

Jak stworzyć projekt używając OgreOde?

Ten dokument to poradnik krok po kroku z czynności jakie należy wykonać, aby poprawnie uruchomić pierwszy projekt używając OgreOde (albo przynajmniej to co ja musiałem zrobić, żeby to osiągnąć). Większość punktów okaże się pewnie nikomu nieprzydatna, gdyż postaram się wrzucić na svn-a skompilowane biblioteki, a projekt z dodanymi odpowiednimi odwołaniami do katalogów i bibliotek już stworzyłem, co nie znaczy, że wszystko od razu będzie działać. Część rzeczy każdy będzie musiał zainstalować u siebie na komputerze, żeby poprawnie skompilować projekt.

#### 1. Instalacja Ogre3D

Ten punkt akurat musi być spełniony (i pewnie większość już ma Ogre3D SDK zainstalowane), ale dla kompletu:

- Wchodzimy na <http://www.ogre3d.org/download/sdk> i ściągamy obecną wersję Ogre3D dla swojej wersji Visual Studio (czyli pewnie dla VS 2008 SP1:

[http://sourceforge.net/projects/ogre/files/ogre/1.6.4/OgreSDKSetup1.6.4\\_VC90.exe/download](http://sourceforge.net/projects/ogre/files/ogre/1.6.4/OgreSDKSetup1.6.4_VC90.exe/download))

-Instalujemy do wybranego katalogu. Katalog ten będzie z automatu zapisany jako zmienna środowiskowa OGRE\_HOME i tak się do niego będę odnosić w dalszej części.

-zrobione :D

#### 2. ODE

Dla nadgorliwych podpunkt a. o tym jak skompilować ODE, żeby poprawnie współpracowało z OgreOde. Dla tych, którym nie chce się bawić wersję skompilowaną postaram się umieścić na svn-ie. Dla mniej nadgorliwych podpunkt b., czyli co zrobić z gotową biblioteką.

Warte zauważenia jest, że instalując Ogre3D SDK mamy gotową skompilowaną wersję ODE, lecz jest ona trochę przestarzała, potrzebujemy bardziej aktualnej (dla czego wyjaśni się w punkcie 3.).

##### a. Kompilacja

-Pobieramy źródła ODE z

<http://sourceforge.net/projects/opende/files/ODE/0.11.1/ode-0.11.1.zip/download>

-Wypakowujemy do jakiegoś katalogu (nazwijmy go dla ustalenia uwagi ODE), najlepiej w miarę blisko OGRE\_HOME, co by nie biegać za dużo między folderami później, ale nie w środku, bo może się coś pomieszać.

-Uruchamiamy wiersz poleceń, przechodzimy do katalogu ODE/build i wpisujemy polecenie *premake4 vs2008* (dla innych wersji Visual Studio odpowiednio inny numer; więcej informacji *premake4 --help*). Stworzony zostanie katalog vs2008 a w nim pliki .sln i .vcproj.

-Otwieramy solucję, wybieramy konfigurację kompilacji . Wybór jest ponoć dowolny, ale polecam ReleaseSingleLib, czyli biblioteka statyczna na floatach skompilowana jako Release – raz, że Release, więc później nie będzie trzeba przekompilowywać, dwa, że Single, bo z floatami jest bezpiecznie, a z doublami miałem w OgreOde jakieś problemy (ponoć się da, ale nie chciało mi się kombinować), a trzy, że Lib czyli biblioteka statyczna, czyli nie trzeba się bawić w linkowanie dynamiczne dll-ek.

-W ODE/lib/\$(NazwaKonfiguracji) mamy gotową bibliotekę.

-zrobione :D

##### b. Kopiowanie

Mając bibliotekę (czy to z kompilacji, czy z svn-a) musimy zmienić jej nazwę na „ode.lib” i przekopiować do OGRE\_HOME\lib nadpisując istniejącą tam wersję. Ponadto musimy zaktualizować pliki nagłówkowe ode, żeby odpowiadały nowej wersji. W tym celu musimy mieć źródła ODE (link wyżej). Wszystkie pliki z ODE\include\ode kopiujemy do OGRE\_HOME\include\ode.

-zrobione :D

### 3. OgreOde

Tym razem będziemy ściągać, konfigurować i kompilować OgreOde. Ponownie postaram się umożliwić pominięcie etapu kompilacji poprzez umieszczenie skompilowanych wersji w repozytorium.

#### a. Źródła

Ściągnięcie źródeł polecam każdemu ponieważ zawierają sporo przykładów kodu, materiałów, tekstur itp. Ponadto rozpakujemy je bezpośrednio do OGRE\_HOME, co ułatwi odwoływanie się do bibliotek.

-Adres :

<http://ogreaddons.svn.sourceforge.net/viewvc/ogreaddons/trunk/ogreode.tar.gz?view=tar>

-rozpakowujemy bezpośrednio do OGRE\_HOME, powinien zostać utworzony katalog „ogreode” a w nim wszystkie źródła itp.

