogy több szigetet kellene kirajzolni (max 49), és nem akarom terhelni a memóriát, ezért csak 1-et töltök be 3D-s modellként, ami viszont kerek (a szigetek széle meg ugyebár kissé szabálytalan), ezért miután kiderült, hogy hány db szigetet kell valójában kirajzolni, mindegyikhez hozzárendelek egy-egy tömböt, amiben a megfelelő vertexek véletlengenerátorral előállított eltolási értékei lennének (x;z). Ezeket az adatokat kellene eljuttatni a vertex shaderhez, ami aztán eltologatná a vertexeket.

(mivel sok adatról van szó (max 49*24*2*4 B), amiket szakaszosan kellene átadni (szigetenként 1 tömb), ezért gondoltam a textúrában tárolt adatokra, mert az amúgy is vektorosan tárolja a színértékeket, és 1 textúrával le lehetne fedni az összes szigetet, így adatot másolni is csak egyszer kellene. De ez még odébb van, azért megpróbálom valahogy sima tömbbel is ha lehet.)

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.08.15 14:43

Inkább egy vertexpufferben adom át, egy másik csatornán keresztül, az egyszerűbb... (remélem lehetséges)

Idevágó kérdésem az lenne, hogy a SetStreamSource (0,VBPtr,0,0); hívás után (a sűgó szerint az utolsó 0 miatt nem lesz automatikus léptetés a pufferben) hogyan lehet a vertex shaderen belül a puffer egy adott pontjára ugrani?

Az áttetszőséggel kapcsolatban lenne még egy kérdésem:

Írtad a könyvedben, hogy ha több modell van, akkor először a tömöreket érdemes megjeleníteni, majd az áttetszőeket olyan sorrendben, hogy elsőnek a kamerától legtávolabbit, és utolsónak a legközelebbit. Mi a helyzet akkor, ha ezt nem lehet megvalósítani, mert pl az egyik áttetsző test konkáv, és közte van a másik áttetsző test? (mintha 1 mintás pohárba tennék egy másikat)

=====

Vá: Szakmai kérdések

Írta: admin - 2009.08.19 17:36

Gyorsan mert rohanok: az első kérdésre hirtelen azt tudnám válaszolni, hogy sehogy. Nem arra találták ki a shader-t, hogy tömböket dolgozzon fel (hiszen minden egyes vertexre, és ami még rosszabb, minden pixelre lefut!).

Az átlátszóság kapcsán: a pre-render queue sorrend valóban csak áttetsző, konvex textek esetén okoz teljesítmény növelést. A poharas példában a Z-buffer-t valóban nem érdemes kikapcsolni, mindössze a kirajzolási sorrendre hagyatkozva, hiszen akkor rossz eredményt kapunk.

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.08.25 09:07

Hát ezzel szívtam egy kicsit. Megírtam textúrára, erre közölte velem, hogy vertexshaderben nem használható a tex2D utasítás, végül kisebb tömbökkel sikerült megoldani.

Most az lenne a kérdésem, hogy hogyan lehet több effektet keverni, tehát pl ha az egész képernyőt kicsit zöldesre szeretném színezn, mintha egy napszemüvegen keresztül nézném, akkor annak mi a bevett technikája?

=====

Vá: Szakmai kérdések

Írta: admin - 2009.08.25 10:55

Megírtam textúrára, erre közölte velem, hogy vertexshaderben nem használható a tex2D utasítás

Ez tény, a vertex shader a csúcspontokkal dolgozik. (Valójában SM 3.0-tól a vertex shader-ből is elérhetők a textúra adatai.)

Most az lenne a kérdésem, hogy hogyan lehet több effektet keverni, tehát pl ha az egész képernyőt kicsit zöldesre szeretném színezn...

Megteheted, hogy több pass-t definiálsz a shaderben, erről írtam is a könyvben. Viszont ahhoz, hogy kiemeld a zöld színt, erre nincs szükség. A pixel shaderben igény szerint módosítható a szín, azaz esetekben megnövelheted a zöld komponenst. Persze kísérletezni kell, hogy a megfelelő hatást eléred (tanácsos pl. RenderMonkey-ban tesztelgetni).

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.08.25 17:49

Erre én is gondoltam, de több effektem van a több modell miatt. A zöldítést meg nem csak a modell pixeljeire gondoltam, hanem mindenre (háttérre is). Vagy egy effekt pixel shadere lefut a már kirajzolt, független modellekre is?

=====

Vá: Szakmai kérdések

Írta: admin - 2009.08.26 13:09

Ez esetben elegánsabb lenne egy post-screen shaderrel megoldani a dolgot. A lényeg, hogy a jelenetet egy offscreen textúrába rendereled ki, majd ezen futatod le a "zöldítő" pixel shadert. A fogalomra ráuglizva számos jó leírást találhatsz.

Íme néhány:

<http://blogs.msdn.com/coding4fun/archive/2006/11/06/997852.aspx>

http://www.gamedev.net/community/forums/topic.asp?topic_id=488779

Az első managed példa, de a lényeg ugyanaz.

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.08.29 12:30

Üdv ismét!

Egy kis magyarázat kellene a könyv 111. oldalához. A projekciós transzformációban a Z koordináta leképezése nem teljesen világos. A képletekben az u és a v mit jelent?

