

Informatyka praktyczna. Wprowadzenie

Aleksander Denisiuk
Uniwersytet Warmińsko-Mazurski
Olsztyn, ul. Słoneczna 54
aleksander.denisiuk@uwm.edu.pl

Wprowadzenie

Wprowadzenie

Najnowsza wersja tego dokumentu dostępna jest pod adresem

<http://wmii.uwm.edu.pl/~denisjuk/uwm>

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

Wprowadzenie

Początek OpenGL

Wprowadzenie

❖ Początek OpenGL

- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- 1981: Silicon Graphics
- Koniec lat 1980: IRIS GL
(Integrated Raster Imaging System
Graphical Library) — proprietarna
- 1992: OpenGL (Open Graphics
Library) — otwarta



SiliconGraphics
Computer Systems



Elastyczność

Wprowadzenie

❖ Początek OpenGL

- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- Brak jekiegokolwiek kodu, sama specyfikacja
 - ◆ każdy producent (sprzętu jak i oprogramowania) może implemetować na swojej platformie
- Rozszerzenia (NV_, AGL_)
 - ◆ brak standardowego mechanizmu rozszerzeń
 - ◆ szczególny problem na Windows, gdzie nagłówki OpenGL są nadal w wersji 1.1

Otwartość

Wprowadzenie

❖ Początek OpenGL

- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- 1992: ARB (OpenGL Architecture Review Board)
 - ◆ testy zgodności implementacji
- Wiodący standard grafiki w czasie rzeczywistym
 - ◆ jedyny wieloplatformowy

OpenGL a Windows

Wprowadzenie

❖ Początek OpenGL

❖ OpenGL a Windows

❖ Ewolucja kart graficznych

❖ VAO

❖ Shadery

❖ Rozwój OpenGL

- 1993: Windows NT: rywal Uniksowi w sieci

- ◆ brak biblioteki graficznej
- ◆ decyzja zaimplementować OpenGL



- 1994: Windows NT 3.5, implementacja OpenGL

- ◆ prowizoryczna
- ◆ brak optymalizacji i przyśpieszenia sprzętowego
- ◆ Ostrzeżenie **KB121282**:
“OpenGL Screen Savers May Degrade Server Performance”

DirectX

Wprowadzenie

❖ Początek OpenGL

❖ OpenGL a Windows

❖ Ewolucja kart graficznych

❖ VAO

❖ Shadery

❖ Rozwój OpenGL

- Microsoft: własne 3W API dla gier pod Windows
 - ◆ WinG (translacja poleceń do GDI)
 - ◆ 1995: Microsoft nabywa RenderMorphics
 - Reality Lab



- ◆ 1996: DirectX
 - DirectDraw
 - DirectInput
 - DirectPlay
 - DirectSound
 - Direct3D

- Direct3D: bardzo niewygonna
- implementacja OpenGL 1.1 w Windows 95 oraz NT 4.0

Wojna API

Wprowadzenie

- ❖ Początek OpenGL

- ❖ OpenGL a Windows

- ❖ Ewolucja kart graficznych

- ❖ VAO

- ❖ Shadery

- ❖ Rozwój OpenGL

- Microsoft: OpenGL do grafiki „profesjonalnej” (CAD). Do programowania gier lepiej nadaje się DirectX



- 1996: John Carmack (id Software) — port Quake na OpenGL dla Windows, **porównanie OpenGL a Direct3D**
- 1997: Alex St. John — **odpowieź w obronie Direct3D**

Wojna API, cd

Wprowadzenie

❖ Początek OpenGL

❖ OpenGL a Windows

❖ Ewolucja kart graficznych

❖ VAO

❖ Shadery

❖ Rozwój OpenGL

- Microsoft: OpenGL jest profesjonalną biblioteką i nie będzie wspierana przez “zwykłe” karty graficzne
- SGI: Odpowiedź: porównanie OpenGL a Direct3D pod kątem inżynierskim, nie marketologowym
- Direct3D w wersji 5 został porównywalnym z OpenGL

