

Wykłady zlotowe

wersja wstępna (fixme)

Zlot Programistów Delphi 2008
Kraków, Akademia Górniczo-Hutnicza, 14 - 16 marca 2008
<http://www.delphi.org.pl>

Mikołaj "Virion" Milewski
email: wyklady@delphi.org.pl

Wersja: 0.3.0 build 44 (10 lutego 2008)

Spis treści

Zmiany w curriculum zlotowym 2008 (changelog)	2
1 Słowo wstępne na temat wykładów	3
O Ojcu Derektorze d/s Wykładów	3
2 Wykłady	4
Plan wykładów	4
Bogdan „Bodek” Polak: Jaką platformę wybrać – Win32 czy .NET?	5
Andrzej „Sir Daban” Dąbrowski: Aplikacje wielojęzyczne w Delphi	6
Wojciech „Wirus” Dziedzic : Jak szybko przejść od projektu do gotowej aplikacji bazodanowej?	6
Daniel „Wloochacz” Grabowski : AnyDAC rządzi!	7
Przemysław „Zeman” Gula: Dziedziczenie wielokrotne w Delphi nie gryzie	9
Maksymilian „Maks” Leszczyński: Jak tworzyć udane oprogramowanie?	10
Marcin „Frodo2000” Molak: DB2 9.5 pureXML – przetwarzanie dokumentów XML od kuchni	11
Wojciech „Spook” Sura : Renderowanie grafiki 3D	12

Zmiany w curriculum zlotowym 2008 (changelog)

- wersja 0.3 (10 lutego 2008):
 - Dodany wykład Bogdana „Bodka” Polaka o RAD Studio 2007 i Win32 vs .NET
 - Dodany wykład Maksymiliana „Maksa” Leszczyńskiego o tworzeniu udanych programów
 - Dodany wykład Daniela „Wloochacza” Grabowskiego o AnyDAC
 - Dodany wykład Wojtka „Wirusa” Dziedzica o produkcji aplikacji bazodanowych
 - Dodany wykład Wojtka „Spooka” Sury o grafice 3D
 - Poprawki redakcyjne
- wersja 0.2 (4 lutego 2008):
 - Dodany wykład Przemysława „Zemana” Guli o dziedziczeniu wielokrotnym
 - Dodany wykład Andrzeja „Sir Dabana” Dąbrowskiego o aplikacjach wielojęzykowych
 - Poprawki redakcyjne
- wersja 0.1 (28 stycznia 2008):
 - Dodany wykład Marcina „Frodo2000” Molaka o XMLu w DB2
 - Usunięte lorem ipsum („wykład” Jana Kowalskiego)
- wersje 0.0.* (ok. połowy stycznia):
 - Notka biograficzna o Ojcu Derektorze d/s Wykładów, Mikołaju „Virionie” Milewskim
 - Lorem ipsum: „wykład” Jana Kowalskiego
 - Krótka walka z \LaTeX owymi źródłami z poprzedniego roku

1 Słowo wstępne na temat wykładów

W dniach 14-16 marca 2008 (piątek-niedziela) na Akademii Górniczo-Hutniczej w Krakowie odbędzie się doroczny Zlot Programistów Delphi.

W ramach Zlotu organizowane są wykłady - zajmą całą sobotę. Stroną wykładową opiekuje się Ojciec Derektor d/s Wykładów, czyli ja. Planując wykłady staramy się z jednej strony o to, żeby były one możliwie "techniczne" i prowadzone przez doświadczonych praktyków. Skądinąd wychodzimy z założenia, że "prozę programistycznego życia" zlotowicze mają na codzień - z tego względu pewna część wykładów wykracza poza zagadnienia ściśle związane z Delphi albo je uzupełnia.

Wykłady odbędą się w dwóch ścieżkach po 4 wykłady. Każdy wykład zajmie 1.5 godziny. Po każdym wykładzie będzie dodatkowe 30 minut czasu na dyskusje i pytania do wykładowcy - osoby zainteresowane mogą w tym czasie porozmawiać z wykładowcą, osoby niezainteresowane mogą skorzystać z przerwy.

