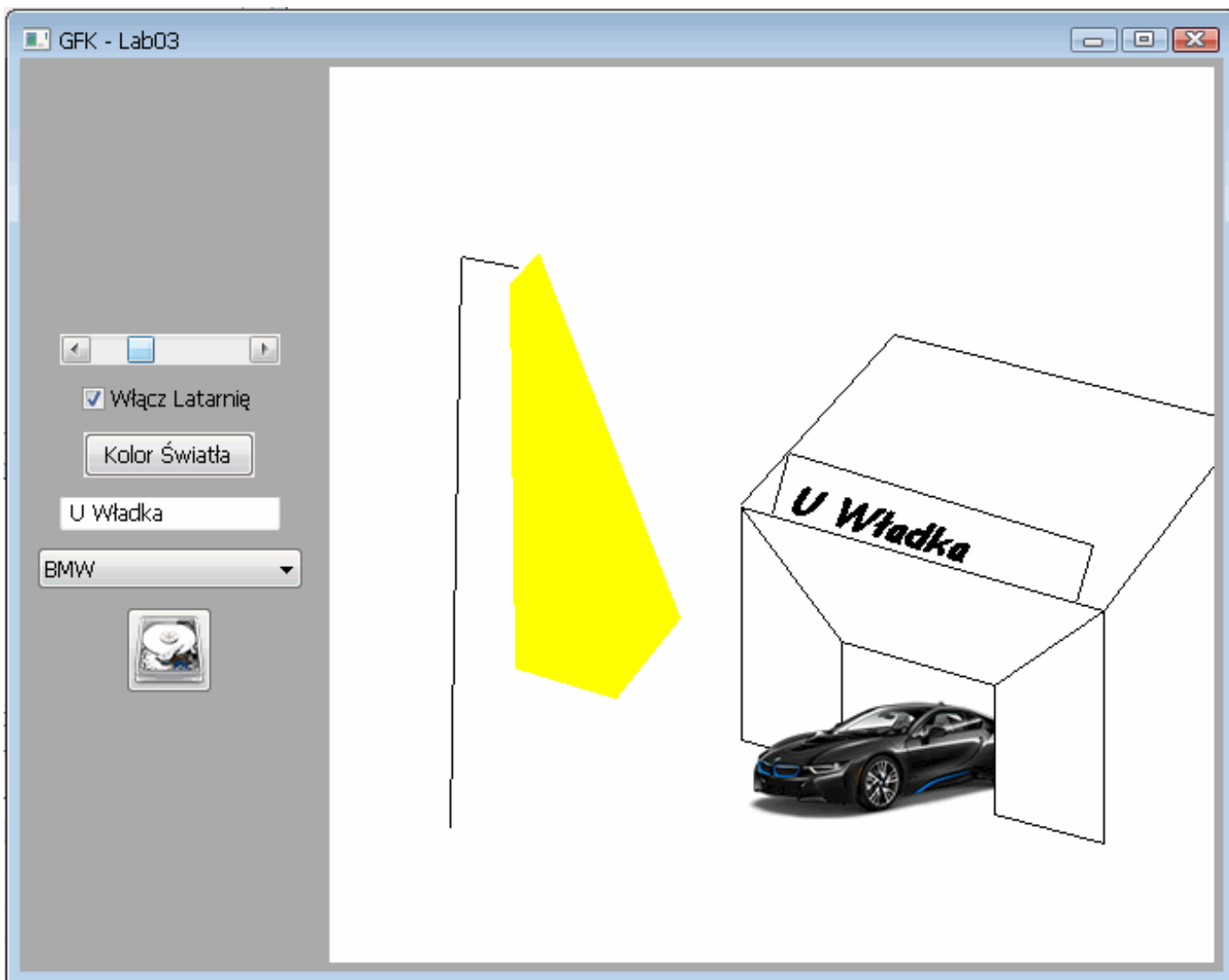


Prosta grafika w wxWidgets.