=====

Vá: Szakmai kérdések

Írta: admin - 2009.09.05 16:03

Az u és a v a két ismeretlen az egyenletben (tehát nem textúra koordináták, csak kissé szerencsétlen volt a névválasztás), amelyekre azért volt szükség, hogy a transzformációt mátrixos alakban is kifejezhessük. A többi gondolom világos, elég részletes a levezetés.

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.09.11 13:46

Oké, mostmár világos

Most az lenne a kérdésem, hogy a könyvben lévő (121. o. 5.33 vagy 87.o. 5.1) ábrák, és az azokhoz hasonlóakat magad rajzoltad, vagy van valami hiteles webhely, ahonnan lehet letölteni ilyeneket?

Abban az esetben, ha saját gyártmányod, megkérdezném, hogy felhasználhatok-e párat (főként a blokkvázlatokat) egy powerp. bemutatóhoz?

=====

Vá: Szakmai kérdések

Írta: admin - 2009.09.11 16:18

A könyvben minden ábra saját. Felhasználhatod, de valamilyen formában utalj a forrásra. Kösz, hogy rákérdeztél.

=====

Vá: Szakmai kérdések

Írta: YAQ - 2009.09.12 09:44

Meglesz :)

Van itt még valami, amivel nem vagyok tisztában a működést illetően. Azt tudom, hogy a csúcspont árnyaló minden vertexre lefut (ill gondolom azokra, Begin () - End () között rajzolódnak ki) és azt is tudom, hogy a képpontárnyaló meg minden képpontra lefut, ami a geometriai szakasz kimenete. Amit nem tudok, az az, hogy pontosan mikor?

Ha van pl egy négyzetem (vagy komplexebb alakzat) akkor kirajzoláskor a vertex shader először 3 csúcspontra fut le, majd az interpolált kimenetére rögtön lefut a pixel shader, aztán ugyanez lejátszódik a következő háromszögre; vagy lefut a vertexshader az összes kirajzolandó háromszögre, aztán annak kimenetére a geometriai szakasz (közben az összes interpolált eredmény valahol tárolódik, és csak ezután jön a pixelshader?

Tehát nekem a pontos mmechanizmus nem világos. Esetleg valami harmadik megoldás a valóság?

=====

Vá: Szakmai kérdések

Írta: admin - 2009.09.12 13:17

Az tehát egyértelmű, hogy a vertex shader csúcspontokat dolgoz fel, a pixel shader pedig interpolált értékek alapján számolja ki a végleges képpontok színét.

A kérdésedre azért egy harmadik alternatíva a válasz, mert a GPU-ban párhuzamosították a vertex-,

raszterizáció- és pixel műveleteket (több vertex, illetve pixel feldolgozó egység létezik a 3D chip-ben, lásd "3D-történelem" című fejezet, 3. oldal). A párhuzamos feldolgozás jó hatással van a teljesítményre, ugyanakkor előreláthatatlanná teszi a fejlesztő számára, hogy pontosan mi is történik a kis háromszögeivel, ha már egyszer átadta őket a GPU-nak.

Míg a geometria néhány háromszögének csúcspontjait még a vertex shader darálja éppen, addig mások már a raszterizációban vesznek részt, mindeközben a pixel shader pedig ontja a képpontokat, a már korábban interpolált értékek alapján.

Ennek egy mellékhatása az is, hogy nem ajánlott lekérdezni a már "leküldött" vertexek színét, stb., mivel ez igencsak megakasztja a végrehajtást, komoly lassulást okozva.

Vá: Szakmai kérdések

Írta: YAQ - 2009.09.12 19:42

Jó kis módszer...:D

Közben felütötte a fejét egy másik katasztrófa.

Egyetemen Javaban kellene programozni, amit visual studioban nem igazán lehet (legalábbis én nem tudok).

Szétnéztem a választható programok között (eclipse és társai), de egyik sem nyerte meg igazán a tetszésem.

Van valami módszer arra, hogy vs-ben is lehessen java progit írni?

Ha nincs, akkor tudnál valami jó és könnyen kezelhető IDE-t ajánlani, ami windows-os?

Vá: Szakmai kérdések

Írta: admin - 2009.09.14 14:01

Érdeemes kipróbálni, bár debugolni nem tudsz majd:

<http://www.improve.dk/blog/2007/09/29/compiling-java-in-visual-studio>

Eclipse helyett pedig:

<http://www.netbeans.org/>

Gyakorlati C++ példaprogramjai

Írta: KingBela - 2009.10.22 18:04

Tisztelettel érdeklődöm, hogy a Gyakorlati C++ könyv példaprogramjai honnan, milyen linkről tölthetők le?

Üdv: Béla

=====

Vá: Gyakorlati C++ példaprogramjai

Írta: admin - 2009.10.26 00:55

Feltöltve - link a könyvajánló cikk oldalán:

<http://www.nyisztorkaroly.org/szakkonyv-ajanlo/gyakorlati-c.html>

Üdv,
Károly

=====

kapcsolat

Írta: fodormik - 2009.12.13 19:43

Kedves Károly,

Bocs, hogy ide írok de nem találtam sehol "kapcsolat" linket (lehet, hogy az én hibám). Egy játék miatt kereslek, amibe tanácsot szeretnék tőled kérni. Kérlek írd a miklos.fodor@gmail.com-ra, hogy fel tudjuk venni a kapcsolatot

Köszönöm,

Miklós

=====

Vá: kapcsolat

Írta: admin - 2009.12.16 00:44

Üdv!