Materiały wykładowe zostaną udostępnione na stronach pozlotowych w serwisie delphi.org.pl :

<http://www.delphi.org.pl>

Pozdrawiam serdecznie i do zobaczenia na Zlocie !

Mikołaj "Virion" Milewski
Ojciec Derektor d/s Wykładów
email: wyklady@delphi.org.pl

PS: Jeśli zauważycie jakieś literówki, połamane odnośniki URL czy inne błędy, bardzo proszę o informację mailem na mój adres - z góry serdecznie dziękuję. W mailach proszę podawać wersję dokumentu (jest na samym początku), numer strony i jakieś informacje o umiejscowieniu i rodzaju błędu.

O Ojcu Derektorze d/s Wykładów

Ojcem Derektorem d/s Wykładów A.D. 2008 jest po raz pierwszy nowa osoba, Mikołaj „Virion” Milewski. Mikołaj Milewski ma 22 lata, pracuje w słujskiej firmie **DGB IT**. Jego ulubiony aspekt programowania to niuansy tworzenia komponentów, okazjonalnie tworzy również strony WWW. W Zloty Programistów Delphi angażuje się od dwóch lat, w roku 2007 wygłosił nagrodzony wykład. Poza pracą i narzeczoną nie ma już czasu prawie na nic. Nadal jeździ rowerem i potem nałogowo śpi, przez co notorycznie spóźnia się do pracy ...

2 Wykłady

Plan wykładów

Plan wykładów przedstawiamy w poniższych tabelach :

Rozpoczęcie	Zakończenie	Ścieżka wykładowa I	Ścieżka wykładowa II
9:30	11:00		2 wykłady
11:00	11:30		Pytania / przerwa
11:30	13:00		2 wykłady
13:00	13:30		Pytania / przerwa
13:30	15:00		Przerwa obiadowa
15:00	16:30		2 wykłady
16:30	17:00		Pytania / przerwa
17:00	18:30		2 wykłady
18:30	19:00		Pytania

Układ wykładów i warsztatów w ścieżkach – **fixme WERSJA WSTĘPNA, MOŻE SIĘ ZMIENIĆ!**

Ścieżka I (wykłady w sali dużej nr 3, parter, bud. A4):

Rozpoczęcie	Zakończenie	Wykład
9:30	11:00	Imię „fixme” Nazwisko Temat fixme
11:00	11:30	Pytania / przerwa
11:30	13:00	Imię „fixme” Nazwisko Temat fixme
13:00	13:30	Pytania / przerwa
13:30	15:00	Przerwa obiadowa
15:00	16:30	Imię „fixme” Nazwisko Temat fixme
16:30	17:00	Pytania / przerwa
17:00	18:30	Imię „fixme” Nazwisko Temat fixme
18:30	19:00	Pytania

Ścieżka II (wykłady w sali mniejszej nr 69z, piętro I, bud. A4):

Rozpoczęcie	Zakończenie	Wykład
9:30	11:00	Imię „fixme” Nazwisko Temat fixme
11:00	11:30	Pytania / przerwa
11:30	13:00	Imię „fixme” Nazwisko Temat fixme
13:00	13:30	Pytania / przerwa
13:30	15:00	Przerwa obiadowa
15:00	16:30	Imię „fixme” Nazwisko Temat fixme
16:30	17:00	Pytania / przerwa
17:00	18:30	Imię „fixme” Nazwisko Temat fixme
18:30	19:00	Pytania

Bogdan „Bodek” Polak: Jaką platformę wybrać – Win32 czy .NET?

Streszczenie

W trakcie wykładu chciałbym podzielić się z programistami moimi doświadczeniami – dlaczego czasami warto wybrać platformę .NET Framework, ale również dlaczego i gdzie nie warto tego robić. Wiemy, że od ponad 5 lat firma Microsoft próbuje udowodnić, że jedyny słuszny wybór to .NET. Z kolei większość programistów tworzących aplikacje graficzne, desktopowe i klient-serwer próbuje udowodnić firmie Microsoft, że platforma ta nie interesuje ich. Jeżeli pragniesz dowiedzieć się w jakich sytuacjach warto wykorzystać platformę .NET oraz jak w tym może pomóc środowisko RAD Studio 2007 to zapraszam na mój wykład.

