

PROGRAMAVIMAS IR PROGRAMINĖ ĮRANGA

OpenGL ir DirectX technologijų spartos palyginimas programuojant trimačius vaizdus

Ernestas Filatovas

Vilniaus Gedimino technikos universiteto
Fundamentinių mokslų fakulteto
lektorius, daktaras
Vilnius Gediminas Technical University,
Faculty of Fundamental Sciences,
Lecturer, Doctor
Saulėtekio al. 11–615, LT-10223, Vilnius
El. paštas: Ernest.Filatov@gmail.com

Grigorijus Melničenko

Lietuvos edukologijos universiteto
Gamtos, matematikos ir technologijų fakulteto
docentas, daktaras
Lithuanian University of Educational Sciences,
Faculty of Science and Technology,
Assoc. Professor, Doctor
Studentų g. 39, LT-08106, Vilnius
El. paštas: gmelnicenko@gmail.com

Vytautas Jakštys

Lietuvos edukologijos universiteto Gamtos, matematikos ir technologijų fakulteto magistras
Lithuanian University of Educational Sciences, Faculty of Science and Technology, Master
Studentų g. 39, LT-08106, Vilnius
El. paštas: vitaakas@gmail.com

Pagrindiniai trimačių vaizdų apdorojimo konkurentai yra DirectX ir OpenGL technologijos. Straipsnyje pateikiama šių technologijų lyginamoji analizė. Eksperimentiškai tiriama DirectX ir OpenGL technologijų veikimo sparta. Eksperimentams skirta testinė programinė įranga sukurta populiariausiomis programavimo kalbomis C++ ir C#. Darbe pateikiami ir analizuojami tyrimų rezultatai, gauti atlikus kelis skirtingus spartos testus naudojant sukurtą programinę įrangą.

Įvadas

Grafinis duomenų apdorojimas ir vaizdavimas plačiai naudojamas daugelyje sričių – moksle, versle, inžinerijoje, mene, reklamoje, laisvalaikio poreikiams ir kt. Modernus pasaulis jau neįsivaizduojamas be trimatės kompiuterinės grafikos. Su trimate grafika susiduriama tokiose srityse, kaip realaus laiko modeliavimo sistemos, kompiuteriniai žaidimai, filmai. Čia trimatės grafinės informacijos kokybė ir atnaujinimo sparta yra vienas svarbiausių parametru,

lemiančių jos naudojimo efektyvumą, patogumą, patrauklumą ir populiarumą.

Daugelį metų programuotojų bendruomenėje diskutuojama apie praktikoje naudojamų trimačių vaizdų apdorojimo technologijų privalumus ir trūkumus. Pagrindiniai konkurentai trimačių vaizdų programavime srityje yra bibliotekų rinkiniai *DirectX* ir *OpenGL*. Vienareikšmiškai įvardinti, kuri iš šių technologijų yra geresnė, labai sudėtinga.

Minėtos technologijos dažniausiai naudojamos kuriant programinę įrangą C++ ir C# progra-

mavimo kalbomis. Mokslinėje literatūroje šiuolaikinių *OpenGL* ir *DirectX* versijų greitaveikos eksperimentinio lyginimo rezultatų neskelbiama. Interneto tinklapiuose ir forumuose nemažai rašoma apie konkrečios technologijos pranašumus, arba trūkumus, tačiau dauguma iš pasisakymų yra subjektyvus ir neparemti eksperimentiniais tyrimais. O keli internete skelbiami *DirectX* ir *OpenGL* technologijų veikimo spartos eksperimentinių tyrimų rezultatai (*OpenGL*, 2013; Sebastian, 2012) nėra patikimi, nes eksperimentai yra neišsamūs (atlikti tik vienu kompiuteriu, tik viena operacine sistema, per mažai eksperimentų). Todėl šiuo darbu siekiama atlikti *DirectX* ir *OpenGL* technologijų lyginamąją analizę, o taipogi eksperimentiškai iširti jų greitaveiką. *DirectX* ir *OpenGL* technologijos nagrinėjamos C++ ir C# programavimo kalbų pagrindu.

Trimačių vaizdų programavimo technologijos

Konkuruoja dvi trimačių vaizdų (trimačės grafikos) programavimo technologijos – *DirectX* ir *OpenGL*.

OpenGL (angl. *Open Graphics Library*) – tai grafinės kompiuterio techninės įrangos programinė sąsaja (bibliotekų rinkinys). Bibliotekos susideda iš funkcijų, leidžiančių programuotojui atvaizduoti objektus vaizduoklyje, atlikti įvairias operacijas su jais, kurti aukštos kokybės grafinius vaizdus ir tekstūras, naudojamas objektams spalvinti (*OpenGL*, 2013; Shreiner et al., 2013).

OpenGL bibliotekas būtų galima apibūdinti kaip komandas, leidžiančias aprašyti dvimačius ar trimačius objektus, bei komandų rinkinį, kuris nustato, kaip objektas vaizduojamas ekrane, t. y. įkeliamas į ekrano buferį (angl. *Framebuffer*), saugomą darbinėje atmintyje. *OpenGL* gali veikti dviem režimais: tiesioginiu (angl. *Immediate mode*) ir netiesioginiu (angl. *Retained mode*). Pirmu atveju grafinius objektus formuojantys primityvai (primityvios figūros, pvz., trikampiai) vizualizuojami ekrane tiesiogiai ir nenaudojama papildoma vaizdo plokštės atmintis primityvams saugoti (informacija apie primityvus prarandama juos vizualiza-

vus). Antru atveju, programos vykdymo metu, primityvios figūros išsaugomos specialiaame sąrašė, kuris saugomas vaizdo plokštės atmintinėje. Tokiu būdu primityvai gali būti atvaizduoti ekrane kelis kartus ir panaudoti keliems skirtingiems objektams piešti.