A carlos at nyisztorkaroly.org címen vagyok elérhető.

=====

Képernyő kezelés

Írta: konzolcowboy - 2010.02.13 09:49

Ismét szeretnék tőled segítséget kérni, mert a fejlesztés során egy kicsi akadályba ütköztem.

A programomban van egy bázis képernyő osztály, amely azokat az adattagokat, és virtuális függvényeket tartalmazza, amelyeket minden belőle származtatott képernyőnek implementálnia kell.

Ezen felül van egy képernyő manager osztály, aminek az lenne a feladata, hogy egy listában tárolja az aktív képernyőket, frissíti az adataikat, és lerendereli őket. Nos nekem ezzel a listával van a bajom, mert ugye a képernyőkben csak a bázis osztály a közös, így, ha bemásolnék a list tárolóba mondjuk egy származtatott fő képernyőt, akkor annak csak a bázis osztály része másolódna be, így az objektum felszeletelődne.

A kérdésem az lenne, hogy nincs egy rávezető ötleted, hogy miképpen lehetne a képernyőket egységesen kezelni?

Azért gondoltam egy képernyő manager osztályra, mert vannak esetek, amikor egy képernyő eltakar egy másikat (Pl. pause képernyő).

Előre is köszönöm a segítséget.

=====

Vá: Képernyő kezelés

Írta: admin - 2010.02.13 10:24

A polimorfizmus a válasz a kérdésedre, és úgy érzem, akad itt némi félreértés is. Amennyiben neked van egy

```
// interfész, pure virtual fv-k
class IScreen
{
void update( const double dt ) = 0;
void render() = 0;
}

// az interfészt implementáló osztályok
class CMainMenu : public IScreen
{
...
}

class CPauseMenu : public IScreen
{
...
}
```

```
// vector, amit a manager tart karban
// a különböző menük pointerét tárolja
std::vector myScreenVector;
A myScreenVector tömb az IScreen interface-ből származtatott és azt implementáló osztályok mutatóival töltöd fel, így a polimorfizmus garantálja az adott objektum dinamikus típusának megfelelő viselkedést.
```

Ergo, ha végigiterálsz a vektoron, és az elemeknek meghívod az update() és render() függvényeit, akkor a megfelelő objektum (CMainMenu, CPausedMenu, stb.) függvényei kerülnek végrehajtásra.

Az a lényeg, hogy pointereket használj, illetve gondoskodj a felszabadításukról, ha már nincs rá szükséged.

P.s.

Nincs semmilyen "felszeletelődés", a polimorfizmus az OO-nyelvekben így működik.

=====

Vá: Képernyő kezelés

Írta: konzolcowboy - 2010.02.13 11:55

Értem, köszönöm a válaszod.

A hibát ott követtem el, hogy nem pointerrel dolgoztam, csak változókkal, így azok értéke másolódott be a list objektumba félig meddig.

Az interface osztálynak lehetnek adattagjai?

És ha igen, akkor hogyan (hol) kell nekik értéket adni?

=====

Vá: Képernyő kezelés

Írta: admin - 2010.02.13 23:20

Igen, lehetnek. Több lehetőség is van az értékadásra, pl. a származtatott osztályból közvetlenül (ha nem private-ok), a megfelelő, paraméteres ctor meghívásával:

```
CPausedMenu::CPausedMenu( /*paraméterek*/ ) : IScreen( /*param*/ )
```

Érdemes beszerezni egy jó C++-könyvet, mielőtt belevágnál komolyabb fejlesztésbe.

=====

Vá: Szakmai kérdések

Írta: konzolcowboy - 2010.02.26 21:11

Szépen alakulgat az első kis játékprogramocskám, de ismét szeretnék egy kis segítséget kérni. Szembesültem azzal a problémával, amiről te is említést teszel a Shaderprogramozás könyvedben. A falbontó játékomban a labda egy gyorsabb gépen sokkal gyorsabban mozog, mint egy lassabb gépen. Alkalmaztam a könyvben leírt módszert, nevezetesen, hogy vegyük figyelembe a képkockák között eltelt időt:

$d = (\text{pixel/s}) * dt$, ahol

pixel/s: az egy másodperc alatt megtett távolság pixelben,
dt: két frame között eltelt idő.

Viszont sajnos nem igazán akar működni a dolog, mert a sebességkülönbség így is megvan. :S

Lehet, hogy félre értelmeztem a képletet?

Előre is köszönöm a választ!

Vá: Szakmai kérdések

Írta: admin - 2010.02.26 22:05

A dt (azaz a két képkocka közt eltelt időt) értékét karbantartod, illetve jól számolod ki?

Mert valószínűleg ez lesz a gond.

Ugyanis józan paraszti ésszel is belátható, hogy a képlet működik: minél gyorsabb gépen futtatod, annál kisebb lesz a frame-ek közötti idő, és ezáltal gyorsabb gépen belassul a labda mozgása. Ez biztosítja, hogy gyorsabb gépen sem mozog gyorsabban az egység (labda), mint egy gyengébb konfiguráción.

Vá: Szakmai kérdések

Írta: konzolcowboy - 2010.02.27 08:50

Felhasználtam azt a timer class-t, amit a Shaderprogramozás könyvben mutattál be, de nem kizárt, hogy rosszul használom.

