

# OpenGL 4.0 w systemie Windows.

Dawid Zarzycki

# Plan prezentacji

- Czym jest OpenGL
- Ewolucja biblioteki
- Mechanizm rozszerzeń
- OpenGL według Microsoft
- Tworzenie kontekstu renderingu

# Czym jest OpenGL?

- OpenGL zdefiniowano jako „interfejs programowy dla sprzętu graficznego”.
- Niskopoziomowe API do renderowania.
- Praca z tą biblioteką nie polega na wydawaniu poleceń typu „Narysuj ten model w tym miejscu”.

# Ewolucja biblioteki.

- Prekursorem była biblioteka IRIS GL firmy Silicon Graphics.
- Firma SGI chcąc sprzedawać więcej swoich komputerów potrzebowała większej ilości oprogramowania działającego na ich sprzęcie.
- Inni producenci również zdali sobie z tego sprawę i w efekcie narodziła się OpenGL ARB (ang. *Architecture Review Board*).

# Ewolucja biblioteki.

- Pierwszego lipca 1992 roku zaprezentowano specyfikację OpenGL w wersji 1.0.
- Z biegiem czasu rada ARB rozrastała się o wielu kolejnych członków, z których większa część to producenci sprzętu.
- W 2006 roku SGI przekazują pieczę nad standardem firmie Khronos Group.

# Ewolucja biblioteki.

- Ostatnio nowe wersje specyfikacji OpenGL są publikowane częściej niż raz do roku.
- W 2010 roku przedstawiono wersje 3.3 i 4.0.
- Aktualnie najnowsza wersja to 4.2.

# Mechanizm rozszerzeń

- Producenci mogą konkurować ze sobą nie tylko na polu wydajności i jakości obrazu, ale także dodawać nowe funkcjonalności do swoich implementacji.
- Nie jest niczym niezwykłym, że rozszerzenia jednego producenta z czasem obsługiwane są przez innych producentów.

# Mechanizm rozszerzeń

- Często zdarza się, że wszyscy zgadzają się, że dane rozszerzenie jest na tyle ciekawe, że zaimplementują je wszyscy (*EXT\_*).
- Istnieją też rozszerzenia zatwierdzone przez ARB (*ARB\_*). Tego rodzaju rozszerzenia najczęściej oznaczają ostatni etap, zanim wejdą do standardu.



# Funkcje wycofane

- Standard OpenGL cały czas ewoluował, do każdego kolejnego wydania dodawano nowe funkcje. W tym czasie niczego nie usuwano.
- Jednak kontynuowanie tego podejścia stało się w końcu nierealne.
- Architektura komputerów i GPU znacznie się zmieniła.
- Sztuczki wydajnościowe odchodzą do lamusa, dlatego część funkcji stała się przestarzała.

# Funkcje wycofane

- W ARB podjęto decyzje, że po raz pierwszy w historii biblioteki zostanie z niej usunięta część „przestarzałego balastu”.
- W nowych aplikacjach pisanych w OpenGL 3.0 stosowanie odrzuconych funkcji było zabronione.
- Wówczas wydawało się to dobrym pomysłem...

# OpenGL 3.0

- Jak to bywa z planami, często coś je krzyżuję.
- W skład grupy ARB wchodzi producent sprzętu graficznego, a ci mają swoich klientów, z którymi trzeba się liczyć.
- Porozumienie = funkcję przestarzałe – kiedyś je usuniemy (mówiono, że w GL 3.1).

# OpenGL 3.1

- W przypadku 3.1 staliśmy się świadkami dzielenia włosa na czworo.
- Rzeczywiście usunięto z rdzennej specyfikacji wszystkie przestarzałe elementy, ale dodano nowe rozszerzenie *GL\_ARB\_compatibility*.
- Rdzeń + wszystkie stare funkcję, które obiecaliśmy usunąć, a których nie usuneliśmy.

# OpenGL 3.2

- W wersji 3.2 sytuacje nieco naprawiono. Usunięto nieszczęsne rozszerzenie i zamiast tego podzielono specyfikację na dwa profile – rdzenny i zgodnościowy.
- Czemu ciężko usunąć przestarzałe elementy?
- Profil zgodnościowy jeszcze długo będzie nam towarzyszył.

# OpenGL 4.x

- Dodawane są nowe rozszerzenia, nowe funkcję, typy wyliczeniowe etc.
- Dalej mamy dwa profile.

# Profil rdzenny.

- Profil rdzenny w swojej najprostszej skróconej postaci to nic więcej, jak tylko „programy cieniujące”. Są one potrzebne, aby zrobić cokolwiek.
- Nie ma w nim wbudowanego modelu oświetlenia, wygodnych stosów macierzy etc.

# OpenGL według Microsoft

- OpenGL pojawiła się na platformie Win32 wraz z wydaniem systemu Windows NT 3.5.
- Microsoft dostarcza ogólną programową implementację GL 1.1.
- Microsoft nie bierze udziału w rozwoju OpenGL już od wielu lat.
- OpenGL32.dll – implementuję tylko wersje 1.1.