Dalis *OpenGL* funkcijų yra skirtos elementariausiems geometriniais objektams (taškams, atkarpoms, daugiakampiams) piešti. Kita dalis – objektų atvaizdavimui ekrane, įskaitant apšvietimą, spalvinimą, išsidėstymą, atvaizdavimą trimatėje koordinacių sistemoje ir kt. Taip pat yra numatytos funkcijos, leidžiančios tiesiogiai kreiptis į ekrano buferį, t. y. galima skaityti ir rašyti į atskirus ekrano pikselius. Tiek tekstas, tiek vaizdai vaizduoklio ekrane konstruojami iš taškų (pikselių) matricos.

OpenGL atlieka tik skaitymą iš ekrano buferio ir rašymą į jį. *OpenGL* nepalaiko ir aukšto lygio komandų, aprašančių trimačius objektus. Tokios komandos leistų atvaizduoti sudėtingus geometrinius objektus, pavyzdžiui, lėktuvus ar automobilius. *OpenGL* tik sudaro galimybes programuotojui visa tai sukurti pačiam iš nedidelio skaičiaus geometrinių primityvų: taškų, linijų ir daugiakampių.

OpenGL sudaro kelios plačiai naudojamos bibliotekos: *GL*, *GLU*, *WGL*, *GLX*, *GLUT*, *GLFW*, *GLM*, *WebGL* ir kt., kurių detalesnis aprašymas pateiktas šaltiniuose (*OpenGL*, 2013; Shreiner et al., 2013).

DirectX – tai bendras aplikacijų programavimo sąsajų (API) rinkinys, skirtas daugialypės terpės užduotims tvarkyti, ypač kurti tuos žaidimus, kurie vėliau bus žaidžiami naudojant *Microsoft* produktus, tokius kaip *Windows*, *Xbox* ir *Xbox 360* (*DirectX*, 2013; *DirectX* Graphics, 2013; Luna, 2008).

Nors *DirectX* bibliotekų rinkinys iš pradžių buvo skirtas žaidimų programavimui *Windows* operacinei sistemai, šiandien jis yra naudojamas ne tik žaidimams, bet vaizdą bei garsą apdorojančioms ir išvedančioms programoms, bei matematiniams skaičiavimams. Naujesnėse *Windows* operacinėse sistemose, tokiose kaip *Windows 7* ir *Windows 8*, *DirectX* tapo ir pagrindiniu operacinės sistemos grafinės informacijos atvaizdavimo varikliu.

Šiuolaikinės *DirectX* versijos objektus vizualizuoja tik tiesioginiu režimu.

DirectX yra platinamas dviem programų paketais:

- *Runtime* (skirtu naudotojams);
- SDK (skirtu programuotojams).

DirectX technologija susideda iš kelių skirtingų su COM (angl. *Component Object Model*) suderinamų komponentų (bibliotekų): *DirectX Graphics*, *DirectInput*, *DirectPlay*, *DirectSound*, *DirectSound3D*, *DirectMusic*, *DirectX Media Object*, *DirectX Media Objects*, kurių detalesnis aprašymas pateiktas šaltiniuose (Luna, 2008; Sherrod, Jones, 2011).

COM trumpai gali būti apibrėžtas kaip technologinis programinės įrangos kūrimo standartas, kuris remiasi tarpusavyje sąveikaujančiais komponentais. Programos, sukurtos remiantis COM standartu, iš tiesų nėra savarankiškos, o sudaro tarpusavyje sąveikaujančių COM komponentų rinkinį. Šie komponentai gali būti naudojami vienu metu skirtingomis aplikacijomis (COM, 2013; What is COM, 2013). „Microsoft“ šią technologiją pristatė 1993 m. COM technologija yra kitų *Microsoft* sukurtų ir plačiai naudojamų technologijų pagrindas, pavyzdžiui, *OLE*, *ActiveX*, *COM+*, *DCOM*, *DirectX*, *Windows Runtime* (Sherrod, Jones, 2011).

Kaip minėta, yra dvi pagrindinės technologijos trimatei grafikai programuoti: *DirectX* ir *OpenGL*. Rinkoje dažniau naudojama *Microsoft* sukurta technologija *DirectX*, ypač žaidimuose ir grafinėse sistemose, kuriose būtinas aparatinis spartinimas. *OpenGL* technologija plačiau naudojama CAD sistemose, vizualizacijai moksliniuose tyrimuose, skrydžių simuliacijai (Mažutis, 2009). Tačiau, renkantis trimačių vaizdų programavimo priemonę, reikia išskirti ir gerai išanalizuoti pagrindinius *DirectX* ir *OpenGL* technologijų pranašumus ir trūkumus.

Išanalizavus literatūros šaltinius (Comparison, 2013; Hill, Kelley, 2007; Mažutis, 2009; *OpenGL...*, 2013; *OpenGL*, 2013; Shreiner et al., 2013; Wright et al., 2010), pateikiamos pagrindinės *OpenGL* technologijos savybės.

Teigiamos savybės:

- Galima naudoti įvairiose operacinėse sistemose;
- Gerai išnaudoja pritaikytos aparatūros spartinimo galimybes;
- Turtinga funkcijų biblioteka;
- Palaiko daugiau nei vieną grafikos sąsają (angl. *Multiple Graphics Adapters*);
- Palaiko išplėtimą (angl. *Extentions*);
- Nemokama;
- Stabili.

Neigiamos savybės:

- Ne visos vaizdo plokštės palaiko aparatinę *OpenGL* spartinimą;
- Naudojimui būtinos papildomos bibliotekos;
- Versijos pasirinkimo problema (dėl produkto atvirumo yra išleistos kelios skirtingų kūrėjų *OpenGL* versijos);
- Kai kurios bibliotekos yra mokamos.