A játék indításakor inicializálom a timer osztályt, és minden update alkalmával meghívom a timer osztály update függvényét.

Beilleszttem a d-t kiszámoló kódot.

```
double d = 0.0f;
```

```
dt = ((const double)myScreenManager->myTimer->getFPS() * (const double)myBalls.myXVel)*  
(myScreenManager->myTimer->getFrameTime());
```

A kapott d értéket hozzáadom a labda aktuális x kordinátájához.

Ezt természetesen Y -ra is kiszámolom.

Nem lehet, hogy a double-int konverzió okozza a gondot? Mert a labda x, y kordinátája int-ben van, így hozzáadásnál csinállok egy konverziót.

Közben megoldódott a probléma. Elgépelttem a kódot, azért nem működött. :woohoo:

=====

Vá: Szakmai kérdések

Írta: admin - 2010.03.02 00:16

Épp akartam válaszolni. (A fenti kód nyilvánvalóan rossz, örülök, hogy erre időközben magadtól rájöttél.)

=====

Vá: Szakmai kérdések

Írta: YAQ - 2010.03.04 00:25

Üdv!

Rég írtam már ide.....:)

Kellene írnom egy mini grafika keretrendszert BGI-vel megspékelve (v hasonlót), ezzel kapcsolatban lenne egy linkelési hibám a VS C++ szerint :(

Leírom a kódrészleteket:

Főprogram:

```
#include
#include
#include "graphics.h"

#include "Renderer.h"
#include "Object.h"
```

Renderer Engine();

```
int main () {
int gd = VGA, gm = VGAMED;
int page = 0;
initwindow(640, 480);
Object *cube = new Object();
cube->Draw ();
closegraph ();
delete cube;
return (0);
}
```

Object.h:

```
#if !defined (OBJECT_H)
#define OBJECT_H
#include
```

```
#include "Renderer.h"
class Object {
private:
    Vertex *pVertexBuffer;
    Vertex **ppIndexBuffer;
    unsigned int NumOfTriangles;
public:
    Object ();
    ~Object ();
    void Draw ();
};
#endif
```

Object.cpp:

```
#include "Object.h"
#include "graphics.h"

extern RenderEngine;

Object::Object () {
//itt van pár csúcspont megadás (azok jók)
}

Object::~~Object() {
//felszabadítás
}

void Object::Draw () {
    Engine.Render (ppIndexBuffer,ppIndexBuffer,ppIndexBuffer);
//ezek Vertex-ekre mutató pointerek
}
```

Renderer.h:

```
#ifndef RENDERER_H
#define RENDERER_H
#include "Vector3D.h"

typedef struct _Vertex {
    Vector3D Position;
    Vector3D Colour;
} Vertex;

class Renderer {
public:
    Renderer ();
    ~Renderer ();
    void Render (Vertex *v1, Vertex *v2, Vertex *v3);
};
```

#endif

Renderer.cpp:

```
#include "Renderer.h"
```

```
void Renderer::Render (Vertex *v1, Vertex *v2, Vertex *v3) {  
}
```

Nah hát bocsánat, hogy ilyen hosszúra sikeredett, amit ki lehetett azt töröltem belőle.

tehát a célom az lenne, hogy azokat a függvényeket, amik a centrális v egyéb vetítéstípusokat, Z-buffer kezelést, háromszögek kitöltését, és hasonló grafika specifikus dolgokat végeznek, külön válasszam a 3D-s modellt definiáló résztől, ahol a vertexkezelés, transzformálások, és egyebek lennének, és azt szeretném, hogy a fenti Renderer típusú Engine-t el lehessen érni bárhol a programban, kb úgy, mintha a directx-es fv-eket hívogatnám, de a fenti kódnál a linker az alábbi hibákat jelzi:

```
1>Object.obj : error LNK2020: unresolved token (0A00038F) "class Renderer Engine"  
(?Engine@@@3VRenderer@@@A)
```

```
1>Object.obj : error LNK2001: unresolved external symbol "class Renderer Engine"  
(?Engine@@@3VRenderer@@@A)
```

```
1>D:\Programfejlesztés\00@Komputergrafika\HF1 Átdöfött kockaDebug\gbi.exe : fatal error LNK1120: 2  
unresolved externals
```

Sajna a linker hibákkal sosem tudtam mit kezdeni, nem is igazán értem, hogy mi a baja, így gondoltam megkérdezek valakit, akinek jóval több tapasztalata van a téren...:)

Választ előre is köszönöm

=====

Vá: Szakmai kérdések

Írta: YAQ - 2010.03.08 23:27

Még egy kérdés!

Kellene z-buffer programot csinálni, ami bgi-ben sikerült is, ám directx-ben nem tudom, hogyan férhetek hozzá a bufferhez.

Van rá mód, vagy saját tömböt kell csinálni?

=====

Vá: Szakmai kérdések

Írta: admin - 2010.03.11 23:31

Jó lenne, ha némileg pontosítanád a kérdést. Szoftveresen akard emulálni a mélységpuffert, vagy a hardveres Z-puffer aktuális tartalmát szeretnéd kiolvasni?

A DirectX és a BGI kombónak pedig nincs sok értelme - feltéve, hogy BGI alatt a régi, DOS-os Borland Graphics Interface-t érted (?).