Rozgromienie sterowników

Wprowadzenie

❖ Początek OpenGL

❖ OpenGL a Windows

❖ Ewolucja kart graficznych

❖ VAO

❖ Shadery

❖ Rozwój OpenGL

- Sterowniki implementujące OpenGL na Windows NT zostały oparte o MCD (Mini-Client Driver)
- Microsoft nie pozwoliła na licencjonowanie MCD na Windows 95
- SGI: Installable Client Driver (ICD)
- Producenci sprzętu: nowe sterowniki
- Producenci gier: gry pod OpenGL

Ewolucja kart graficznych

Wprowadzenie

❖ Początek OpenGL

❖ OpenGL a Windows

❖ Ewolucja kart graficznych

❖ VAO

❖ Shadery

❖ Rozwój OpenGL

- Koniec lat 1990: OpenGL standardem 3W grafiki
 - ✦ CAD, Quake 2, Unreal, Half-Life
- Pierwsze dedykowane 3W karty graficzne
 - ✦ ATI 3D Rage, S3 ViRGE
 - ✦ Voodoo Graphics (3Dfx Interactive)
 - Glide API
 - ✦ NVIDIA
 - GeForce 256
 - GPU (Graphics Processing Unit)
 - T&L (Transform & Lighting)
- Początek 2000: NVIDIA GeForce 2, ATI Radeon 7000
 - ✦ OpenGL, Direct3D



Zmiana paradygmatu

Wprowadzenie

- ❖ Początek OpenGL

- ❖ OpenGL a Windows

- ❖ Ewolucja kart graficznych

- ❖ VAO

- ❖ Shadery

- ❖ Rozwój OpenGL

- Na początku lat 2000 wydajność GPU wzrasta wykładniczo
- CPU nie jest potrzebny do renderowania 3W grafiki w czasie rzeczywistym
- CPU nawet jest *wązkim gardłem* w procesie renderowania
 - ◆ potrzebne są nowe metody, żeby obejść CPU

Obiekt tablicy wierzchołków

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- *Immediate mode*: program przekazuje ciąg poleceń do GPU
- *Nowa metoda*: obiekt bufora
 - ◆ tworzony i przechowywany w pamięci GPU
 - ◆ OpenGL: *VAO (Vertex Array Object)*, *VBO (Vertex Buffer Object)*
 - ◆ Direct3D: *Vertex Buffer*

Shadery

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- 2000: Microsoft, Direct3D 8.0
 - ◆ vertex shaders
 - ◆ pixel shaders
 - ◆ Asembler GPU
- 2003: Direct3D 9.0
 - ◆ High-Level Shader Language (HLSL)

Stagnacja OpenGL

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- Brak wsparcia shaderów
- Dopiero w 2004: OpenGL 2.0, OpenGL Shading Language (GLSL)
- 2004–2006: dominacja Direct3D 9.0, zaledwie kilka gier na OpenGL
- 2005: Xbox 360 ze wsparciem Direct3D 9.0
- żadnych wiadomości od ARB

Stagnacja OpenGL

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**

- 2006:
 - ◆ OpenGL 2.1
 - ◆ Direct3D 10.0 (razem z Windows Vista)
- Zmiana kierunku rozwoju sprzętu:
 - ◆ eliminacja immediate mode, ustalonych funkcji
 - ◆ co raz bardziej programowalne GPU
- ARB, SGI: brak odpowiedzi

Nowy OpenGL

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- 2006, SIGGRAPH: zarządzać rozwojem OpenGL będzie Khronos Group
 - ◆ konsorcjum producentów (sprzętu i oprogramowania)
 - ◆ rozwój otwartych standardów
 - OpenGL
 - COLLADA

Longs Peak, Mt. Evans

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**

- Miały zostać opublikowane w 2007: latem oraz w październiku
 - ✦ dociągnąć do Direct3D 10.0
 - wyeliminować immediate mode
 - działać tylko na buforach i shaderach
 - Technical Sub-Groups
 - ✦ Longs Peak
 - kompatybilna z ówczesnym sprzętem
 - kompatybilna wstecz ze starym OpenGL
 - ✦ Mt. Evans
 - złamie kompatybilność wstecz
 - będzie podstawą dla przyszłości
- 30 października 2007: trzeba poczekać

