

Wymagania ogólne dotyczące projektów zaliczeniowych przedmiotu "Programowanie wizualne".

1. Wymagania ogólne

- 1.1. Projekty dotyczą implementacji aplikacji umożliwiającej tworzenie map do gier 2D.
- 1.2. Aplikacja – MapDesigner - ma umożliwić składanie map z opisanych zasobów znajdujących się w plikach o formacie opisanym w punkcie 5 (zostanie wyspecyfikowany w późniejszym terminie).
- 1.3. Każdy student realizuje projekt indywidualnie. Istnieje możliwość pracy w dwuosobowych grupach, natomiast w takim przypadku wymagana jest większa funkcjonalność wspomnianych aplikacji i uzgodnienie tego z prowadzącym.
- 1.4. Przy implementacji projektów należy stosować się do wymagań standardu kodowania opisanego w punkcie 3.
- 1.5. Ocena z projektów wystawiana jest zarówno na podstawie zgodności z wymaganiami, jak i terminowości ich oddawania. Kara za spóźnienie wynosi całą ocenę za każdy rozpoczęty tydzień spóźnienia. Termin oddania projektu mija dokładnie w dniu zajęć danej grupy o godzinie rozpoczęcia zajęć.
- 1.6. Przy oddawaniu projektów należy je wcześniej wyczyścić z elementów, które są wynikiem kompilacji, a następnie zatwierdzić na swoim koncie w systemie SVN, dostępnym pod adresem
https://svn.cs.put.poznan.pl/svn/prog_wiz/oznaczenie_grupy/nr_indeksu
- 1.7. Interfejs aplikacji należy wykonać w technologii WPF z wykorzystaniem architektury opisanej w punkcie 2.

2. Architektura aplikacji

- 2.1. Należy stosować wzorzec projektowy *Aplikacja wielowarstwowa* z wydzielonymi warstwami:

- UI – interfejsy użytkownika,
- BLC – warstwa logiki biznesowej,
- DAO – warstwa dostępu do danych,

Do wyżej wymienionych warstw wspólne są następujące biblioteki pomocnicze:

- CORE – typy wyliczeniowe i ustawienia aplikacji,
- INTERFACES – wszystkie interfejsy.

- 2.2. Każdy z powyżej wymienionych elementów architektury stanowi osobną bibliotekę dll, przy czym warstwa UI jest aplikacją.
- 2.3. Nazwa przestrzeni nazw (namespace) ma być następująca:

Nazwisko.NazwaAplikacji.NazwaSkładowej, gdzie:

1. Nazwisko – bez komentarza – można pominąć polskie litery,
2. NazwaAplikacji – jak podano wcześniej,
3. NazwaSkładowej – np. BLC, DAO itp.

2.4. Każda klasa, typ wyliczeniowy i interfejs powinny znajdować się w osobnym pliku.

3. Standard kodowania

3.1. Wymaga się kodowania w standardzie

<http://weblogs.asp.net/lhunt/attachment/591275.ashx>

Uproszczonego do następujących punktów:

- P.1. z wyłączeniem 1.4.3
- P.2.1. z wyłączeniem: 7
- P.2.2.
- P.3.1. z wyłączeniem 7-20
- Bez punktu 3.2
- P. 4.1. z wyłączeniem 2-5
- P. 4.2 z wyłączeniem 18-22,25-32
- P. 4.3-4.6 proszę potraktować jako zalecenia.

3.2. Za niestosowanie się do powyższych standardów obniżana będzie ocena z projektu

4. Opis zadania

4.1. Korzystając z zasobów, które mogą być zapisane w wielu plikach, należy umożliwić użytkownikowi budowanie map 2D. W zasadzie budowanie mapy będzie wyglądało jak układanie dużego obrazka z małych części zwanych zasobami.

4.2. Każdy z zasobów opisany jest następującymi parametrami:

- nazwa,
- rozmiar,
- pozycja,
- domyślna warstwa,
- domyślny kolor transparentny,
- domyślny obszar zabroniony.

4.3. Zasoby wczytujemy z przygotowanych wcześniej plików. W zasadzie jest to para plików: plik opisujący zasób i obrazek (plik źródłowy) na którym zasoby zostały zaznaczone.

4.4. Zasoby można ustawiać na różnych warstwach. Na przykład w pierwszej kolejności ustawiamy podłoże terenu, następnie domki, ścieżki itp., a na końcu różnego rodzaju ozdobniki typu kamienie, kwiatki itd.

4.5. *Domyślny obszar zabroniony* dla zasobu jest to potencjalny obszar, po którym ludzik w grze nie powinien móc wejść. Na przykład zasób reprezentujący drzewo będzie miał obszar zabroniony wokół pnia. Oczywiście są to domyślne wartości i na etapie budowy mapy można to zmienić.

4.6. Dla mapy musi istnieć możliwość zaznaczania obszarów zabronionych, tj. takich po których postać w grze nie może się poruszać (np. budynki, drzewa, skały itd.

4.7. Każdą mapę można zapisać i odczytać.

4.8. Należy zaimplementować filtr do zasobów umożliwiający wyszukiwanie po atrybutach.