Konspekt

1. Krótkie wprowadzenie do platformy .NET
 - (a) Kod zarządzany a kod natywny
 - (b) Pseudokod a kompilacja w locie
 - (c) Czym jest .NET Framework 3.5?
 - (d) Różnice między: ASP.NET 2.0, Windows Forms, Windows Presentation Foundation, .NET Compact Framework
2. Kilka wniosków – Win32 czy .NET?
 - (a) Czy warto zostać przy Win32?
 - (b) Kiedy lepiej skorzystać z .NET Framework?
 - (c) VCL.NET – czyli rozwiązanie pośrednie
 - (d) Czego trzeba się nauczyć przy przejściu na platformę .NET?
3. Trochę dłuższe wprowadzenie do RAD Studio 2007
 - (a) Czym się różni RAD Studio od Delphi?
 - (b) Nowości w RAD Studio 2007 przeznaczone dla .NET
 - (c) Różnice między Developer Studio a RAD Studio
4. Blok demonstracyjny (najdłuższy)
 - (a) ASP.NET 2.0
 - (b) Blackfish SQL
 - (c) ECO III
5. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Bogdan jest trenerem i starszym konsultantem BSC Polska, gdzie od ponad trzech lat zajmuje się prowadzeniem szkoleń i wykładów, przygotowaniem prezentacji, prowadzeniem konsultacji i wspomaganiem społeczności programistów. W firmie pełni obowiązki CodeGear Technical Evangelist oraz Borland Technical Consultant na terenie Polski. Wcześniej był programistą, a także współzałożycielem firmy programistycznej oraz aktywnym członkiem społeczności programistów Delphi w Polsce. Ma ponad 15 lat doświadczeń w pracy z narzędziami programistycznymi firmy Borland/CodeGear, w tym wiele lat doświadczeń przy tworzeniu rozwiązań graficznych i bazodanowych, takich jak programy multimedialne, GIS, CRM, rachunkowe i inne.

Andrzej „Sir Daban” Dąbrowski: Aplikacje wielojęzyczne w Delphi

Streszczenie

Na forum dyskusyjnym wielokrotnie padały pytania o tworzenie aplikacji wielojęzycznych, a zwłaszcza z uwzględnieniem kodowania UTF-8. Chciałbym zaprezentować rozwiązanie podstawowe, Translation Manager dostępny od lat w środowisku Delphi. Rozwiązanie to jest wystarczające dla większości tworzonych aplikacji wielojęzycznych. Mam także coś dla lubiących skrajne wyzwania i twierdzących, że Translation Manager to za mało. Dla osób z wysokimi wymaganiami pokażę darmowy, „wszystkomający” moduł do tworzenia wielojęzyczności – GNU GetText. Chcecie mieć możliwość dodawania języków bez kompilacji? Proszę bardzo. Zmiana języka w czasie działania aplikacji – nic trudnego. A może chcecie w pliku tekstowym, notatnikiem dodawać dodatkowe frazy do przetłumaczenia? Pokażę jak tego dokonać, na żywo, z wykorzystaniem obydwu narzędzi oraz komponentów TNT.

Konspekt

1. Ogólne wprowadzenie do wielojęzyczności (pojęcia Unicode, ASCII, itp.)
2. Translation Manager
 - (a) Napisanie krótkiej aplikacji pod asystą tłumacza
 - (b) Utworzenie wersji językowych
 - (c) Przełączanie języków
3. GNU GetText
 - (a) Kilka słów o licencji
 - (b) Napisanie krótkiej aplikacji z GNU GetText
 - (c) Utworzenie wersji językowych
 - (d) Przełączanie języków w trakcie pracy aplikacji
 - (e) Dodawanie tekstów do tłumaczenia za pomocą notatnika, bez ponownej kompilacji aplikacji
4. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Andrzej jest trenerem i konsultantem technicznym BSC Polska, pełni obowiązki CodeGear Technical Evangelist w Polsce. Ukończył Wydział Cybernetyki Wojskowej Akademii Technicznej. Po odejściu z wojska pracował jako programista i kierownik działu programistów. Od dziesięciu lat używa narzędzi IDE Borland/CodeGear, specjalizuje się w aplikacjach bazodanowych.