OpenGL 3.0

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- Lipiec 2008
- Nie było to *Longs Peak*
 - ◆ został immediate mode
 - ◆ nie został włączony w specyfikację nowy model obiektowy
 - ◆ nie było planów, co włączyć w przyszłe wersje
 - ◆ trochę nowych możliwości
 - ◆ deprecation model
 - wszystkie metody, związane z immediate mode zostały zaznaczone jako przestarzałe
 - pełna kompatybilność ze starymi metodami

Ciąg dalszy

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**

- Liczne protesty społeczności
- Programiści Windows zaczęli porzucać OpenGL na rzecz Direct3D
- Czyżby wojna API została przegrana?

Ciąg dalszy

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- Okazało się, że nie jest tak źle :-)
 - ✦ OpenGL 3.0 miał cechy, których nie było w Direct3D 10.0:
 - dostęp do nowych funkcji w Windows XP
- Marzec 2009: OpenGL 3.1
 - ✦ eliminacja immediate mode
- Czerwiec 2009: OpenGL 3.2
 - ✦ Geometry Shaders
 - ✦ dorównanie z Direct3D 10.0

Nowy OpenGL. Kontekst

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- *Context* — obiekt pośredniczący pomiędzy programem a urządzeniem OpenGL.
Przy tworzeniu kontekstu:
 - ◆ główny numer wersji
 - ◆ drugorzędny numer wersji
 - ◆ opcjonalne parametry
- Każda wersja OpenGL (powyżej 3.0) ma część funkcji *deprecated*, które mogą zostać usunięte w następnych wersjach
 - ◆ Core Profile flag
 - ◆ Compatibility Profile flag
 - ◆ Forward Compatible flag
 - ◆ Debug flag

OpenGL 4.0

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- 11 marca 2010
 - ◆ OpenGL 4.0
 - współcześnie GPU
 - Direct3D 11
 - Teselacja
 - 64-bitowa dokładność obliczeń
 - ◆ OpenGL 3.3
 - kompatybilna ze starym sprzętem
 - jak najwięcej możliwości OpenGL 4.0
- W przykładach będzie używany OpenGL nowszy niż 3.3, *Core profile*
 - ◆ w szczególności, żadnych `glBegin`, `glEnd`, `glVertex3f`, `glColor3f`
 - ◆ wszystkie przykłady pójdą na 3.3