Išanalizavus literatūros šaltinius (Cai et al., 2011; Comparison, 2013; *DirectX*, 2013; *DirectX...*, 2013; Guo et al., 2012; Luna, 2006 ir 2008; Mulholland, Hakala, 2004; Sherrod, Jones, 2011) straipsnyje yra pateikiamos pagrindinės *DirectX* technologijos savybės.

Teigiamos savybės:

- Paprasta panaudoti;
- Labai gerai išnaudoja pritaikytos aparatūros spartinimo galimybes;
- Didžioji dauguma vaizdo plokščių palaiko aparatinę *DirectX* spartinimą;
- Integruota į *Microsoft* operacines sistemas;
- Yra daug patogių programavimo įrankių, nesudėtingai palaikančių *DirectX* bibliotekas;
- Labai daug dokumentacijos ir pavyzdžių;
- Platus naudojimas, kas verčia nuolat tobulinti *DirectX*;
- Palaiko išplėtimą;
- Palaiko daugiau nei vieną grafikos sąsają;
- Yra CAD (angl. *Computer-aided design*) sistemų, kurios turi failų formatą, tiesiogiai pritaikytą *DirectX* aplinkai;
- Nemokama;
- Stabili.

Neigiamos savybės:

- Palaiko tik vieną operacinių sistemų šeimą *Windows*;
- Kad būtų galima naudoti, reikalingos papildomos bibliotekos.

Svarbu pažymėti, kad *OpenGL* ir *DirectX* architektūros yra skirtingos: *OpenGL* technologija duomenims perduoti naudoja procedūrinį modelį, o *DirectX* technologija remiasi COM modeliu, kurio naudojimo sudėtingumą šiaandien supaprastina *Microsoft .NET Framework* (Comparison, 2013; Guo et al., 2013; Mulholland, Hakala, 2004).

C++ ir C# programavimo kalbos programuojant trimačius vaizdus

Kaip minėta ankstesniame skyriuje, pagrindinės trimačių vaizdų programavimo technologijos yra *DirectX* ir *OpenGL*. Kad būtų galima naudoti šias technologijas, programuotojui reikia parinkti programavimo kalbą. Yra sukurta gausybė programavimo kalbų, tačiau populiariausios yra kelios. Kompanija TIOBE kas mėnesį skelbia populiariausių pasaulyje programavimo kalbų sąrašą. Pirmoje lentelėje pateikiama informacija pagal 2013 m. balandžio mėnesio duomenis (Tiobe..., 2013).

Taigi tarp populiariausių modernių programavimo kalbų, tinkamiausių trimačiams vaizdams programuoti, yra trys: C++, Objective-C ir C#. Objective-C yra pagrindinė programavimo kalba, skirta *OS X* ir *iOS* operacinėms sistemoms ir naudojama *Apple* kompanijos produktų programavimui (Kochan, 2011). Šiame straipsnyje apsisistota ties C++ ir C# programavimo kalbomis, kurios gerai pritaikomos *Windows* operacinių sistemų šeimai ir jos grafikos programavimui. Abejose kalbose galima naudoti tiek *DirectX*, tiek *OpenGL* technologijas.

Norint programiškai valdyti (programuoti) trimatę vizualizaciją reikia prijungti atitinkamas *OpenGL* ir *DirectX* bibliotekas prie kuriamos programos kodo. Pastebėtina, kad, vizualizuojant tiek *OpenGL*, tiek *DirectX* technologijomis, C++ ir C# programavimo kalboms sukurtos ir naudojamos skirtingos bibliotekos. Kaip buvo

1 lentelė. Populiariausios programavimo kalbos 2013 m. balandžio mėn. duomenimis

| Pozicija | Programavimo kalba | Naudojimas |
|----------|--------------------|--------------|
| 1 | C | 17,862 proc. |
| 2 | Java | 17,681 proc. |
| 3 | C++ | 9,714 proc. |
| 4 | Objective-C | 9,598 proc. |
| 5 | C# | 6,150 proc. |
| 6 | PHP | 5,428 proc. |
| 7 | (Visual) Basic | 4,699 proc. |
| 8 | Python | 4,442 proc. |
| 9 | Perl | 2,335 proc. |
| 10 | Ruby | 1,972 proc. |

minėta ankstesniame skyrelyje, dėl kodo atvirumo yra išleistos kelios skirtingų kūrėjų *OpenGL* bibliotekos. Tyrimuose buvo naudojamos tokios populiariausios bibliotekos:

- C++ kalboje darbui su *OpenGL* technologija naudojama nemokama biblioteka *glut32.dll* (Freeglut, 2013), o su *DirectX* technologija – biblioteka *d3dx11_43.dll* (Download, 2013).
- C# kalboje darbui su *OpenGL* technologija naudojamos nemokamos bibliotekos *Tao.OpenGL.dll*, *Tao.Paltform.Windows.dll*, *Tao.FreeGlut.dll* (Mono, 2013), o su *DirectX* – *Microsoft.DirectX.dll*, *Microsoft.DirectX.Direct3D.dll* ir *Microsoft.DirectX.Direct3DX.dll* (Download, 2013).

Galimi ir kiti deriniai (taip pat mokamos *OpenGL* bibliotekos), tačiau eksperimentiniams tyrimams buvo pasirinktos populiariausių nemokamų bibliotekų naujausios versijos.

Eksperimentiniai tyrimai

Naudodamas *DirectX* ir *OpenGL* technologijas, programuotojas negali valdyti, kada vizualizaciją (angl. *Rendering*) atliks centrinis procesorius (CPU), o kada vaizdo plokštės procesorius (GPU). Kompiuterio išteklius bibliotekos paskirsto automatiškai.

