

Projekt realizowany w ramach przedmiotu Multimedia i Interfejsy za zgodą dr inż. Mariusza Szwoch, drugi rok kierunku Informatyka na Politechnice Gdańskiej.

Temat: Trójwymiarowy interaktywny model nowego gmachu wydziału Elektroniki, Telekomunikacji i Informatyki Politechniki Gdańskiej.

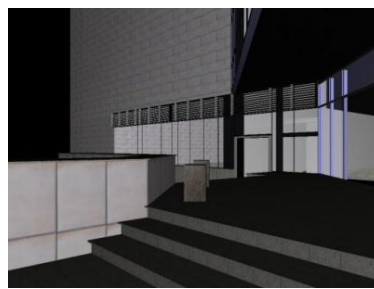
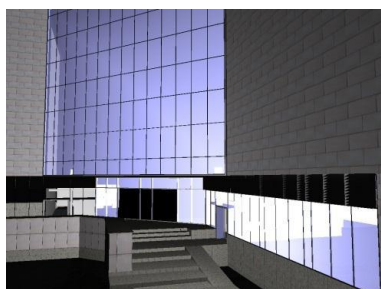
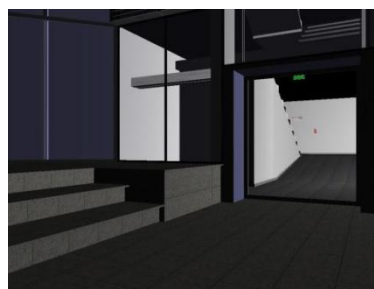
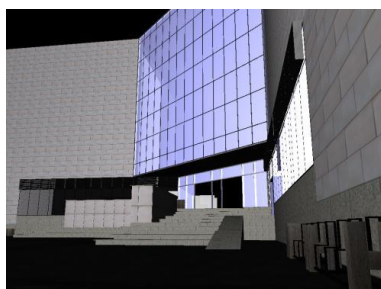
Funkcjonalność podstawowa:

- odwzorowanie kształtu budynku i większości sprzętów będących na wyposażeniu z wykorzystaniem OpenGL'a;
- wykorzystanie tekstur jak najbardziej przypominających prawdziwe (większość dzięki zdjęciom);
- reagowanie modelu na położenie gracza, tj. automatyzacja ruchu drzwi, windy, etc.;
- oświetlenie zgodne z rzeczywistym (światłem głównie słonecznym)
- ruch po budynku z podstawową detekcją kolizji;
- zapis modelu w sposób umożliwiający łatwe przeniesienie z platformy WinOS na dowolną (zapis w formacie własnym);

Funkcjonalność optymistyczna:

- implementacja dźwięku (zakres informacyjny, np. nagrane rozmowy studentów, krótkie wypowiedzi nt. pomieszczeń, opis budynku, etc.) z wykorzystaniem OpenAL'a;
- dwa tryby oświetlenia: tryb dzienny i nocny;
- w pełni działający system detekcji kolizji umożliwiający symulację faktycznego ruchu po budynku;

Kilka screenów pokazowych z pierwszej wersji modelu (7/11/2010):



Screeny dostępne w wyższej rozdzielczości na stronie http://code-silo.pl/model_eti_2010/

Autor: Maciej Grzymkowski, student Informatyki na ETI na Politechnice Gdańskiej

1. Wykorzystane oprogramowanie i biblioteki

Grafika – OpenGL
Obsługa zdarzeń i okna – WinApi
Dźwięk – OpenAL

(najnowsze wersje dostępne w listopadzie 2010r.)

Model został przygotowany w programie Microsoft Visual Studio 2010 (kompilacja i linkowanie).

Do tworzenia obiektów wykorzystano kilka prostych programów do generowania list prymitywów – całość dostępna w projekcie Modelator w załączonych plikach, m.in.:

- Hraniastolup – aplikacja umożliwiająca tworzenie sekwencji prostych obiektów; umożliwia określenie współrzędnych, ścian, tekstur, ilości powtórzeń, przesunięcia „co obiekt”,
- Modelator – podstawowy program do szybkiego tworzenia opakowanych sześciokątów; możliwy wybór tekstur, współrzędnych i selekcji ścian oraz większej ilości obiektów,
- i inne.

2. Właściwa budowa – sposób konstruowania modelu

Całość projektu bazuje na idei maksymalnej uniwersalności zarówno kodu programu, jak i wykorzystywanych współrzędnych obiektów. Model przewiduje możliwość wykorzystania różnych map, w chwili obecnej tylko jednej – mapy ETI. Każda mapa składa się z pewnej listy sektorów.

Przykładowo, ETI zbudowane jest z:

- części zewnętrznej,
- sektora budynków,
- sektora lewej strony,
- sektora prawej strony,
- sektora środkowego.