OpenGL 4.1–4.6, Vulkan

Wprowadzenie

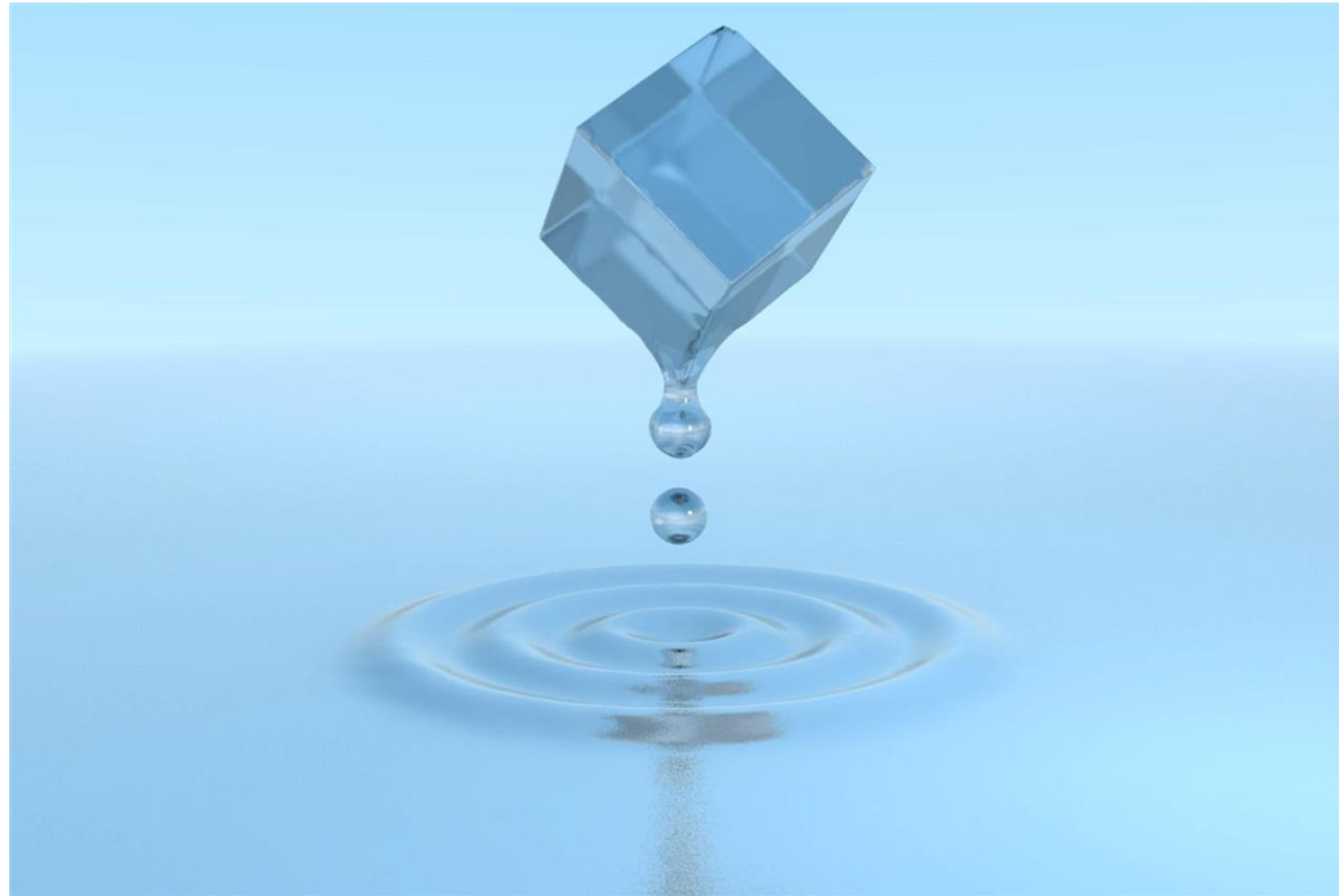
- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**

- 26 czerwca 2010 — OpenGL 4.1
 - ◆ nowa wersja GLSL
 - ◆ zupełna zgodność z OpenGL ES
- 8 sierpnia 2011 — OpenGL 4.2
- 6 sierpnia 2012 — OpenGL 4.3
- 22 lipca 2013 — OpenGL 4.4
- 11 sierpnia 2014 — OpenGL 4.5
- 16 lutego 2016 — Vulkan 1.0
- 31 lipca 2017 — OpenGL 4.6
- 3 grudnia 2024 — Vulkan 1.4
- 28 sierpnia 2026 — Vulkan 1.4.361