=====

Vá: Szakmai kérdések

Írta: admin - 2010.03.11 23:53

és azt szeretném, hogy a fenti Renderer típusú Engine-t el lehessen érni bárhol a programban, kb úgy, mintha a directx-es fv-eket hívogatnám,

Sehol sem látom a kódban az Engine deklarációját.

Az extern kulcsszó használata nem indokolt - ugye nem így akarod létrehozni az Engine példányt?

Az extern kulcsszó arra való, hogy jelezd a fordítónak, hogy az adott változót valahol már létrehozták, és bár nem "látja" mégse foglalkozzon vele. A linker persze majd ellenőrzi, hogy ez valóban megtörtént-e, és ha mégsem, akkor jön az esetemben is felbukkanó, (vagy más) hiba.

Ahhoz, amit te szeretnél, az Egyke (Singleton) mintát kellene használni. (Ez egy alapvető és egyszerű szoftvertervezési minta, számtalan helyen olvashatsz róla.)

Illetve megteheted, hogy egyszerűen #include-dal behúzod a Renderer-t a megfelelő helyre (ahol most a rosszul használt extern szerepel), és példányosítod, majd használod, végül megszünteted (kompozíció).

=====

Vá: Szakmai kérdések

Írta: YAQ - 2010.03.12 22:55

A hardveres Z-buffer aktuális tartalmát szeretném kezelni(olvasni, majd alkalmazásban számolni vele, és végül felülírni)

A BGI-t (igen, a borland-os, és itt lenne egy olyan kérdés, hogy ismersz-e olyan elérhető verziót, ami letölthető, van benne egérkezelés, és VSC++ -ban is használható ?, mert ami nekünk van, az egy bgi.a fájl, amit a VS nem képes kezelni, weben találtam egy másik változatot, de abban nincs egérkezelés, és sokkal lassabb a DevC++-által használt .a-s változatnál :() és a DirectX-et eszem ágában sincs keverni, de a gyakvezérünk azt mondta, h legyen benne a programban a z-buffer kezelés jól láthatóan, meg háromszög kirajzolás is valami növekményes módszerrel (ezt már lepötyögtem neki)

Az Engine deklarációja pedig a mainben van, és ahogy a hsz-ből ki vettem, vagy az egyke minta, vagy a gondos és szorgalmas #include marad. (valami még egyszerűbb (lustább megoldást kerestem), de most asszem elleszek azzal, ha mindent globálisként hozok létre, meg egy forrásállományba dolgozok, mert az osztályokból álló projektet nehéz átbüvíteni devcpp-ba.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2010.04.13 10:27

Szervusz Carlos!

Azt szeretném megkérdezni, hogy Visual C++-ban hogyan lehet több processzoron programozni? Sokat kerestem a neten, de általában olyanokat találtam, amik nem teljesen egyértelműek a számomra. Továbbá a Gyakorlati C++ c. könyvedben nem találtam ehhez leírást (de lehet, hogy bamba vagyok :P). Lenne egy másik kérdésem is. Az x86-os architektúrában vannak az MMX és az SSE kiterjesztések. Ehhez tudtommal assembly programozás kell, ami számomra annyira nem probléma. A kérdésem az volna (amihez egyáltalán nem találtam normális leírást), hogy igaz-e az az állítás, miszerint az x64-es arch. ezeket alpból tartalmazza, azaz nem szükséges külön SSE kódokat implementálni? Az előbbiből egy harmadik kérdésem is eszembe jutott. Egy kódban milyen módosítások kellene, ha 64 biten akarok DirectX programot fejleszteni? Válaszodat előre is köszönöm.

=====

Vá: Szakmai kérdések

Írta: admin - 2010.04.16 16:06

Üdv Quantum!

A C++ önmagában nem támogatja a többszálú programozást, mivel az operációs rendszer szintű funkcionalitás. Az adott platform lehetőségeit az OS (Win, *nix, stb.) SDK-jából lehet kibogarászni (Windows esetében lásd MSDN).

Az MMX SSE kérdésben szinte biztos, hogy az újabb processzorok alpból támogatják, a 64 bitesekről nem is beszélve. A DirectX pl. eleve ki van optimalizálva, hogy felismerje és kiaknázza az MMX vagy a 3DNow! előnyeit.

Személy szerint még nem fejlesztettem 64 bites Windows-ra, úgyhogy többet nem tudok segíteni.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2010.04.16 18:32

Köszönöm válaszodat.

Megpróbálom összeszedni a kérdésemet, mert most kicsit megkavarodtam.

Az, hogy a Cpp nem támogatja a többszálú programozást, azt a több magra értetted, vagy alpból a párhuzamos programozásra? Mert olyat már láttam, hogy külön szálakat lehetett programozni, amik egy processzoron párhuzamosan futnak, ilyenekkel a Java nyelvben is találkoztam.

Az MSDN-n megtaláltam a "magosítást", build néven.

Ha játékot fejlesztek DirectX-ben, szükségem van-e arra, hogy szálakat kezeljek, vagy a DirectX automatikusan párhuzamosít? Gondolok itt arra, hogy külön szálon kell futnia pl. a renderelésnek, a mesterséges intelligenciának és a zenének is.

=====

Vá: Szakmai kérdések

Írta: admin - 2010.04.16 22:20

>> "Az, hogy a Cpp nem támogatja a többszálú programozást, azt a több magra értetted, vagy alapból a párhuzamos programozásra?"

