

# Laboratorium grafiki komputerowej i animacji

## Ćwiczenie I – Wprowadzenie do programowania w Windows.

### Przygotowanie do ćwiczenia:

1. Zapoznać się ze sposobem konstruowania i zasadą działania aplikacji Windows (Wprowadzenie)

### Przebieg ćwiczenia:

1. Pobrać ze strony przedmiotu kody źródłowe przykładowych programów – archiwum Cwic1.
2. Rozpakować archiwum w wybranym przez siebie folderze.
3. Wprowadzić do Visual Studio projekt „szablon1” (Najprościej jest dwukrotnie kliknąć na plik projektu – szablon1.dsw. Można również otworzyć VS i z menu File wybrać Open→Project/Solution, a następnie wskazać plik szablon1.dsw. Projekt otwierany w nowszych wersjach VS będzie musiał być przekonwertowany. Odbywa się to automatycznie po wyrażeniu zgody przez użytkownika.).
4. Przeanalizować kod źródłowy oraz zasoby programu z zastosowaniem materiałów wprowadzających do ćwiczenia.
5. Zmodyfikować aplikację Windows w taki sposób, że stworzona zostanie dodatkowa opcja menu (np. „Zmień napis”) i pod wpływem jej wybrania zmieniać się treść napisu umieszczonego w oknie:
  - a. Utworzyć nową opcję w graficznym edytorze menu, sprawdzić jaka stała liczbowa została powiązana z nową opcją menu (w dalszej części opracowania przyjęto, że wygenerowana stała ma wartość ID\_PLIK\_ZMIEN\_NAPIS);
  - b. Utworzyć w programie statyczną zmienną, która będzie przechowywać informację, który z napisów ma być wyświetlany (np. zmienna flaga równa 0 oznacza, że ma być wyświetlony napis „Pozdrowienia” a zmienna flaga równa 1 oznacza, że ma być wyświetlany napis „Cześć”;
  - c. Uzupełnić obsługę komunikatu WM\_COMMAND o fragment kodu odpowiedzialny za zmianę wartości zmiennej flaga:

```
case WM_COMMAND: // Komunikat przychodzący po wybraniu
                  // opcji menu itp
switch(LOWORD(wParam))
{
    // Wybrano opcję menu Plik->Koniec:
    case ID_PLIK_KONIEC:
        DestroyWindow(hwnd);
        return 0;
    case ID_PLIK_ZMIEN_NAPIS:
        flaga=(flaga==0)?1:0;
        InvalidateRect(NULL, NULL, TRUE);
        return 0;
}
```

Fragment kodu programu zaznaczony na czerwono powoduje, że przy każdorazowym wybraniu opcji menu o identyfikatorze ID\_PLIK\_ZMIEN\_NAPIS

stan zmiennej flaga ulegnie zmianie z 0 na 1 lub z 1 na 0. Wywołanie funkcji InvalidateRect spowoduje wywołanie komunikatu WM\_PAINT i odrysowanie całej zawartości okna programu.

d. Obsługa komunikatu WM\_PAINT musi również zostać zmodyfikowana:

```
case WM_PAINT: // Komunikat przychodzi w przypadku konieczności
                // "odmalowania" lub namalowania okna
                // Wymaż okno, uzyskaj kontekst urządzenia:
                HDCPaint=BeginPaint(hwnd,&PaintStruct);
                // Pobierz rozmiary obszaru roboczego okna
                GetClientRect(hwnd,&rect);
                // Umieść centralnie w oknie tekst:
                if(flaga==0) DrawText( HDCPaint,
                                      "Pozdrowienia",
                                      -1,
                                      &rect,
                                      DT_SINGLELINE|
                                      DT_CENTER|
                                      DT_VCENTER);
                else DrawText( HDCPaint,
                              "Cześć",
                              -1,
                              &rect,
                              DT_SINGLELINE|
                              DT_CENTER|
                              DT_VCENTER);

                // Zakończ rysowanie
                EndPaint(hwnd, &PaintStruct);
                return 0;
```

Przy każdym rysowaniu okna sprawdzany jest stan zmiennej flaga i w zależności od stanu zmiennej wypisywany jest odpowiedni napis.

**Uwaga:**

**W przyszłości mechanizm podobny do zaproponowanego powyżej posłuży do generowania kolejnych klatek animacji 3D.**

6. Uzupełnić aplikację Windows w taki sposób, żeby zmiana napisu (patrz punkt 5) dokonywała się pod wpływem przyciśnięcia wybranego klawisza klawiatury:
  - a. W procedurze okna wprowadzić obsługę nowego komunikatu: WM\_KEYDOWN (jest on przesyłany do procedury, kiedy użytkownik przyciska jakikolwiek przycisk klawiatury)
  - b. W obsłudze komunikatu WM\_KEYDOWN należy ponownie modyfikować zmienną flaga, która będzie sterować tekstem wyświetlanym w oknie:

```
case WM_KEYDOWN:
    if(wParam == 'A')
        if(flaga ==0) flaga = 1;
        else          flaga = 0;
        InvalidateRect(hwnd,NULL,TRUE);
    return 0;
```

7. Uzupełnić aplikację Windows w taki sposób, aby zmiana napisu (patrz punkt 5) następowała automatycznie co 1 sekundę:

- a. W procedurze okna wprowadzić obsługę nowego komunikatu WM\_CREATE (jest on wykonywany jednokrotnie w chwili uruchamiania aplikacji)
- b. W obsłudze komunikatu WM\_CREATE należy zdefiniować budzik Windows, który będzie miał identyfikator 101, będzie wysyłał komunikat WM\_TIMER co 1 sekundę i będzie przywiązany do okna aplikacji:

```
case WM_CREATE:  
    SetTimer(hwnd,101,1000,NULL);  
    return 0;
```

- c. W obsłudze komunikatu WM\_DESTROY (przed wywołaniem funkcji PostQuitMessage(0)) należy dodać funkcję zwracającą budzik do zbioru zasobów Windows:

```
case WM_DESTROY:  
    KillTimer(hwnd,101);  
    PostQuitMessage(0);  
    return 0;
```

- d. W procedurze okna wprowadzić obsługę nowego komunikatu WM\_TIMER (jest on przesyłany do procedury okna w chwili, gdy uruchomi się budzik)
- e. W obsłudze komunikatu WM\_TIMER należy ponownie modyfikować zmienną flaga, która będzie sterować tekstem wyświetlanym w oknie:

```
case WM_TIMER:  
    if(wParam == 101)  
        if(flag == 0) flag = 1;  
        else          flag = 0;  
    InvalidateRect(hwnd,NULL,TRUE);  
    return 0;
```