Šiuolaikiniuose vaizduokliuose vaizdas atnaujinamas apie 60–120 kartų per sekundę (vaizdo atnaujinimo dažnis yra 60–120 Hz),

o priklausomai nuo CPU ir GPU galimybių ir trimačių vaizdų sudėtingumo, jis gali būti atnaujintas (perskaičiuotas) kelis tūkstančius kartų per sekundę. Taigi standartiškai *DirectX* ir *OpenGL* bibliotekose uždėti ribotuvai tam, kad trimačiai vaizdai būtų atnaujinami ne dažniau, nei vaizduoklis gali tai padaryti. Kuriant programinę įrangą eksperimentiniams tyrimams į tai yra atsižvelgta ir toliau darbe pateikiami tie rezultatai, kurie gauti (apskaičiuoti) kompiuterių procesoriais, o ne tie, kurie rodomi vaizduokliuose.

Trimačių vaizdų vizualizacijoje dažniausiai objektai sudaromi iš elementarių figūrų – trikampių (Zhang et al., 2012). Kuo sudėtingesnė figūra, tuo daugiau elementarių trikampių yra panaudota jai suformuoti, tuo daugiau skaičiavimo išteklių reikia vaizdų formavimui vaizduoklyje. Be to, reikia papildomų skaičiavimų išteklių reikia, kai, pavyzdžiui, žaidimuose naudojamas didelis vaizdo gilumas, kuo iš toliau matomas objektas žaidime, tuo didesnis vaizdo gilumas.

Eksperimentiškai buvo siekiama nustatyti *DirectX* ir *OpenGL* technologijų vizualizacijos spartą tokiais atvejais:

- Elementarios (trikampės) figūros judėjimas;
- Sudėtinės (sudarytos iš daugelio elementarių trikampių) figūros judėjimas;
- Figūrų, esančių skirtinguose gilumuose, judėjimas (vizualizuojamos figūros atitolinamos).

Programinė įranga, skirta eksperimentams, buvo sukurta naudojant *Microsoft Visual Studio* aplinką. Visomis sukurtomis programomis, skirtomis eksperimentiniams tyrimams, vizualizuojamos figūros buvo formuojamos standartiniame *Windows Form* lange. Jos buvo vaizduojamos vienodo dydžio, rodymo kampo, perspektyvos, tačiau dėl skirtingose bibliotekose naudojamų skirtingų funkcijų minėtų programų realizacijos skyrėsi.

Darbe eksperimentiniu būdu buvo siekiama nustatyti, kiek skiriasi *DirectX* ir *OpenGL* technologijų sparta, jomis vizualizuojant judančias elementarias (trikampes) ir sudėtines (sudarytas iš daugelio elementarių) figūras.

Buvo atlikti eksperimentai, kai vienu metu buvo vizualizuojamas skirtingas trikampių skaičius:

- 1 trikampis – eksperimentuose vadinama T1;
- 3 trikampiai – eksperimentuose vadinama T3;
- 7 trikampiai – eksperimentuose vadinama T7.

Buvo atlikti ir sudėtinės figūros – sferos vizualizacijos eksperimentai. Sferos buvo sudaromos iš skirtingo elementarių figūrų skaičiaus:

- iš 100 figūrų – eksperimentuose vadinama S100;
- iš 2500 figūrų – eksperimentuose vadinama S2500;
- iš 10000 figūrų – eksperimentuose vadinama S10000.

Kiekvienas eksperimentas (vizualizacija) buvo atliktas nustatant figūroms skirtingus gilumus (atitolinimus). Eksperimentuose buvo naudojami tokie gilumai:

- 0 – figūros nėra atitolinamos;
- 2 – figūros atitolinamos tam tikru atstumu, vaizduoklyje matyti visas figūras;
- 2,5 – figūros atitolinamos ir vaizduoklyje matyti tik trys ketvirtadaliai figūrų;
- 3 – figūros atitolinamos ir vaizduoklyje matyti tik pusė figūrų;
- 3,5 – figūros atitolinamos ir vaizduoklyje matyti tik ketvirtadalis figūrų;
- 4 ir 8 – figūros atitolinamos ir vaizduoklyje figūrų nematyti.

Pavyzdžiui, jei gilumas yra 2, objektas atrodys 2 kartus mažesnis už tą, kurio gilumas yra 0.

Eksperimentai buvo atliekami įvairiuose kompiuteriuose su skirtingomis *Windows* operacinės sistemos versijomis ir skirtingomis kompiuterių komplektacijomis (CPU, GPU, RAM). *DirectX* ir *OpenGL* greitaveika matuojama atnaujinimo kadrais per sekundę (atnaujinimo dažnis) (angl. *Frames Per Second* – FPS).

EEksperimentams sukurtomis programomis gauti vaizdai parodyti 1 pav. ir 2 pav. Juose matyti trikampių ir sferų vizualizacijos, kai gilumas nesiskyrė (jis šiais atvejais lygus 0). Sferų vizualizacijos, kai nustatyti skirtingi gilumai sferai, sudarytai iš 2500 elementarių figūrų, matyti 3 pav.

Ekspirimentuose naudoti kompiuteriai ir jų kompleksacijos patekiamos 2 lentelėje.

Vizualizacijos metu atnaujinimo dažnis šiek tiek kinta, todėl kiekvienas pirmiau aprašytas eksperimentas truko 1 minutę ir buvo kartojamas po 10 kartų kiekviename kompiuteryje. Taigi bendra eksperimentų trukmė buvo mažiausiai 168 valandos. Buvo apskaičiuoti 10-ies eksperimentų rezultatų vidurkiai.