Wojciech „Wirus” Dziejczyk : Jak szybko przejść od projektu do gotowej aplikacji bazodanowej?

Streszczenie

W dużych aplikacjach bazodanowych istotne jest, aby interfejsy wprowadzania danych były ujednolicone. Oczekiwane jest, że moduły takich systemów będą miały taki sam interfejs użytkownika, ale implementować będą właściwą dla siebie funkcjonalność. Poprawia to jakość systemu, ułatwia pracę użytkownikowi i nam programistom. Liczba godzin jaką poświęcamy na szkolenia jest zdecydowanie mniejsza. Takie interfejsy mogą mieć wbudowaną obsługę praw dostępu, modyfikacji danych, wyszukiwania i inne często spotykane funkcjonalności. Zastosowanie procesu FAST pozwala przyspieszyć wytworzenie aplikacji od 5 do 10 razy w stosunku do wytworzenia jej z użyciem tradycyjnych metod. Trzeba zdać sobie sprawę, że kluczem do sukcesu w tym podejściu jest poprawne wydzielenie dziedziny (np. systemy zarządzające

produkcją, systemy bankowe, systemy wspomagające analizę danych naukowych) oraz bardzo starannie przeprowadzona analiza wymagań systemu. Wykład będzie ilustrowany na przykładzie systemu zarządzania produkcją Baza Instrukcji Technologicznych BIT.

Konspekt

1. Wprowadzenie
 - (a) Pierwsze projekty (zarządzanie dokumentacją technologiczną, drukowanie przepustek)
 - (b) Skąd pomysł na realizację projektu BIT (adaptacja filozofii MAGICa do projektu)?
2. Teoria – proces FAST
3. Praktyczne zastosowanie
 - (a) TFormBrowse - co to jest?
 - (b) Generator kodu
 - (c) Szkielet systemu
 - (d) Omówienie podstawowego obiektu w projekcie TFormBrowse
 - (e) Zmienne globalne wpływające na działanie obiektu TFormBrowse
 - (f) Maksymalna parametryzacja systemu
4. Wady i zalety
 - (a) Projektowanie głównego obiektu TFormBrowse
 - (b) Przyrost kodu w stosunku do rozwoju systemu
 - (c) Stosowanie wypracowanego schematu
 - (d) Minimalna ilość modułów
 - (e) Zastosowanie szkieletu do budowy innych systemów
 - (f) Zastosowanie wspólnego kodu dla wszystkich systemów
 - (g) Podział prac programistów (programowanie obiektu TFormBrowse, programowanie logiki biznesowej)
 - (h) Oszczędność czasu
5. Prezentacja systemu BIT – omówienie podstawowych funkcji
6. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Wojciech Dziedzic od dwunastu lat pisze oprogramowanie. Od dziesięciu zajmuje się zawodowo projektowaniem i programowaniem systemów zarządzania produkcją. W pracy wykorzystuje platformę MAGIC, ostatnio również Delphi. Poza pracą relaksuje się uprawiając tak zwane kajakarstwo zwałkowe.

Daniel „Wloochacz” Grabowski : AnyDAC rządzi!

Streszczenie

AnyDAC jest zestawem komponentów dla CodeGear Delphi i Free Pascal Compiler. Dostarcza szybkich i natywnych metod dostępu do różnych baz danych, wliczając w to Microsoft SQL Server, Oracle, IBM DB2, MySQL, Sybase, Firebird i inne. Biblioteka została zoptymalizowana do bezpośredniego dostępu do odpowiednich systemów baz danych w celu pominięcia nadmiarowych warstw dostępowych, takich jak ADO, OLE DB czy ODBC. AnyDAC wykorzystuje wyjątkowe zasady programowania, które czynią go jednym z najlepszych na rynku komponentów dostępowych do baz danych dla Delphi.