A C++ *nem* támogatja a többszálú programozást. Azaz nem rendelkezik szabványos utasításokkal, amelyek a szálak menedzselését tennék lehetővé.

A Java az más, mivel az feltételezi a virtuális gépet (JVM), amely feoldja a platformok közötti különbséget. Amolyan oprendszer az oprendszerben, aminek persze megvan az ára is mellesleg. A C++ forráskód szinten platformfüggetlen ugyan, de csak addig, ameddig nem teszünk bele platformfüggő API-hívásokat. Mivel a C++ fordító nem bytekódot generál, mint a Java, hanem végrehajtható állományokat, az alkalmazást minden egyes target-re (oprendszerre) le kell fordítani és linkelni ahhoz, hogy működjön. Másképp mondva egy Windows-os exe nem fog Linux alatt vagy Mac OSX-en futni.

A másik félreértést is tisztázzuk. A többszálú programozás forrás szinten nem különbözteti meg az egymagos vagy a többmagos processzorokat. Már a 80386-os is képes volt a multithreading-re, de kellett egy Linus Torwalds ahhoz, hogy ezt ki is aknázza. Persze a többszálú programozás valódi előnyeit többmagos vagy multiprocesszoros környezetben lehet igazán élvezni. De hangsúlyozom, fejlesztőként neked ez mindegy, az oprendszer valósítja meg az ütemezést / terhelésmegosztást. Azonban arra is fel kell készülni, hogy többszálú alkalmazások készítésekor újabb problémák merülnek fel, amelyekkel nem szembesülsz egy szál esetén. Ez a fórum kevés ahhoz, hogy a többszálú programozást részletesen kitárgyaljuk, elég haladó témakör, amivel számos könyv foglalkozik. A DirectX nem párhuzamosít automatikusan, azonban egy extra paramétert meg kell adni ahhoz, hogy jelezd a keretrendszernek, hogy több szálát fogsz alkalmazni (legalábbis DX9 alatt). A szálak kezelését neked kell megvalósítanod, beleértve a szinkronizálást és egyéb nyalánkságokat. Szerintem egy egyszerűbb prototípussal kezd, ne egyből egy játékban próbáld ki a többszálás programozást, ahol egyéb tényezők is bekavarhatnak.

Vá: Szakmai kérdések

Írta: Flip - 2010.08.06 16:59

Üdvözlét Mindenkinek!

Nekem 1 C++-al kapcsolatos kérdésem lenne:

Olvastam a könyvben, hogy egy osztályban lehet használni statikus adattagokat is.

Kipróbáltam, és egyszerű esetekkel működik is, ám ha memóriát szeretnék lefoglalni akkor problémákba ütközök.

Tehát ki tudna segíteni valaki, hogy az alábbi kódot hogyan lehet kijavítani:

```
class StaticClass {
public:
    static int *p;
    static void f() {
        p = new int;
    }
};
int StaticClass::*p=NULL;

int main()
{
    StaticClass::f();
    return 0;
}
```

}

A fordító üzenete:

1>Linking...

1>testcpp.obj : error LNK2001: unresolved external symbol "public: static int * StaticClass::p"
(?p@StaticClass@@@2PAHA)

1>D:\cpptestcppDebug\testcpp.exe : fatal error LNK1120: 1 unresolved externals

=====

Vá: Szakmai kérdések

Írta: admin - 2010.08.06 23:03

Rossz helyen van a csillag. ;) A p egy statikus, int* típusú mutató (te pedig egy dereferált pointernek az értékét próbálsz inicializálni a cpp fájlban).

Az ilyen jellegű hibák elkerülése végett azt szoktam javasolni, hogy mindig a típushoz ütköztessük a csillagot, azaz:

```
type* ptr;
```

és nem

```
type *ptr;
```

=====

Vá: Szakmai kérdések

Írta: Flip - 2010.08.07 15:57

Tényleg!, köszönöm a gyors segítséget, magamtól nem jöttem rá :(

Ha nem probléma, lenne még egy hasonló kérdésem. Olvastam a függvény mutatós részt a könyvben és a 9.6.fvptr2.cpp-t ki is próbáltam, ott a fordító először ezt írta:

```
error C3867: 'ClassA::g': function call missing argument list; use '&ClassA::g' to create a pointer to member
```

majd ha a '&ClassA::g' javaslattal éltem, akkor szépen működött is a dolog.

Ezután viszont próbáltam kicsit kreatívabban használni a függvény pointereket, és nem sikerült még rájönnöm, hoy hogyan lehet olyan függvény pointert írni, mely 2 különböző osztály ugyanolyan szignatúrájú metódusára tudna mutatni.

Tehát pl ilyet:

```
class ClassA
```

```
{
```

```
public:
```

```
void fv()
```

```
{
```

```
    cout
```

=====

Vá: Szakmai kérdések

Írta: Quantum - 2010.08.09 10:37

Heló Carlos!

Első sorban egy régebbi segítséget szeretnék megköszönni. Valamiért hiába írtam, nem akarta elküldeni a postomat vagy a rendszer, vagy az én gépem. Aztán elfelejtődött a dolog.

Most ismét segítségedet kérném, jelenleg ténylegesebb kérdésem van.