Skirtinguose kompiuteriuose atliktų vizualizacijų rezultatai pateikiami 4 pav. Siekiant palyginti *DirectX* ir *OpenGL* technologijų trimačių vaizdų vizualizavimo spartą atliekant skirtingų figūrų vizualizacijas, buvo apskaičiuotos kiek-

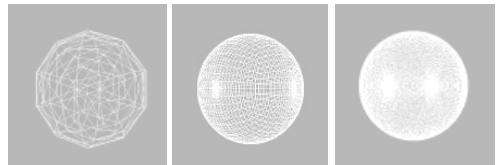
vieno iš šešių eksperimentų, atliktų su visais gilumais, rezultatų vidutinės reikšmės (kiekvieno eksperimento atskirai).

Taikant *DirectX* technologiją visuose eksperimentams naudotuose kompiuteriuose pasireiškia C# kalbos pranašumas (4 pav.). Kitaip veikė *OpenGL* technologija: pastebėta, kad kompiuteryje 1, kompiuteryje 3 ir kompiuteryje 4 C++ kalba pranašesnė nei C#. Be to, beveik visuose eksperimentuose nustatyta, kad *DirectX* pranašumas už *OpenGL* technologiją.

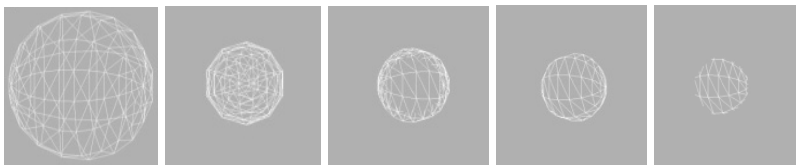
5 pav. yra pateiktos visų kompiuterių kiekvieno eksperimento, atlikto su visais gilumais, rezultatų vidutinės reikšmės. Akivaizdu,