Konspekt

1. Wprowadzenie
2. Architektura AnyDAC
 - (a) Przegląd
 - (b) Standardowe klasy i interfejsy
 - (c) Warstwa fizycznego dostępu do danych
 - (d) Warstwa Data Adapter
 - (e) Warstwa Data Store
 - (f) Warstwa komponentów niewizualnych
 - (g) Warstwa GUIx
 - (h) Warstwa komponentów wizualnych
 - (i) Interfejsy ogólnego użycia
3. Możliwości i wykorzystanie AnyDAC
 - (a) Definicja i kontrola połączeń z baza danych
 - (b) Connection pooling
 - (c) Obsługa transakcji
 - (d) Wykonywanie i optymalizacja zapytań
 - i. Pobierania danych i optymalizacja
 - A. Komponent TADQuery
 - B. Komponent TADClientDataSet
 - ii. Aktualizacja danych
 - A. Komponent TADCommand
 - B. Komponent TADUpdateSQL
 - C. Komponent TADStoredProcedure
 - D. Obsługa błędów
 - (e) Unikalne możliwości
 - i. Agregacje
 - ii. Makra
 - iii. Filtrowanie i sortowanie po stronie klienta
 - iv. Obsługa pól identity
 - v. Mapowanie typów
 - vi. Obsługa wielu zestawów danych
4. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Daniel „włoochacz” Grabowski, założyciel słupskiej firmy DGB IT, programuje tylko okazjonalnie. Większość czasu spędza na wymyślaniu dziwacznych idei, które następnie z mozołem są implementowane przez jego zespół. Szczęśliwie żonaty i przeschczęśliwy tatuś córki Zuzanny. Jego prawdziwą pasją jest jednak łowiectwo – samotne włóczęgi po lasach nocami ze sztucerem w rękę w celu upolowania grubego zwierzka to jest to, co naprawdę włoochacza interesuje.

Przemysław „Zeman” Gula: Dziedziczenie wielokrotne w Delphi nie gryzie

Streszczenie

Czy potrzebowałeś kiedyś dodać wspólną funkcjonalność do różnych wizualnych komponentów? Czy zdarzyło Ci się stworzyć obiekt z wieloma referencjami do niego? Jak rozwiązałeś problem kiedy go zwolnić? Jeśli masz zmienną typu TControl i chcesz zmienić kolor kontrolki nie znając jej właściwego typu, jak to zrobisz? Obecnie spora część programistów Delphi trzyma się z daleka od stosowania interfejsów, a więc i od dziedziczenia wielokrotnego. Celem wykładu będzie pokazanie korzyści płynących ze stosowania takiego dziedziczenia oraz interfejsów. Jednocześnie, żeby być obiektywnym, pokażę też wady i niebezpieczeństwa.

Konspekt

1. Wprowadzenie
 - (a) Na czym polega dziedziczenie wielokrotne?
 - (b) Wady i zalety
2. Przypomnienie podstaw / szybkie omówienie
 - (a) Polimorfizm w klasach
 - (b) Czym są interfejsy i jak z nich korzystać?
 - i. Co to jest interfejs?
 - ii. Zasady korzystania w klasie, prosty przykład
 - iii. Agregacja na wzór Javy
 - iv. Agregacja w Delphi
 - v. Zarządzanie życiem – jak działa, prosty przykład, zalety
 - vi. Niebezpieczeństwa związane ze stosowaniem interfejsów
3. Projektowanie klas
 - (a) Interfejs jako umiejętność klasy
 - (b) Polimorfizm wielokrotny
4. Praktyczne przykłady, pewne przydatne konstrukcje
 - (a) Problem: kiedy zwolnić obiekt, który ma wiele referencji?
 - (b) Dodajmy nową wspólną funkcjonalność do różnych klas / komponentów
 - (c) Konstrukcje kompozytowe – struktury drzewiaste o różnych zachowaniach węzłów
 - (d) Obserwator – zmienmy właściwość obiektu, niech się sam automatycznie odrysuje na wszystkich komponentach
5. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Przemysław (Zeman) Gula programuje od 14 lat, w Delphi od 10 lat, zawodowo od 8. Poza programowaniem interesuje się wschodnimi sztukami walki. Uwielbia grać w piłkarzyki stołowe i (między innymi) gilgać Narzeczoną. Obecnie pracuje jako programista Delphi w warszawskiej firmie Webtel.