Létrehoztam egy dinamikus karaktertömb típusú függvényt, amit mikor meghívok, a visszatérési értékét szeretném egy ugyancsak dinamikus karaktertömbnek átadni. Aztán egy olyan gond üti fel a fejét, hogy az első érték-átadás után meghal a program. Ellenben ha a függvényt egy standard kiíratásra küldöm, minden tökéletesen kiíródik, tehát gondolom valahol a változómmal lehet a hiba.

Íme a kódom (kiemelt részlete):

```
#include  
#include  
#include
```

```
char* Kiemel(char sor, int a, char k);
```

```
int _tmain(int argc, _TCHAR* argv)
```

```
{  
    std::ifstream map;
```

```
    map.open("ITEMS.txt");
```

```
    if(map.fail())
```

```
    {  
        std::cerr
```

Vá: Szakmai kérdések

Írta: Quantum - 2010.08.13 09:03

Már lényegtelen, sikerült beüzemeltetni a stringeket.

Vá: Szakmai kérdések

Írta: admin - 2010.08.15 23:57

@Flip: ez már összetettebb eset, mivel az adott osztály this pointerét is át kell adni. Bővebben lásd <http://www.newty.de/fpt/callback.html#member>

Vá: Szakmai kérdések

Írta: Flip - 2010.08.18 23:55

Köszönöm a linket, elolvastam - hasznos volt.

=====

Vá: Szakmai kérdések

Írta: Flip - 2010.08.19 00:01

Közben eszembe jutott egy másik kérdés is: Még régen hallottam olyasmit, hogy lehetőség van néhány programnyelv forráskódjába más jellegű forráskódot is elhelyezni (a cikkben úgy emlékszem, hogy SQL utasítások voltak Java kódban). Az lenne a kérdésem, hogy régi Clipperben írt programokat esetleg el lehet helyezni hasonló módon valamilyen korszerűbb nyelvben (megvan a Clipper fordító + függvények is)?

=====

Vá: Szakmai kérdések

Írta: admin - 2010.08.19 22:53

A "gazda" nyelv, illetve a fordító képességein múlik. A nem-natív kód beszúrása egyre inkább háttérbe szorul, ugyanis több vele a gond, mint a haszon. Például az Assembly betétek látványa C/C++ kódban optimalizációs törekvésekre utalt, és néhány speciális esetben indokolt is a használata. Az SQL vagy SQL-szerű utasításokat szintén sok nyelv, illetve környezet támogatja, hiszen alacsony szintű adatbáziskezelést tesz lehetővé.

A Clipper viszont már nem annyira elterjedt nyelv (a fejvadász-portálok nem igazán találkoznak clipperes munkajánlatokkal), úgyhogy valószínűleg kicsi az esélye annak, hogy bármelyik korszerű nyelv támogatja a beágyazását.

Persze érdemes még egy kört futni a témában, ajánlom a Clipper wiki oldalát:

http://hu.wikipedia.org/wiki/Clipper_%28programoz%C3%A1si_nyelv%29

Amennyiben nagyon értékes az adott Clipper kód, érdemes megfontolni az átültetését az adott (vagy egy beágyazható) prog.nyelvre.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2010.08.27 15:01

Helló Carlos!

Most, hogy kezdek egyre jobban belemerülni a Visual C++ nyelvbe, találkoztam néhány eltéréssel és új dolgokkal a szabványos C++-hoz képest. Egy példát hadd hozzak: a fájlkezelés. A szabványos

változatban ifstream-et használunk, míg a Microsoft-éban ReadFile() fv. is használható. A kérdésem az lenne, hogy van-e fő különbség a használatán kívül ezek között (pl. gyorsaság), ami miatt érdemes a Visual változatot használni, vagy teljesen mindegy?

Segítséged előre is köszönöm.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2010.09.02 12:58

Helló!

Csak annyit szeretnék mondani, hogy a válasz még mindig érdekel.

Egy másik, nem ebbe a topicba illő kérdésem lenne, hogy hol lehet még beszerezni a Gyakorlati C++ könyvedet, mert mindenhol azt írja, hogy elfogyott. Lesz még új kiadás (eddig szorgalmasan kikölcsönöztem, de a könyvtár csak rövid időre adja ki)?

=====

Vá: Szakmai kérdések

Írta: admin - 2010.09.02 18:00

találkoztam néhány eltéréssel és új dolgokkal a szabványos C++-hoz képest.

Ebben a részben szerepel a válasz a kérdésemre: nem szabványos, azaz pl. nem lehet portolni a kódot. Ettől függetlenül sokszor ez nem szempont, és lehet, hogy az adott platformra jobban kihegyezett a nem-szabványos megoldás.

Mellesleg A ReadFile() pont nem jó példa, ha jól emlékszem a Microsoft sem javasolja a használatát.

A Gyakorlati C++ valóban elfogyott, könyvesboltban nem valószínű, hogy kapható még. Próbálkozz antikváriumokban, vaterán, stb. Esetleg írhat sz a Kossuth-nak, hátha akad még náluk pár példány - bár nem hinném.

Egyelőre nem tervezek újabb kiadást, de megtisztelő az érdeklődésed.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2010.09.02 18:12

Köszönöm szépen a válaszokat.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2011.11.13 13:10

Helló Carlos,

MFC-ben mennyire vagy otthon? Egy SDI-t készítettem, ami dokumentuma egy DirectX megjelenítő.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2011.11.13 15:21

Rám jellemző módon, ahogy felteszem a kérdést egy tapasztaltabbnak, rá legfeljebb egy órán belül magam találom meg a választ.