Każdy sektor może zostać zmodyfikowany, np. powiększony, przeniesiony, powielony. Idąc głębiej, sektory zbudowane są z listy obiektów, tak jakby definicji użytych pod-modeli, i instancji obiektów – faktycznych wystąpień na planie modelu. Dodatkowo, niektóre instancje mogą być powiązane z akcjami, np. ruchem, aktywacją przyciskiem, dźwiękiem.

W chwili obecnej możliwe akcje to ruch, obsługa klawiszy, aktywacja dźwięku i detekcja pozycji użytkownika (widza).

Wszystkie współrzędne czytane są z plików – pozwala to na łatwą i szybką modyfikację, jak i stwarza możliwości ponownego wykorzystania kodu – nie stwarza to żadnych problemów.

3. Szczegółowa konstrukcja map, sektorów, obiektów i instancji – schemat, zawartość i charakterystyka plików i zastosowanego formatu.

3.1. Mapy - przykładowa zawartość pliku mapy (map_zewnatrz.txt):

(wiersze wszystkich plików źródłowych rozpoczynające się znakiem nowego wiersza lub dwoma slash'ami są ignorowane przez funkcję ładującą)

```
Na_zewnatrz
NUMSECTORS 2

#0 - (0, 0.0, 0.0), (1, 1, 1), data/sector_schodyL.txt wejscie_lewe 0.0f
#1 - (4.65, 0.0, 8.06), (1, 1, 1), data/sector_schodyP.txt wejscie_prawe 60.0f
```

Plik mapy zaczyna się nazwą mapy, która może zostać wykorzystana np. przy wybieraniu pewnej konkretnej części budynku.

Następnie słowo kluczowe NUMSECTORS N przekazuje informację o ilości sektorów na danej mapie. Kolejne N wierszy to definicje sektorów wg schematu:

<nr_sektora> - (Tx, Ty, Tz), (Sx, Sy, Sz), <ścieżka> <nazwa_sektora> <y_rot>

- Tx, Ty, Tz: składowe wektora przesunięcia
- Sx, Sy, Sz: składowe wektora skali
- y_rot: rotacja wokół osi OY (tylko ta uwzględniona, reszta wydaje się niepraktyczna)

3.2. Sektory – na przykładzie sector_budynek.txt:

```
NUMMODELS 6
NUMINSTANCES 49

// modele
#0 - data/models/architektura/budynek.txt budynek
#1 - data/models/drzwi_okna/okna_zewnatrz.txt okna
#2 - data/models/architektura/budynek_trzeci.txt budynek_trzeci

// instancje - wystapienia modeli
// budynek - pierwszy (prawy, lub ten z tablica zlota ETI)
#0 - 5, (-6.5, 0, -3.25), (1, 1, 1), (0, 0, 0), 1, n
#1 - 1, (-0.5, 2.312, 2.0464), (1, 1, 1), (0, 0, 0), 1, n
```

Analogicznie do pliku mapy, pierwsze linijki oznaczają liczbę elementów do wczytania. NUMMODELS N i NUMINSTANCES M – odpowiednio ilość modeli i obiektów.

Następne N wierszy przyjmuje formę:

<nr_obiektu> - <ścieżka>, nazwa

- ścieżka zawiera względne położenie pliku obiektu

Kolejne M wierszy:

<nr_instancji> - <id_obiektu>, (<przesuniecie>), (<skala>), (obroty), <typ>, <akcja>

- pole id_obiektu odnosi się do wcześniej zdefiniowanych modeli-obiektów
- środkowe pola w nawiasach to wektory modyfikacji obiektu; pozwala to na wykorzystanie jednego modelu do stworzenia wielu jego wystąpień, o różnej wielkości, obrocie, itp.
- pola typ i akcja umożliwiają podpięcie pod obiekt pewnego zdarzenia

Zdarzenia definiowane są w plikach akcji, przykładowo (sector_schodyL.txt):

```
#25 - 12, (-3.0, 0.57, 4.462), (1, 1, 1), (0, 0, 0), 0, m
data/actions/action_przesuwania_windaL1B.txt
```

25 instancja to 12 zadeklarowany model. Pole typ równe zero oznacza element złożony, kodowany polem akcja. Następną linijką zawiera ścieżkę do pliku definiującego daną akcję związana z obiektem.

- 'm' oznacza ruch + dźwięk + trigger klawiszem

3.3. Przykładowa zawartość pliku akcji (dla powyższego obiektu):

```
// plik przesuwania dla drzwi
#10, (0.0, 0.0, -0.15) - A[3, -1.2, -4.6], B[3.9, -0.7, -4.1], P
music_file = data/sound/door_open.wav
sound_buffer = 2
```

Według schematu:

- #<ilość_kroków_animacji>, (<przesunięcie>) – A[<wsp.>], B[<wsp.>], <trigger>
- music_file = <ścieżka_z_plikiem_muzycznym>
- sound_buffer = <id_bufora_dźwięku>

Ilość kroków animacji pozwala na dostosowanie szybkości lub płynności animacji. Punkty A i B wyznaczają graniastosłup aktywacji – akcje będą wykonywane gdy użytkownik znajdzie się w środku. Trigger umożliwia opcjonalną obsługę dodatkowego kryterium – klawisza (0 – brak). Następne dwa pola wykorzystywane są przez dźwięk. Indeks tablicy źródeł podany na sztywno w związku z (póki co) niewielką ilością dźwięków w modelu; wymaga modyfikacji.

