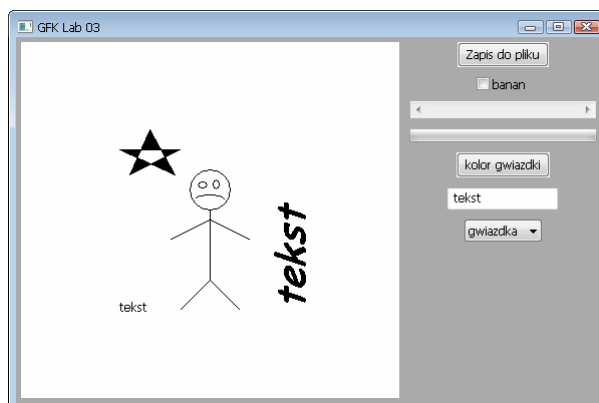


Prosta grafika w wxWidgets.

Zadanie

Napisać prosty program rysujący małpkę na środku przeznaczonego na ten cel panelu. Obok małpki powinien się znaleźć symbol graficzny do wyboru (np. gwiazdka, słońce i księżyc) oraz dwa pola reprezentujące tekst poziomy i obrócony.



Obok panelu z rysunkiem należy umieścić szereg kontrolerek:

- Przycisk typu **wxButton** służący do zapisywania aktualnego obrazka na dysk. Po jego wciśnięciu powinno się pojawić standardowe okno dialogowe pozwalające wybrać nazwę pliku do zapisu oraz katalog docelowy.
- Kontrolka typu **wxCheckBox** do zaznaczania obecności banana na rysunku. Po jego wciśnięciu program powinien wczytać obrazek banana z pliku i wyświetlić go w prawej łapce małpki. Również twarz małpki powinna zmienić wyraz na bardziej radosny.
- Pasek typu **wxScrollBar** do ustawiania kąta pochylenia łapki z bananem.
- Pasek typu **wxGauge** do graficznej prezentacji położenia łapki.
- Przycisk typu **wxButton** do wyboru koloru gwiazdki. Po jego wciśnięciu powinno zostać wyświetlone okno dialogowe wyboru koloru.
- Kontrolka typu **wxEdit** pozwalająca na wprowadzenie tekstu, który będzie rysowany na ekranie.
- Kontrolka typu **wxChoice** do wyboru jednego z trzech obiektów graficznych rysowanych powyżej małpki.

Okno powinno mieć możliwość zmiany rozmiarów. Przy zmianie rozmiarów okna rysunek nie zmienia rozmiarów, jest natomiast zawsze wycentrowany w taki sposób, że punkt połączenia łapek z tułowiem znajduje się w geometrycznym środku panelu rysunkowego.

Cel

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

Środki

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**.
- Przestudiować przykłady tworzenia interfejsu użytkownika z wykorzystaniem *sizersów*.