Leírom a gondomat és a megoldásomat, hátha valaki más is találkozik hasonlóval.

Egy SDI alkalmazást írok MFC keretrendszerben, aminek a view-ja DirectX alapú. Mikor megírtam, hogy ha a Ribbon-on ráklickelek egy ikonra és feljön egy dialógus, akkor kegyetlen módon elkezd villogni a feljövő ablak. Addig rá sem lehet klikkelni a gombokra, míg meg nem változtatom az főablak méretét, de persze a dialógus villog tovább.

Úgy sikerült megoldani, hogy az erőforrásoknál megnyitottam a dialógusomat, majd a Style lehetőséget átállítottam Child-ről Overlapped-re (vagy Popup-ra, mindkettővel megszűnik a gond).

További szép napot! ;)

=====

Vá: Szakmai kérdések

Írta: admin - 2011.11.14 12:53

Szia Quantum!

Örülök, hogy magadtól megtaláltad a megoldást. A villodzás mindenképp arra utal, hogy az ablak küzd egy másikkal a Z-order-beli sorrendért. Ilyenkor általában segít, ha modal-ra állítod az ablak típusát. (Amúgy meg MFC-vel úgy tíz éve foglalkoztam utoljára...)

=====

Vá: Szakmai kérdések

Írta: Quantum - 2011.11.15 20:00

A modal-lal kezdtem, de azzal nekem az volt a gondom, hogy ugye akkor a dialógus úgy kapja meg a fókuszot, hogy semmiképpen nem adja át a főablaknak, nekem meg pont az volt a célom, hogy míg a felhasználó kezeli az ott lévő dolgokat, addig a DX ablakban is tudjon dolgozni, mivel szüksége lesz rá. De köszönöm szépen az ötletet.

Egy olyan kérdésem lenne azonban, hogy most megpróbáltam nulláról felépíteni egy DX alkalmazást a könyved alapján, de az miért van, hogy pl. az UnregisterClass-nál mikor első paraméterként megadom az osztálynevet, oda LPCWSTR-t vár, miért van az, hogy nálam kiabál, hogy nem lehet típuskényszeríteni char tömbből? A Te példakódodban pedig teljesen elfogadja. Kihagytam volna

valamit?

=====

Vá: Szakmai kérdések

Írta: Quantum - 2011.11.15 22:45

Természetesen erre is rájöttem. :) A Character Set alapértelmezetten volt, nem Multi-Byte-on.

Egyébként ezt nem nagyon találtam meg a GyakC++-ben (sikerült megszerezni a könyvet), vagy csak elkerülte a figyelmemet, hogy a Cpp-ben miért jó az előredeklarálásnak?

=====

Vá: Szakmai kérdések

Írta: admin - 2011.11.17 13:00

Mi is a kérdés? Hogy miért jó az előre deklarálás?

A rövid válasz: A fordítási függőségek és a fordítási idő csökkentésére használjunk előre-deklarációt, amikor csak lehet.

Írtam egy cikket is erről, ahol részletesen foglalkozom a témával:

<http://www.nyisztorkaroly.org/programozas-tervezes/az-elore-deklaral-as-forward-declaration.html>.

=====

Vá: Szakmai kérdések

Írta: Quantum - 2011.11.17 13:14

Köszönöm szépen a választ.

Arra lennék még kíváncsi, hogy a függvények törzseit miért kell feltüntetni a forrás elején, ha a bizonyos függvény a kódban hátrébb van, mint ahol meghívjuk? Tudom, hogy ez C hagyaték, de miért van erre mindenképpen szükség, mi van ilyenkor a fordító "fejében"? Aztán gondolom emiatt kell két forrást kreálni egyetlen osztályhoz (már ha ketté szedjük).

=====

Vá: Szakmai kérdések

Írta: Quantum - 2011.11.23 00:09

Közben ezt is megtaláltam a könyvedben. Régen olvastam ezt a részt, meg valamiért a prototípus szóra emlékeztem.

Nem zavarlak tovább a sok bugyuta kérdéssel.

Talán még csak annyit, hogy lehet valami aktuálisat megtudni az új könyvedről?

=====

Vá: Szakmai kérdések

Írta: Quantum - 2012.05.08 21:52

Szia Carlos!

Minden alkotónak jó a pozitív visszajelzés, ezért úgy gondoltam én is írok neked egyet.

Köszönöm szépen ezt a három nagyszerű könyvet, ezeknek köszönhetek most egy nagyon jó munkahelyet (orvosi kutatórendszereket fejlesztünk). Sok szakmai irodalmat olvastam már a programozás területén, de azt hiszem a tieid lettek a legérthetőbbek számomra (ebből a Gyakorlati C++-t emelném ki, annál jobb Cpp könyvet még nem találtam). Remélem fogod tudni folytatni a jövőben az írást, nagyon hasznos, amit csinálsz.

Még egyszer köszönöm és további munkasikereket kívánok! ;)

=====

Vá: Szakmai kérdések

Írta: admin - 2012.05.08 23:37

Köszönöm, Quantum! :) És nagyon örülök, hogy tetszik az új (első?) munkahelyed! Sok sikert!
(Csak nem az evo-nál dolgozol?)

=====

Vá: Szakmai kérdések

Írta: Quantum - 2012.05.08 23:41

Neem, egy debreceni cégnél. :)

=====