Maksymilian „Maks” Leszczyński: Jak tworzyć udane oprogramowanie?

Streszczenie

Dlaczego niektórzy produkują oprogramowanie wysokiej jakości w terminie lub przed terminem, podczas gdy inni (w większości) spóźniają się, przekraczają budżet lub zupełnie zawalają projekt? Istnieje wiele teorii tłumaczących ten stan rzeczy i wiele metodologii próbujących go poprawić. Wykład ten omówi większość problemów z jakimi musi radzić sobie firma produkująca oprogramowanie, a także wskaże jak uniknąć kardynalnych błędów i oszczędzić środki tak, aby projekty kończyły się sukcesem. Lepiej uczyć się na cudzych błędach niż na swoich. Wykład jest przeznaczony tak dla programistów, jak i dla menagerów lub właścicieli firm.

Konspekt

1. Wstęp
2. Narzędzia i infrastruktura
 - (a) Podstawowe zasady programowania
 - (b) Zarządzanie zasobami
 - (c) Skrypty kompilacji (konsolidacji)
 - (d) Śledzenie problemów i nowych funkcji
 - (e) Testowanie
 - (f) Wybór narzędzi
 - (g) Kiedy nie eksperymentować?
3. Pragmatyczne techniki projektowe
 - (a) Lista zadań
 - (b) Kierownik techniczny
 - (c) Codzienna koordynacja i komunikacja zespołu
 - (d) Przeglądy kodu
 - (e) Powiadomienia o zmianie kodu
4. Metodologia pocisku smugowego (wg patentu panów z SAS)
5. Typowe problemy i metody postępowania z nimi
 - (a) Testowanie nietestowalnego programu
 - (b) Powracające błędy
 - (c) Zaprzestanie wykonywania testów
 - (d) U mnie to działa!
 - (e) W firmie nie wykonuje się automatycznych testów
 - (f) Problemy z integracją kodu
 - (g) Problemy z kompilacją i konsolidacją projektu
 - (h) Klienci są niezadowoleni
 - (i) Masz niezdyscyplinowanego programistę
 - (j) Twój szef jest niezadowolony
 - (k) W zespole nie ma współpracy
 - (l) Nie potrafię przekonać innych do istotnych kwestii
 - (m) Nowy standard postępowania nie pomógł

- (n) Nietoświadczeni programiści bez mentora
- (o) Projekt typu „marsz śmierci”
- (p) Ciągłe proponowane są nowe funkcjonalności
- (q) Nigdy nie kończymy

6. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Maks jest programistą systemu ERP w firmie Asseco Business Solutions S.A., programuje od 15 lat, w tym zawodowo od prawie 8. Programuje w Delphi od początku, na poważnie od Delphi 2. Przed powstaniem Delphi 1 pasjonował się Turbo Pascal, pisząc w nim liczne programy edukacyjne oraz gry. Wcześniej jego kariera obijała się o ratownictwo wodne (ma uprawnienia Ratownika Wodnego), a także branżę ubezpieczeniową (agent ubezpieczeniowy). Pracował także w księgarni informatycznej jako sprzedawca oraz w firmie kurierskiej jako kurier rowerowy. Poza tym ma żonę i psa, prowadzi osiadły tryb życia.

Marcin „Frodo2000” Molak: DB2 9.5 pureXML – przetwarzanie dokumentów XML od kuchni

Streszczenie

We wrześniu 2006 firma IBM udostępniła hybrydowy serwer danych DB2 9.1, w którym zintegrowała silnik hybrydowy z relacyjnym. Dzięki temu deweloperzy aplikacji uzyskali wydajną i elastyczną obsługę dokumentów XML, które z natury posiadają strukturę hierarchiczną. W wersji 9.5 serwera danych DB2, która ukazała się na rynku w październiku 2007, możliwości silnika pureXML zostały rozszerzone. Podczas prezentacji zostaną omówione możliwości obsługi dokumentów XML, z naciskiem na nowe funkcjonalności. Przedstawione zostaną przykładowe aplikacje, wykorzystujące jednocześnie model relacyjny i hybrydowy oraz ich integracja z aplikacją napisaną w środowisku Delphi.