Zadanie

Napisać prosty program rysujący warsztat samochodowy na przeznaczonym na ten cel panelu. Obok warsztatu powinna się znaleźć latarnia (włączona lub wyłączona) „świecąca” światłem takiego koloru jaki wybierze użytkownik. Nazwę warsztatu można dowolnie zmieniać wpisując ją w pole do tego celu przeznaczone.



Wygląd aplikacji

Obok panelu z rysunkiem należy umieścić szereg kontroltek:

- Pasek typu `wxScrollBar` do ustawienia pozycji autka
- Kontrolka typu `wxCheckBox` włączająca i włączająca latarnię
- Przycisk typu `wxButton` do wyboru koloru światła latarni. Po jego naciśnięciu powinno zostać wyświetlone okno dialogowe wyboru koloru.
- Kontrolka typu `wxEdit` pozwalająca wprowadzić nazwę warsztatu.
- Kontrolka typu `wxChoice` do wyboru marki autka
- Kontrolka typu `wxBitmapButton` służąca do zapisania obrazka na dysku, po wciśnięciu powinno pojawić się okienko dialogowe pozwalające wybrać nazwę pliku oraz jego typ (PNG, JPEG, TIFF lub BMP).

Okno powinno mieć możliwość zmiany rozmiarów. Przy zmianie rozmiarów okna rysunek nie zmienia rozmiarów, jest natomiast zawsze wycentrowany.

Cel

Zapoznanie się z podstawami programowania interfejsu użytkownika z wykorzystaniem biblioteki `wxWidgets` w środowisku `wxDev-C++`. Obsługa prostych funkcji graficznych w `wxWidgets`. Praktyczne porównanie tworzenia grafiki z podwójnym buforowaniem i bez.

Środki

Środowisko `wxDev-C++`, biblioteka `wxWidgets`.

Zarys możliwego rozwiązania

Do rysowania najlepiej wykorzystać kontrolkę typu `wxPanel`, z którą można związać kontekst rysunkowy typu `wxClientDC`. Rysowanie najwygodniej będzie przeprowadzać w niezależnej funkcji nie powiązanej z żadną kontrolką. Funkcję tę należy zadeklarować jako funkcję składową klasy reprezentującej główne okno programu będącej klasą potomną względem `wxFrame`. Funkcję rysującą będzie można wywoływać w dowolnym momencie, gdy będzie potrzebna. Należy pamiętać również o jej wywołaniu w obsłudze odświeżania panelu `OnUpdateUI`. Funkcja rysująca może działać na kontekście rysunkowym urządzenia `wxClientDC` lub na skojarzonym z nim kontekście buforowanym `wxBufferedDC`. Warto sprawdzić oba rozwiązania, aby przekonać się o zaletach podwójnego buforowania. Do centrowania rysunku warto wykorzystać możliwość ustalania własnego układu współrzędnych w oknie. W tym celu należy skorzystać z funkcji `wxDC::SetDeviceOrigin(...)`.

Jak się przygotować przed zajęciami

Program nie jest trudny, ale przy pierwszym kontakcie z `wxWidgets` może sprawić dużo problemów. Dlatego warto poświęcić trochę czasu na przygotowanie się przed zajęciami.

Sugeruję:

- Koniecznie zapoznać się ze wszystkimi przykładami z wykładu na temat programowania w wxWidgets.
- Zapoznać się z dokumentacją wxWidgets na temat klas wxDC i wxBitmap.
- Zapoznać się z dokumentacją wxWidgets na temat klasy wxImage w stopniu potrzebnym do
- wczytywania i zapisywania obrazów. Funkcje dotyczące edycji i przekształcania obrazów typu wxImage na razie można pominąć.
- Prześledzić na podstawie wykładu, przykładów dołączonych do wxWidgets oraz dokumentacji w jaki sposób korzystać ze standardowych okien dialogowych typu wxColourDialog czy wxFileDialog.
- Przystudiować przykłady tworzenia interfejsu użytkownika z wykorzystaniem tzw „sizersów”.