1 pav. *Trikampių vizualizacijos*



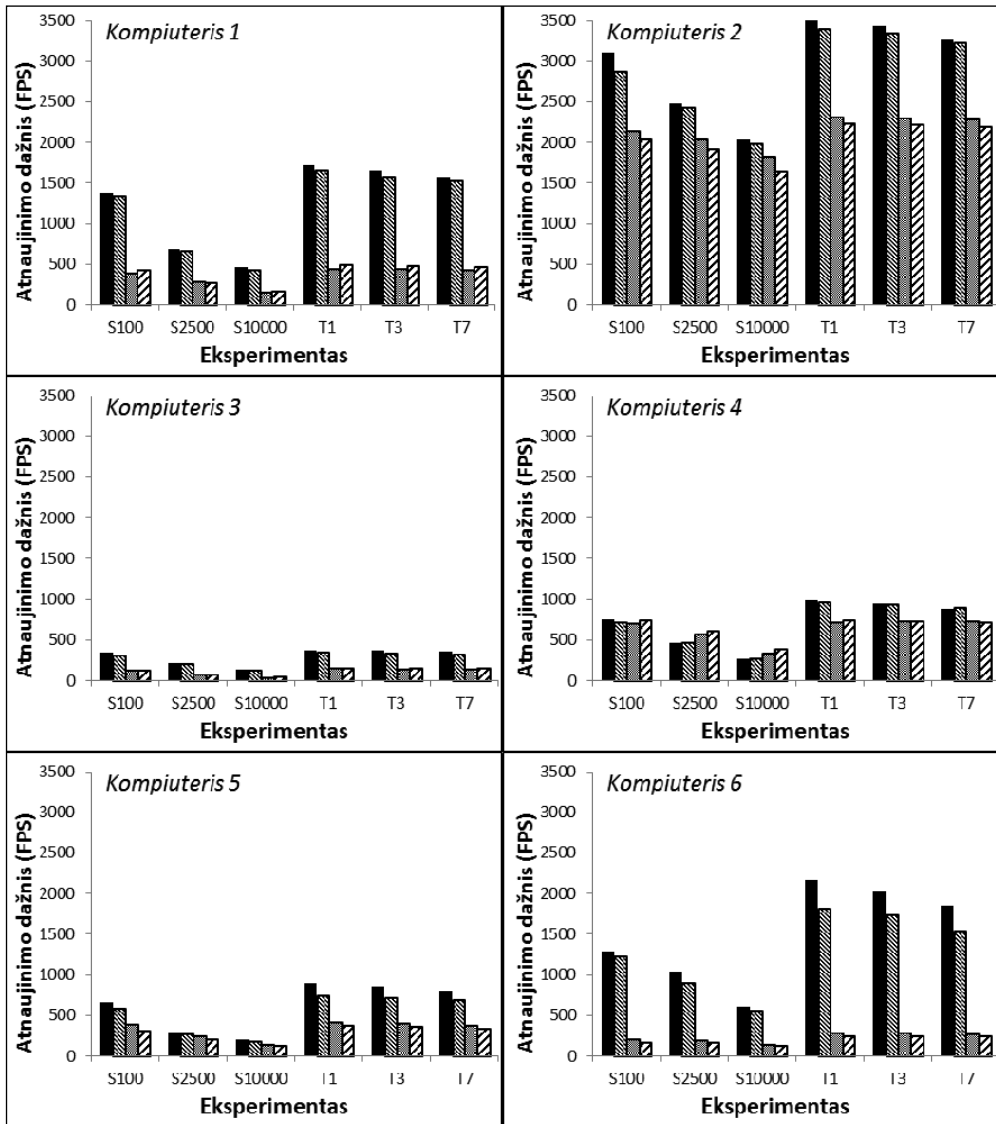
2 pav. *Sferų vizualizacijos*



3 pav. *Sferos vizualizacijos su skirtingu gilumu (0; 2; 2,5; 3; 3,5)*

2 lentelė. *Ekspirimentams naudoti kompiuteriai*

| Kompiuterio pavadinimas | Kompiuterio modelis | Operacinė sistema | CPU (centrinis procesorius) | GPU (vaizdo plokštės procesorius) | RAM (darbinė atmintinė) |
|-------------------------|-------------------------|-------------------|--|--|-------------------------|
| Kompiuteris 1 | Lenovo X201 | Windows 7 x64 | Intel Core i7 M 620 2.67GHz | Intel Graphic Media Accelerator HD | 8 GB |
| Kompiuteris 2 | Asmeninis kompiuteris 1 | Windows 7 x64 | Intel Core Quad Q9300 2.5GHz | 2xAMD Radeon HD 6900 Series | 16 GB |
| Kompiuteris 3 | Asus EeePC | Windows XP x32 | Intel(R) Atom(TM) 1.6GHz | Mobile Intel(R) 945 Express Chipset Family | 2 GB |
| Kompiuteris 4 | Samsung NP350V5C | Windows 8 x64 | Intel Core i3-3110M CPU 2.40GHz | Intel HD Graphics 4000 | 4 GB |
| Kompiuteris 5 | ASUS KJ501J | Windows 7 x64 | Pentium Dual Core T4200 2.00GHz | Mobile Intel 45 Express Chipset Family | 4 GB |
| Kompiuteris 6 | Asmeninis kompiuteris 2 | Windows 7 x64 | AMD Athlon 64 X2 Dual Core Processor 5000+ 2.6 GHz | ATI Radeon Xpress 1150 | 4 GB |

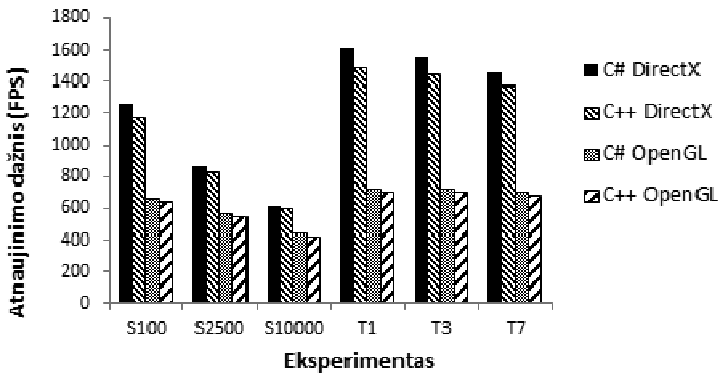


4 p a v. Skirtingų figūrų vizualizavimo eksperimentų rezultatai

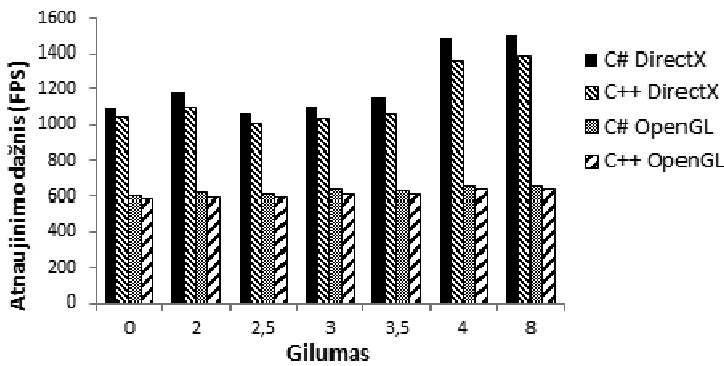
kad *DirectX* technologija yra pranašesnė už *OpenGL*. Esant gana nesudėtingam vizualizacijos uždaviniui (eksperimentas S100) *DirectX* yra 1,94 karto spartesnė už *OpenGL*, kai programuojama C# kalba, ir 1,86 karto, kai programuojama C++. Šis pranašumas sumažėja, kai vizualizacijos uždavinys tampa sudėtingesnis (eksperimentas S10000) – *DirectX* spartesnė

1,41 karto, kai programuojama C#, ir 1,43 karto, kai programuojama C++.

Kaip ir galima buvo tikėtis, visuose kompiuteriuose, didinant vizualizuojamų elementarių figūrų skaičių (eksperimentai T1, T3, T7), *DirectX* ir *OpenGL* technologijų vizualizacijos sparta mažėja. Lyginant elementarių figūrų vizualizacijas (eksperimentai T1 ir T7), *DirectX*



5 pav. Visu kompiuterių skirtingų figūrų judėjimo eksperimentų rezultatų vidurkiai



6 pav. Visu kompiuterių skirtingų gilumų rezultatų vidurkiai

veikimo sparta sumažėja 10 proc, kai programuojama C# kalba, ir 8 proc – kai C++; veikiant *OpenGL* – atitinkamai 2 proc. ir 3 proc.

Vizualizacijos sparta mažėja, kai didinamas sudėtinės figūros elementų skaičius (eksperimentai S100, S2500, S10000). Lyginant sudėtinių figūrų vizualizacijas (eksperimentai S100 ir S10000), *DirectX* atveju sparta sumažėja 51 proc., kai programuojama C# kalba ir 50 proc. – kai C++; naudojantis *OpenGL* – atitinkamai 33 proc. ir 34 proc.

Taigi *DirectX* technologija uždavinio sudėtingumui yra jautresnė nei *OpenGL* – sudėtingėjant uždaviniui *DirectX* sparta krinta greičiau.

