

Wstęp

Rozpakuj projekt. Za pomocą programu Visual Studio, otwórz plik ProjektVisualStud. Projekt zawiera następujące pliki:

- `main.cpp` — główny plik z kodem. Tu piszemy nasz program,
- `winbg12.cpp` — plik z funkcjami graficznymi,
- `winbg12.h` — plik z definicjami funkcji graficznych.

Pamiętaj: Często kompiluj projekt i patrz, czy wszystko działa!

Pierwsze kreski

Wewnątrz funkcji `main` wpisz:

```
graphics( 200, 200);  
line( 0, 0, 200, 200);  
line( 100, 100, 100, 0);  
wait();
```

Uwaga: Pamiętaj o średnikach!

Skompiluj i uruchom program. Pierwsza linia tworzy okno grafiki, dwie następne rysują linie, zaś ostatnia czeka z zamknięciem okna na naciśnięcie dowolnego klawisza.

Ćwiczenia

Używając funkcji `line(x1, y1, x2, y2)` i `circle(x, y, r)`, wykonaj następujące zadania:

- Zidentyfikuj, jak ułożony jest układ współrzędnych (X, Y) w oknie.
- Narysuj kwadrat.
- Narysuj ludzika.
- Narysuj koła olimpijskie.

Zmienne

Pewne powtarzające się parametry (jak pozycja, promień, itp.), możemy zastąpić zmiennymi. Wykorzystując je, możemy wyliczyć odpowiednie współrzędne np:

```
int r, h;  
r = 50;  
h = 100;  
  
line(10, 0, 0, h);  
line(10, 0, 2 * r, h);  
circle(10 + r, h, r);
```

Możemy używać wszelkich działań: `+`, `-`, `*`, `/`, oraz funkcji matematycznych `sqrt()`, `sin()`, ..., które są dostępne po dołączeniu biblioteki `math.h`.

Pamiętaj: Pierwsza linia deklaruje zmienne. Trzeba zadeklarować wszystkie zmienne, których będziemy używać! (szczegóły, na kolejnych zajęciach)

Zauważ, że wartość zmiennej jest nadpisywana, możemy więc napisać:

```
int w;  
w = 50;  
circle( 10, w, 10);  
w = w + 20;  
circle( 10, w, 10);  
w = w + 20;  
circle( 10, w, 10);  
w = w + 20;  
circle( 10, w, 10);
```

W efekcie, wyświetlą się cztery kółka narysowane koło siebie. Przetestuj.

Ćwiczenia

Każdy program przetestuj dla paru ustawień zmiennych, by zobaczyć czy działa poprawnie.

- Napisz program, który dla zmiennych x , y , s tworzy okno o rozmiarach x , y i na środku narysuje koło o promieniu s .
- Dla zmiennej d , narysuj cztery dotykające się koła o średnicy d w prawym górnym rogu okna.
- Dla zmiennej y narysuj koła olimpijskie w odległości y od górnej krawędzi.
- Skopiuj poprzedni kod trzy razy i w każdym fragmencie zmodyfikuj wartość zmiennej y .

Pętle

Pierwszą automatyzacją są pętle. Pętla wykonuje pewną operację, dopóki pewien warunek jest spełniony, np:

```
int x;  
x = 0;  
  
while (x < 200) {  
    line(x, 10, x, 190);  
    x = x + 10;  
}
```

Taki program będzie wykonywany w następujący sposób:

- wpisujemy 0 do zmiennej x ,
- sprawdzamy czy $x < 200$ i jeśli warunek jest spełniony przechodzimy dalej,
- rysujemy linię,
- zwiększamy zmienną x o 10,
- i znów: sprawdzamy czy $x < 200$,
- rysujemy linię,
- zwiększamy zmienną x o 10,
- sprawdzamy czy $x < 200$,
- rysujemy linię,
- ...
- gdy wreszcie x przekroczy 200, pętla się skończy i program pójdzie dalej.

Ostatecznie program narysuje pionowe kreski dla kolejnych $x = 0, 10, 20, \dots$

Zauważ: Program nie narysuje linii dla $x = 200$, bo komputer najpierw sprawdzi, że $x < 200$ i przerwie pętlę.

Ćwiczenia

- Napisz program, który narysuje kratkę z odstępem w .
- Narysuj rząd stycznych do siebie kół o promieniu r , zaczynając od lewej strony.
Przemyśl: jeśli x to pozycja środka koła na osi X , to:
 - jaka wartość powinna jej być przypisana przed pętlą,
 - o ile powinna być zwiększana,
 - jaki warunek musi spełniać, aby nie rysować poza oknem?
- Pisząc jedną pętlę w drugiej, zapełnij cały obrazek przylegającymi kółkami.
Czy da się je lepiej upakować?
- Narysuj wiersz kółek, których promienie zmniejszają się jak $\frac{1}{n}$.