3.4. Obiekty, czyli podstawowy element modelu

W chwili obecnej, poszczególne modele tworzone są z prymitywów – wielokątów z miażdżącą przewagą trójkątów. Aby wczytywać inne typy, wystarczyłoby oznaczenie obecnych modeli jako ‘proste’, a następne jako ‘zaawansowane’ i podpiąć pod te dwa typy dwie różne funkcje renderujące/wczytujące.

Fragment pokazujący przykładową definicję kilku ścian:

```
NUMPOLLIES 124
// regal_pion #1
// regal_pion_A
0 0 0 0 1
0 0.55 0 0 0
0.01 0.55 0 1 0
9

0.01 0.55 0 1 0
0.01 0 0 1 1
0 0 0 0 1
9
```

- pierwsza linijka zawiera deklarację ilości trójkątów do wczytania
- następna linijka zwykle zawiera informację o przeznaczeniu danej ściany, czasem dodatkowo podział na podgrupy modelu (np. ‘ściany’, ‘podłogi’, ‘okna’, etc.)
- kolejne linie należy łączyć w „paczki” po 4 – pierwsze trzy to punkty wyznaczające trójkąt (pierwsze trzy współrzędne) i pozycja „na teksturze”, co pozwala na sporą elastyczność w tworzeniu modeli. Czwarty wiersz to id tekstury – te ładowane są wewnątrz programu (informacje w kodzie .cpp programu).

4. Dźwięk

Implementacja z wykorzystaniem OpenAL SDK – dźwięk poruszania się, ruchomych drzwi i opcjonalnie włączanie muzyczki.

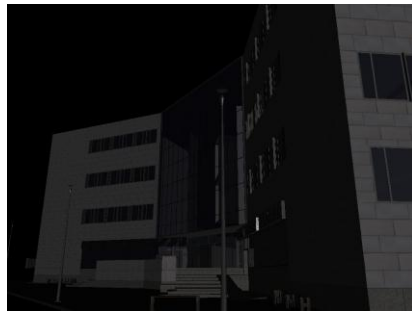
Dźwięki pobrane ze strony <http://www.freesound.net> oraz z gry The Neverhood (muzyka: Terry S. Taylor).

5. Sterowanie

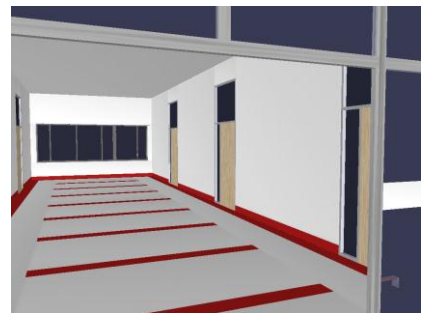
- A, W, S, D – podstawowy ruch,
- Q, E, PGUP, PGDN lub mysz – obrót,
- R, F – odpowiednio unoszenie się i opadanie,
- X – uruchomienie muzyki,
- L – włączanie/wyłączanie oświetlenia (całkowicie),
- N – przełączanie pomiędzy oświetleniem nocnym i dziennym (AMBIENT),
- I – naświetlanie (przytrzymane) – oraz G, Y, H, J, T, U – sterowanie – dodatkowe oświetlenie (DIFFUSE),
- B – włączanie/wyłączanie przenikania (wpływ na okna),
- F1 – przełączanie pomiędzy trybem okienkowym, a pełnym ekranem,
- F5 – ponowne załadowanie plików źródłowych (umożliwia aktualizację/modyfikację modelu bez potrzeby ponownego uruchamiania),
- ESC – koniec.

10 lutego 2011

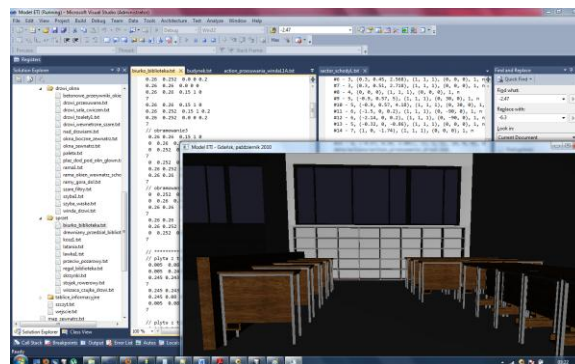
6. Obecna wersja – 10 lutego 2011



Tryb nocny a dzienny.



Kilka screen'ów...



„Proces twórczy”.



I widok ogólny. Pozdrawiam ☺

Screeny dostępne w wyższej rozdzielczości na stronie http://code-silo.pl/model_eti_2010/