OpenGL Przykład

Wprowadzenie

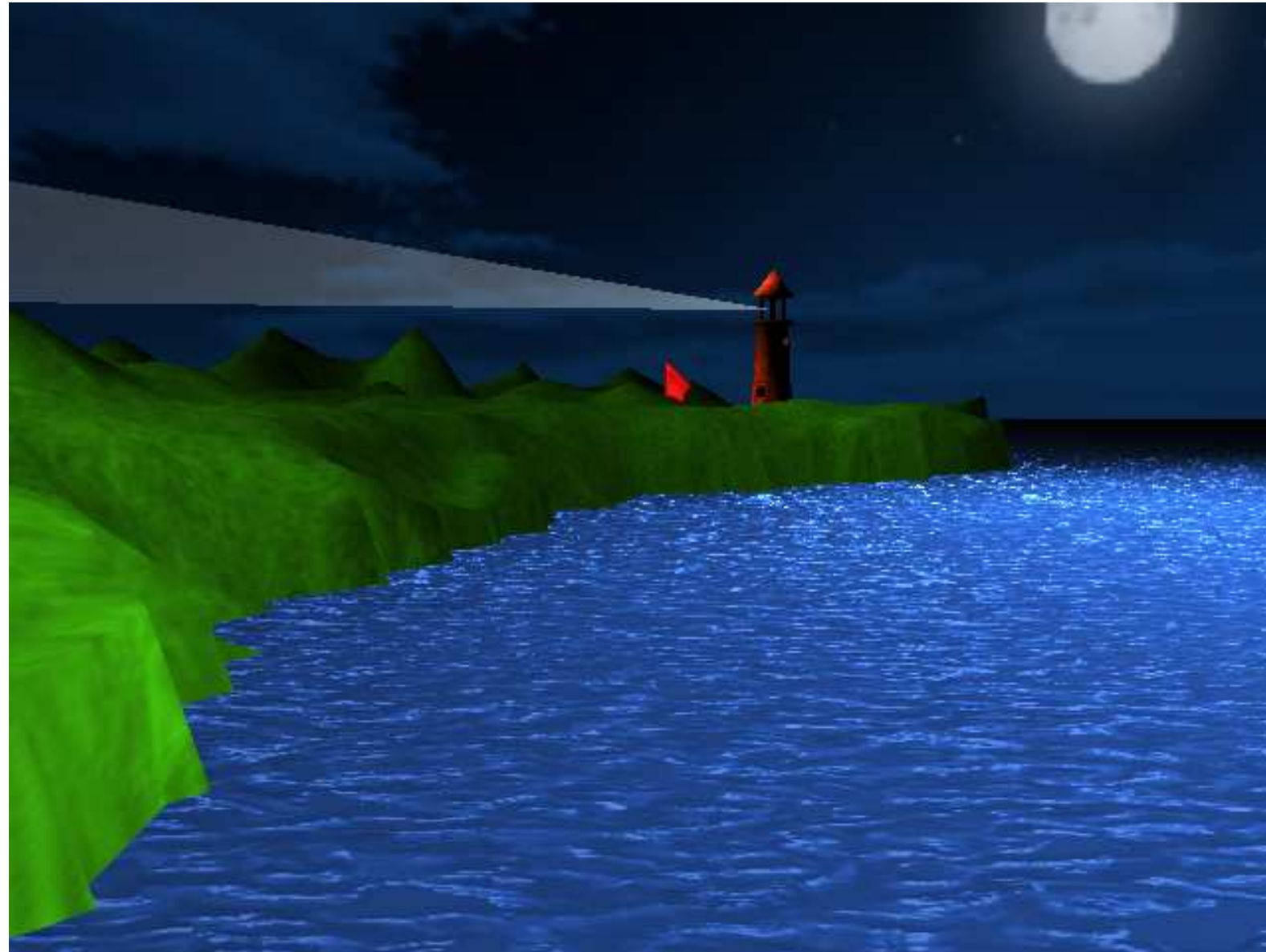
- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**



OpenGL Przykład studencki

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**



OpenGL

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**

- Wieloplatformowość
 - ◆ Windows, Linux, MacOS
 - Windows, Linux, MacOS
 - iPhone, Android
 - ◆ OpenGL ES
- WebGL

Vulkan

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ Rozwój OpenGL

- Bytecode dla shaderów, kompilacja w czasie kompilacji programu
- Wieloplatformowość
- Lepsze wykorzystanie równoległości
- Zmniejszenie obciążenia CPU
- Ray tracing — niezależne API
- Kodowanie/dekodowanie wideo
- Demo

Wymagania sprzętowe. OpenGL 4

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**

- NVIDIA GPU — GeForce GTX 400
- AMD/ATI GPU — Radeon HD 5000
- najnowsze sterowniki
- prawie każda karta wyprodukowana po roku 2006

Biblioteki wspomagające

Wprowadzenie

- ❖ Początek OpenGL
- ❖ OpenGL a Windows
- ❖ Ewolucja kart graficznych
- ❖ VAO
- ❖ Shadery
- ❖ **Rozwój OpenGL**

- GLFW
- GLEW