Konspekt

1. Potrzeba składowania dokumentów XML w bazach danych
2. Repozytorium dla dokumentów hierarchicznych w DB2
3. Przetwarzanie dokumentów XML za pomocą języków SQL/XML oraz XQuery
4. Walidacja, transformowanie i dekompozycja dokumentów XML
5. Przykłady aplikacji wykorzystujących możliwości silnika pureXML w DB2 9.5
6. Delphi i DB2 razem - przygotowanie aplikacji
7. DB2 Express-C - nowe zasady licencjonowania bezpłatnej bazy danych
8. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Marcin Molak pracuje jako specjalista w dziale oprogramowania (SWG) IBM Polska. Zajmuje się wsparciem technicznym dla producentów oprogramowania, zagadnieniami tworzenia aplikacji opartych na serwerach danych IBM oraz tematyką narzędzi analityczno-raportujących. W latach 2006-2007 członek zespołu programistów Polskiego Centrum Joomla!

Wojciech „Spook” Sura : Renderowanie grafiki 3D

Streszczenie

Wykład będzie składał się z trzech części. Na początku wspomnę krótko o historii grafiki trójwymiarowej i wyjaśnię, jak od strony matematycznej działają renderery. Druga część będzie zawierała informacje o tym, co zrobić, by się nie narobić – czyli jak renderować grafikę 3D korzystając z gotowych narzędzi, takich jak OpenGL i DirectX. Pokażę również, napisany przez siebie, prosty silnik renderujący grafikę trójwymiarową. Na końcu obejrzymy kilka ciekawostek – niecodzienne pomysły realizowane przez pomysłowych programistów na całym świecie.

Konspekt

1. Krótka historia grafiki trójwymiarowej
2. Trochę matematyki, czyli jak to działa?
 - (a) Zupełne podstawy – powtórka z liceum
 - (b) Algorytm malarza
 - (c) Algorytm Z-bufora
 - (d) Raycaster i raytracer
 - (e) Jak to wygląda w praktyce, czyli Spk3DPro
3. Co zrobić, żeby się nie narobić – korzystamy z gotowców
 - (a) OpenGL
 - (b) DirectX
4. Wielki przekręt, czyli realistyczny rendering
 - (a) Oszukujemy przy projektowaniu sceny (tekstury)
 - (b) Oszukujemy przy obliczaniu oświetlenia (cieniowanie Phong, HDRI)
 - (c) Oszukujemy przy przetwarzaniu gotowego renderu (postprocessing)
5. Szalone pomysły (przeważnie akademickie)
 - (a) Realtime raytracing
 - (b) Quake 3 w 95 kB, czyli Kkrieger
6. Galeria – grafika 3D w grach i renderach.
7. Podsumowanie i pytania od uczestników wykładu

O wykładowcy

Wojciech "Spook" Sura jest studentem V roku Matematyki z Informatyką na Uniwersytecie Wrocławskim. W Delphi programuje już od ponad siedmiu lat - pisze hobbystycznie, jak również bierze udział w komercyjnych projektach. Grafiką trójwymiarową zaczął się interesować w momencie, gdy światło dzienne ujrzała wersja 2.42 darmowego Blendera. Prowadzi na uczelni laboratorium, w ramach którego uczy studentów obsługi tego ciekawego narzędzia. Matematyczne aspekty renderowania grafiki trójwymiarowej są tematem jego pracy magisterskiej. Był również autorem warsztatów na Zlocie Programistów Delphi 2006. Prowadził i prowadzi kółka informatyczne dla licealistów i gimnazjalistów. Gdy nie siedzi przed komputerem, relaksu szuka przemierzając okolice Wałbrzycha konno i na rowerze.

I to by było na tyle !

Do zobaczenia na
Złocie Programistów Delphi 2008
w Krakowie!