#### b. Kompilacja

Jak już pisałem punkt można pominąć.

-Otwieramy plik OGRE\_HOME\ogreode\scripts\VC8\OgreOde\_Source.sln. Nie ma wersji dla Visual Studio 2008, ale sam sobie skonwertuje, więc nie ma problemu. Nie należy się przejmować błędami, że nie znaleziono jakiegoś projektu.

-Wyrzucamy z solucji projekt „ode”, którego nie może załadować (jak ktoś ma źródła ODE to może dodać ten projekt do solucji, ale nie wiem czy to coś zmienia. Ja dodałem, bo tak ktoś gdzieś napisał, żeby zrobić. Nie stwierdziłem jednak, żeby był ten projekt używany, ale nie chciało mi się później sprawdzać czy się kompiluje bez niego).

-Pomieszać trzeba we właściwościach projektów i to całkiem sporo. Poniżej tabelka co gdzie dla jakiego projektu powinno się znajdować (prawdopodobnie większość, jak nie wszystko, będzie dobrze ustawiona, ale już nie pamiętam co zmieniałem, więc wrzuciłem całą konfigurację).

-Rekompilujemy projekty OgreOde\_Core, OgreOde\_Prefab i OgreOde\_Loader.

-zrobione :D

Projekt	OgreOde_Core	
Konfiguracja	Debug	Release
Additional Include Directories (Configuration Properties -> C/C++ -> General)	\$(OGRE_HOME)\ogreode\include \$(OGRE_HOME)\include	\$(OGRE_HOME)\ogreode\include \$(OGRE_HOME)\include
Additional Library Directories (Configuration Properties -> Linker -> General)	\$(OGRE_HOME)\lib	\$(OGRE_HOME)\lib
Additional Library Directories (Configuration Properties -> Linker -> Input)	ode.lib ogremain_d.lib	ode.lib OgreMain.lib
Command Line (Configuration Properties -> Build Events -> Post-Build Event)	copy "\$(OutDir)\\$(TargetFileName)" "\$\$(OGRE_HOME)\Bin\\$(ConfigurationName)"	copy "\$(OutDir)\\$(TargetFileName)" "\$\$(OGRE_HOME)\Bin\\$(ConfigurationName)"

Projekt	OgreOde_Loader	
Konfiguracja	Debug	Release
Additional Include Directories (Configuration Properties -> C/C++ -> General)	\$(OGRE_HOME)\ogreode\loader\include \$(OGRE_HOME)\ogreode\include \$(OGRE_HOME)\ogreode\prefab\include \$(OGRE_HOME)\include \$(OGRE_HOME)\ogreode\tinyxml	\$(OGRE_HOME)\ogreode\loader\include \$(OGRE_HOME)\ogreode\include \$(OGRE_HOME)\ogreode\prefab\include \$(OGRE_HOME)\include \$(OGRE_HOME)\ogreode\tinyxml
Additional Library Directories (Configuration Properties -> Linker -> General)	\$(OGRE_HOME)\lib \$(OGRE_HOME)\ogreode\lib\\$(ConfigurationName) \$(OGRE_HOME)\ogreode\prefab\lib\\$(ConfigurationName) \$(OGRE_HOME)\ogreode\tinyxml\\$(ConfigurationName)	\$(OGRE_HOME)\lib \$(OGRE_HOME)\ogreode\lib\\$(ConfigurationName) \$(OGRE_HOME)\ogreode\prefab\lib\\$(ConfigurationName) \$(OGRE_HOME)\ogreode\tinyxml\\$(ConfigurationName)

Additional Library Directories (Configuration Properties -> Linker -> Input)	tinyxmld.lib OgreOde_Core_d.lib OgreOde_Prefab_d.lib OgreMain_d.lib	tinyxml.lib OgreOde_Core.lib OgreOde_Prefab.lib OgreMain.lib
Command Line (Configuration Properties -> Build Events -> Post-Build Event)	copy "\$(OutDir)\\$(TargetFileName)" "\$(OGRE_HOME)\Bin\\$(ConfigurationName)"	copy "\$(OutDir)\\$(TargetFileName)" "\$(OGRE_HOME)\Bin\\$(ConfigurationName)"