# OpenGL według Microsoft

- Sterowniki nie mogą dodawać nowej funkcjonalności do biblioteki `opengl32.dll`, dlatego wykorzystano do tego mechanizm rozszerzeń.
- Rozszerzenia `WGL_`.

# Windows Vista/7

- OpenGL działa podobnie jak w starszych wersjach.
- Jednak w Vista/7 ostateczny wynik prezentowany jest użytkownikowi przy użyciu tzw. Systemu DCE (*Desktop Composition Engine*).
- Wcześniej każde okno renderowało do pikseli dostępnego pulpitu.
- Teraz okno renderuje na powierzchni, której jest przekazywany nowy komponent systemu operacyjnego o nazwie *Desktop Window Manager* (DWM).

# Ładowanie rozszerzeń

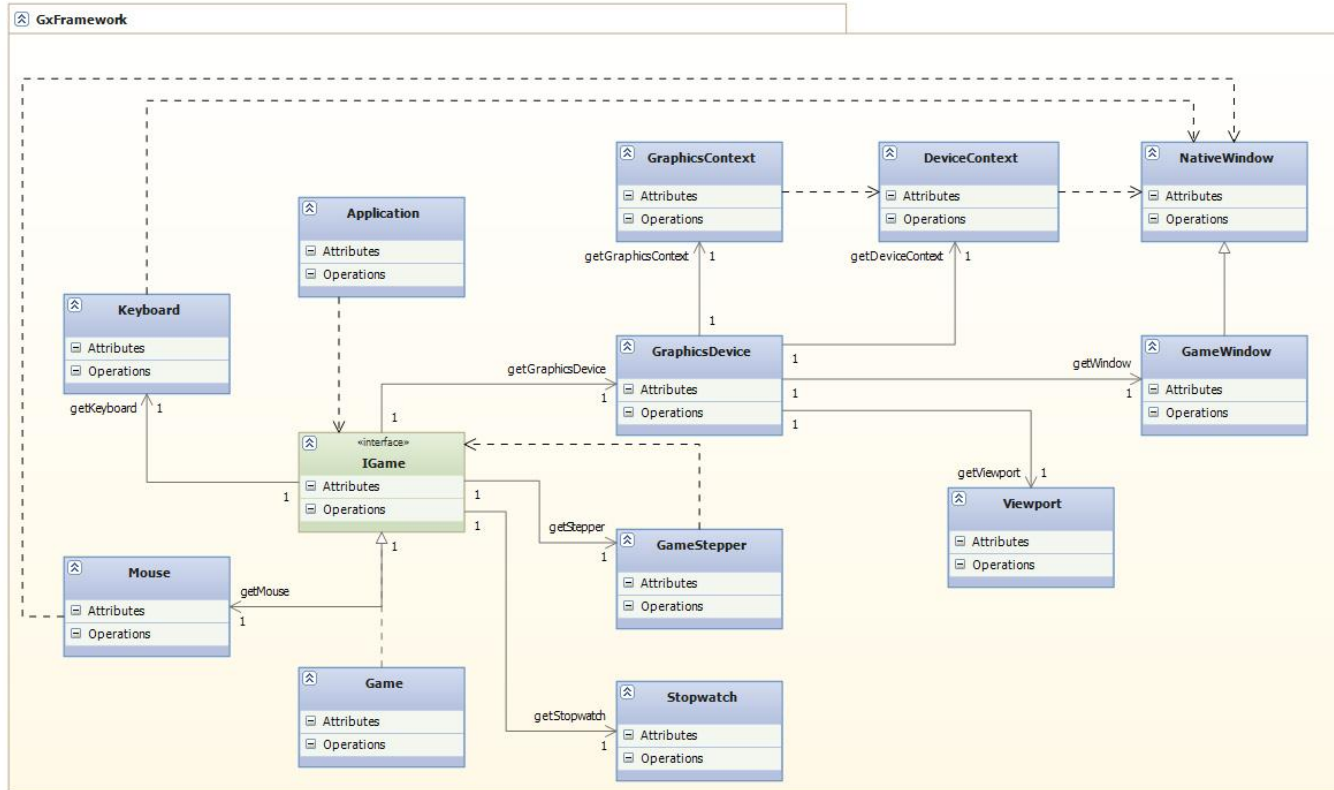
- *PROC wglGetProcAddress(LPSTR procName);*
- Wersja funkcji jest zależna od aktualnego kontekstu renderingu.
- Skąd brać prototypy funkcji?
  - <gl/gl.h> - przestarzałe
  - gl3.h, glext.h, wglext.h – dobre 😊
  - GLEW, GL3W - zautomatyzowane

# Tworzenie kontekstu

- Aby załadować funkcję potrzebne do stworzenia kontekstu, musimy... stworzyć kontekst.
- Pokazane w praktyce 😊.



# GxFramework



# Pytania na kolokwium

- Dlaczego musimy tworzyć dwa okna aby stworzyć kontekst renderingu w nowszych wersjach OpenGL?
- Czym jest biblioteka OpenGL i kto był prekursorem tej biblioteki?

Dziękuję za uwagę 😊