Apskaičiuoti visų šešių kiekvieno gilumo eksperimentų rezultatų vidurkiai pateikiami 6 pav. Teoriškai, didinat gilumą, atnaujinimo dažnis

turėtų didėti, nes vizualizuojant mažesniu gilumu, ekrane piešiama didesnė figūra, o tai reikalauja daugiau skaičiavimo resursų. Tuo įsitikinama lyginant atvejus, kai gilumas yra 0 ir 2, – atnaujinimo sparta antru atveju yra vidutiniškai didesnė. Tačiau tais atvejais, kai piešiama tik figūros dalis (gilumai 2,5; 3; 3,5), procesoriams reikia atlikti papildomų skaičiavimų. Todėl, kai gilumas yra 2,5, atnaujinimo sparta sumažėja lyginant su rezultatu, kai gilumas yra 2. Tais atvejais, kai vizualizuojant rodoma tik figūros dalis, kuo mažesnę dalį reikia vizualizuoti, tuo atnaujinimo sparta yra didesnė. Kai gilumas yra 4 ir 8, vizualizacijos skaičiavimai atliekami, tačiau ekrane vaizdo nematyti, todėl atnaujinimo dažnis padidėja, palyginant su visais gilumo atvejais.

Tinklalapiuose <http://www.cpubenchmark.net> ir <http://www.videocardbenchmark.net> skelbiami nepriklausomos bendrovės *PassMark® Software Pty Ltd* kompiuterio centrinio procesoriaus ir vaizdo plokštės našumo testų rezultatai. Testai atliekami skirtinguose savanoriškai dalyvaujančių *PassMark* programoje vartotojų kompiuteriuose. *CPU Benchmarks* indeksas gaunamas apskaičiuojant vidurkį iš skirtingų 8 centrinio procesoriaus našumo testų (CPU, 2013). 2013 m. gegužės 17 d. duomenimis, sparčiausio procesoriaus indeksas buvo 14969. *Videocard Benchmarks* indeksas gaunamas apskaičiuojant vidurkį iš skirtingų trijų vaizdo plokštės našumo testų (PassMark, 2013). Tais pačiais duomenimis, sparčiausios vaizdo plokštės indeksas buvo 8404. Minėti testai detalčiau aprašyti *PassMark® Software Pty Ltd* bendrovės tinklapyje <http://www.passmark.com/>.

Trečioje lentelėje pateikti eksperimentuose naudotų kompiuterių *MarkPass CPU* ir *MarkPass G3D* indeksai (reitingai) bei skirtingų figūrų vizualizavimo eksperimentų rezultatai (*DirectX* ir *OpenGL* technologijų spartos santykis), apskaičiuojant kiekviename eksperimente gautų rezultatų vidurkį. *MarkPass CPU* – centrinio procesoriaus indeksas pagal *CPU Benchmarks* – <http://www.cpubenchmark.net> (žiūrėta 2013-05-17); *MarkPass G3D* – vaizdo plokštės indeksas pagal *Videocard Benchmarks* – <http://www.videocardbenchmark.net> (žiūrėta 2013-05-17).

3 lentelė. *DirectX* ir *OpenGL* technologijų spartos skirtinguose kompiuteriuose santykis

| Kompiuteris | MarkPass CPU | MarkPass G3D | DirectX ir OpenGL spartos santykis | |
|---------------|--------------|--------------|------------------------------------|------|
| | | | C# | C++ |
| Kompiuteris 1 | 2709 | 113 | 3,50 | 3,16 |
| Kompiuteris 2 | 3191 | 2x1983 | 1,38 | 1,41 |
| Kompiuteris 3 | 311 | 4 | 2,78 | 2,41 |
| Kompiuteris 4 | 3036 | 479 | 1,15 | 1,08 |
| Kompiuteris 5 | 1163 | 40 | 1,90 | 1,87 |
| Kompiuteris 6 | 1311 | 29 | 6,49 | 6,54 |

Iš 3 lentelėje pateiktų duomenų matyti, jog kai kompiuterio procesorius „galingas“, o vaizdo plokštė „silpna“ (kompiuteris 1, kompiuteris 3, kompiuteris 5, kompiuteris 6), tai *DirectX* technologija yra daug pranašesnė už *OpenGL*

LITERATŪRA

CAI, J.; LIU, Q.; YU, X. (2011). Research and realization of shadowgraph performance technology based on *DirectX*. In: *Proceedings of International Conference on Graphic and Image Processing (ICGIP 2011)*.

Comparison (2013). Comparison of *OpenGL* and *Direct3D*. In: *Wikipedia, the free encyclopedia* [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Comparison_of_OpenGL_and_Direct3D>.

COM (2013). Component Object Model. In: *Wikipedia, the free encyclopedia* [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Component_Object_Model>.

CPU (2013). CPU Benchmarks – *CPU Test Information* [žiūrėta 2013 m. gegužės 17 d.]. Prieiga

nepriklausomai nuo programavimo kalbos. Kitais atvejais, kai vaizdo plokštės yra galingesnės, *DirectX* technologijos pranašumas sumažėja. Tačiau, norint tiksliau iširti šias tendencijas, reikėtų atlikti daugiau eksperimentų skirtinguose kompiuteriuose.

Išvados

Straipsnyje palygintos ir eksperimentiškai iširtos dvi populiariausios trimačių vaizdų vizualizavimo technologijos – *DirectX* ir *OpenGL* naudojant C# ir C++ programavimo kalbas. Atlikti minėtų technologijų spartos eksperimentiniai tyrimai bei lyginamoji analizė leido padaryti tokias išvadas:

- *DirectX* technologija trimačius vaizdus vizualizuoja sparčiau nei *OpenGL*, vidutiniškai 1,9 karto.
- Sudėtingėjant vizualizavimo uždaviniui, *DirectX* technologijos sparta krinta greičiau nei *OpenGL* technologijos ir skirtumas tarp nagrinėtų technologijų spartų sumažėja.
- *DirectX* technologija geriau išnaudoja kompiuterio centrinio procesoriaus ir vaizdo plokštės išteklius, ypač atliekant nesudėtingas vizualizacijas.
- Vizualizuojant trimačius vaizdus *DirectX* technologija, C# programavimo kalba yra pranašesnė už C++ kalbą. *OpenGL* technologijos taikyme programavimo kalbos pranašumo negalima nustatyti.

per internetą: <http://www.cpubenchmark.net/cpu_test_info.html>.