Projekt	OgreOde_Prefab	
Konfiguracja	Debug	Release
Additional Include Directories (Configuration Properties -> C/C++ -> General)	\$(OGRE_HOME)\ogreode\prefab\include \$(OGRE_HOME)\ogreode\include \$(OGRE_HOME)\include \$(OGRE_HOME)\ogreode\tinyxml	\$(OGRE_HOME)\ogreode\prefab\include \$(OGRE_HOME)\ogreode\include \$(OGRE_HOME)\include \$(OGRE_HOME)\ogreode\tinyxml
Additional Library Directories (Configuration Properties -> Linker -> General)	\$(OGRE_HOME)\lib \$(OGRE_HOME)\ogreode\lib\\$(ConfigurationName) \$(OGRE_HOME)\ogreode\tinyxml\\$(ConfigurationName)	\$(OGRE_HOME)\lib \$(OGRE_HOME)\ogreode\lib\\$(ConfigurationName) \$(OGRE_HOME)\ogreode\tinyxml\\$(ConfigurationName)
Additional Library Directories (Configuration Properties -> Linker -> Input)	OgreOde_Core_d.lib OgreMain_d.lib tinyxmld.lib	OgreOde_Core.lib OgreMain.lib tinyxml.lib
Command Line (Configuration Properties -> Build Events -> Post-Build Event)	copy "\$(OutDir)\\$(TargetFileName)" "\$(OGRE_HOME)\Bin\\$(ConfigurationName)"	copy "\$(OutDir)\\$(TargetFileName)" "\$(OGRE_HOME)\Bin\\$(ConfigurationName)"

c. Kopiowanie

-Jeśli kompilowaliśmy OgreOde to wszystko już powinno się znajdować na swoim miejscu.

- Jeśli nie to pliki nagłówkowe projektu OgreOde\_Core powinny znaleźć się w \$(OGRE\_HOME)\ogreode\include, z projektu OgreOde\_Prefab w \$(OGRE\_HOME)\ogreode\prefab\include a z projektu OgreOde\_Loader w \$(OGRE\_HOME)\ogreode\loader\include. Następnie należy ściągnąć z naszego repozytorium paczkę OgreOde.rar (już wkrótce ;) i rozpakować w OGRE\_HOME, a odpowiednie pliki bibliotek umieszczone zostaną w odpowiednich folderach.

-zrobione :D

4. Ogre SDK AppWizard

Jakby ktoś nie wiedział to są template'y do VS, użyteczne do tworzenia nowych projektów używających Ogre3D

-Adres:

[http://sourceforge.net/projects/ogreconglo/files/Ogre%20SDK%20VC9.0%20Appwizard/1.5.1/ogresdkwizard90\\_v1\\_5\\_1.zip/download](http://sourceforge.net/projects/ogreconglo/files/Ogre%20SDK%20VC9.0%20Appwizard/1.5.1/ogresdkwizard90_v1_5_1.zip/download)

-Rozpakować to należy w jakieś sensowne miejsce, ponieważ VS będzie używać plików zawartych w tym katalogu, więc po instalacji nie można usunąć tego folderu

-Uruchamiamy VC9\_Setup.js (zakładam że każdy ma Visual Studio 2008 – po więcej informacji zapraszam do readme z tego folderu.

-Uwaga dla Windows Vista i Windows 7 – jako że skrypt wymaga uprawnień administratora mogą wystąpić problemy z uprawnieniami. Aby to obejść należy rozpakować archiwum gdzieś na dysku systemowym (na inne nie bardzo da się wejść z konsoli). Następnie należy uruchomić wiersz poleceń z uprawnieniami administratora, przejść do rozpakowanego folderu i uruchomić VC9\_Setup.js.

-bangła :D

5. Nowy projekt

To jest mniej więcej opis z komentarzami tego co zrobiłem tworząc nowy projekt dla naszej gry. I oczywiście nikt tego nie przeczyta, bo to już zostało zrobione raz i starczy ;).

-Uruchamiamy VS. Klikamy Nowy Projekt, -> Visual C++ -> OGRE SDK Application, nadajemy nazwę i OK. Pojawia się kreator, można tam poeksperymentować z jakimiś opcjami, ja wybrałem: Application Type – Minimal Application i Additional Options : CEGUI Support i Postbuild Copy.

-Włazimy we właściwości projektu i do Additional Include Directories (Configuration Properties -> C/C++ -> General) dodajemy "\$\$(OGRE\_HOME)\ogreode\include", "\$\$(OGRE\_HOME)\ogreode\loader\include" i "\$\$(OGRE\_HOME)\ogreode\prefab\include". Do Additional Library Directories (Configuration Properties -> Linker -> General)

"\$(OGRE\_HOME)\ogreode\lib\\$(ConfigurationName)",

"\$(OGRE\_HOME)\ogreode\loader\lib\\$(ConfigurationName)" i

"\$(OGRE\_HOME)\ogreode\prefab\lib\\$(ConfigurationName)". W obu przypadkach dotyczy

to zarówno konfiguracji Debug jak i Release. Do Additional Dependencies (Configuration

Properties -> Linker -> Input) w konfiguracji Debug dodajemy OgreOde\_Core\_d.lib,

OgreOde\_Loader\_d.lib i OgreOde\_Prefab\_d.lib, a w Release OgreOde\_Core.lib,

OgreOde\_Loader\_d.lib i OgreOde\_Prefab.lib.

-piszemy grę :D