DirectX Graphics (2013). *DirectX Graphics and Gaming*. Windows Dev Center [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <[http://msdn.microsoft.com/en-us/library/windows/desktop/ee663274\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/desktop/ee663274(v=vs.85).aspx)>.

DirectX (2013). In: *Wikipedia, the free encyclopedia* [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <<http://lt.wikipedia.org/wiki/DirectX>>.

Download (2013). *Download Center – Download DirectX Software Development Kit from Official Microsoft Download Center* [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <<http://www.microsoft.com/en-us/download/details.aspx?id=6812>>.

Freeglut (2013). *The freeglut Project* : About [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <<http://freeglut.sourceforge.net/>>.

GUO, P.; CHENG, Y.; LIU, Z. J. (2012). Design and Implementation of 3D Graphics Rendering Engine with *DirectX*. In: *Communications and Information Processing*. Springer.

HILL, F. S.; KELLEY, S. M. (2007). *Computer Graphics Using OpenGL*, 3/E. Prentice Hall.

KOCHAN, S. G. (2011). *Programming in Objective-C*. Addison-Wesley Professional

LUNA, F. D. (2006). *Introduction to 3D Game Programming with DirectX 9.0c: A Shader Approach*. Plano, Tex.: Wordware Pub.

LUNA, F. D. (2008). *Introduction to 3D Game Programming With DirectX 10*. Plano, Tex.: Wordware Pub.

MAŽUTIS, T. (2009). *Trimatės grafikos elementų panaudojimas vizualizacijos sistemose*: Magistro darbas.

Microsoft DirectX3D (2013). In: *Wikipedia*, the free encyclopedia [interaktyvus] [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <http://en.wikipedia.org/wiki/Microsoft_Direct3D>.

Mono (2013). *Download Mono* [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <<http://www.gomono.com/mono-downloads/download.html>>.

MULHOLLAND, A.; HAKALA, T. (2004). *Programming Multiplayer Games*. Wordware Publishing, Inc.

OpenGL – The Industry Standard for High Performance Graphics. The Industry's Foundation for High Performance Graphics [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <<http://www.OpenGL.org/>>.

OpenGL (2013). In: *Wikipedia*, the free encyclopedia [interaktyvus] [žiūrėta 2013 m. gegužės 28 d.]. Prieiga per internetą: <<http://en.wikipedia.org/wiki/OpenGL>>.

OpenGL v DirectX [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <<http://www.blitzbasic.com/Community/posts.php?topic=53219>>.

PassMark (2013). *PassMark Software – Video Card Benchmarks – Video Card Test Information* [žiūrėta 2013 m. gegužės 17 d.]. Prieiga per internetą: <http://www.videocardbenchmark.net/gpu_test_info.html>.

SEBASIAN, A. (2012). *Valve: OpenGL is faster than DirectX – even on Windows* [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <<http://www.extremetech.com/gaming/133824-valve-OpenGL-is-faster-than-DirectX-even-on-windows>>.

SHERROD, A.; JONES, W. (2011). *Beginning DirectX 11 Game Programming*. Course Technology Press, Boston, MA, United States.

SHEINER, D.; SELLERS, G.; KESSENICH, J. M.; LINCEA-KANE, B. M. (2013). *OpenGL programming guide: the official guide to learning OpenGL, versions 4.1*. Upper Saddle River, NJ: Addison-Wesley.

TIOBE Software – Coding Standard Company. In: *TIOBE Programming Community Index for April 2013* [žiūrėta 2013 m. gegužės 17 d.]. Prieiga per internetą: <<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>>.

What is COM? – COM: Component Object Model technologies [žiūrėta 2013 m. liepos 10 d.]. Prieiga per internetą: <<http://www.microsoft.com/com/default.mspx>>.

WRIGHT, Jr. R. S.; HAEMEL, N. S.; SELLERS, G.; LIPCHAK, B. (2010). *OpenGL SuperBible: comprehensive tutorial and reference*. Addison-Wesley.

ZHANG, P.; WANG, H. Y.; LI, Y. G.; MAO, S. F.; DING, Z. J. (2012). Monte Carlo simulation of secondary electron images for real sample structures in scanning electron microscopy. *Scanning*, vol. 34(3), p. 145–150.

COMPARISON OF OPENGL AND DIRECTX TECHNOLOGIES IN 3D PROGRAMMING

Ernestas Filatovas, Grigorijus Melničenko, Vytautas Jakštys

Summary

In this study, three-dimensional rendering technologies *DirectX* and *OpenGL* were reviewed. A comprehensive survey, evaluation of advantages and disadvantages, and a comparative analysis of these technologies have been made.

By applying C# and C++ programming languages, the software has been developed for testing the speed of *DirectX* and *OpenGL* technologies. A relatively large number of experiments were conducted in order to determine the speed of these technologies. The experiments included visualization of basic and

complex shapes movements, and different depth of shapes.

The experimental investigations were performed in Windows and they have revealed that the speed of the *DirectX* technology is higher. This factor is the most significant one in three-dimensional rendering, especially in the development of computer games. The *DirectX* technology was proved to be fastest while programming in C#. Moreover, the experimental investigations have indicated that the *DirectX* technology uses CPU resources more efficiently.