

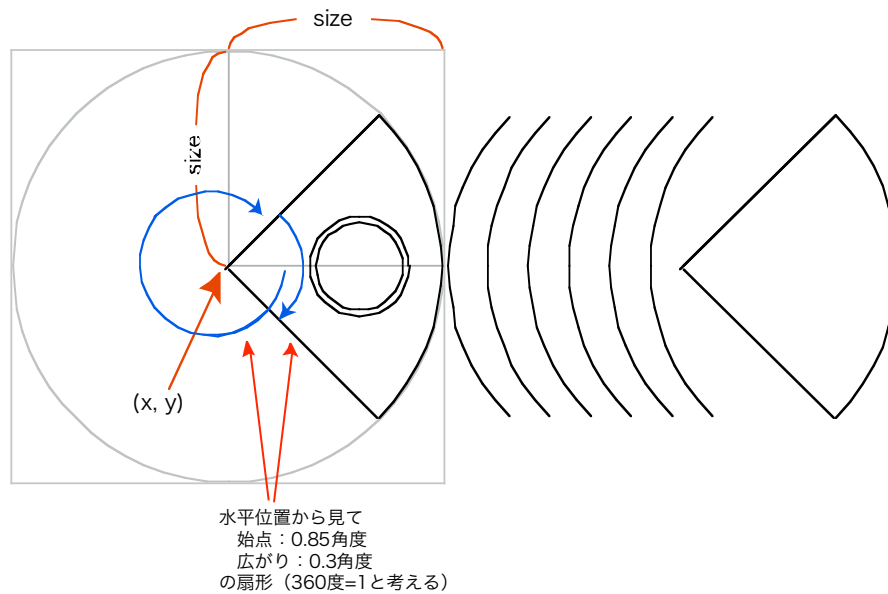
■ アニメーション

これ以降は自分の作りたい絵柄を最終週までに仕上げるつもりで取り組んで下さい。

サンプルプログラムについてそれぞれポイントをまとめておきます。参考にしてください。プログラムを見比べながら、試しに手元で実行してみると良いでしょう。

○ fish.c サカナの絵を一つ描く

単に一つ絵を描くだけです。



基本的に扇形と円からできています。

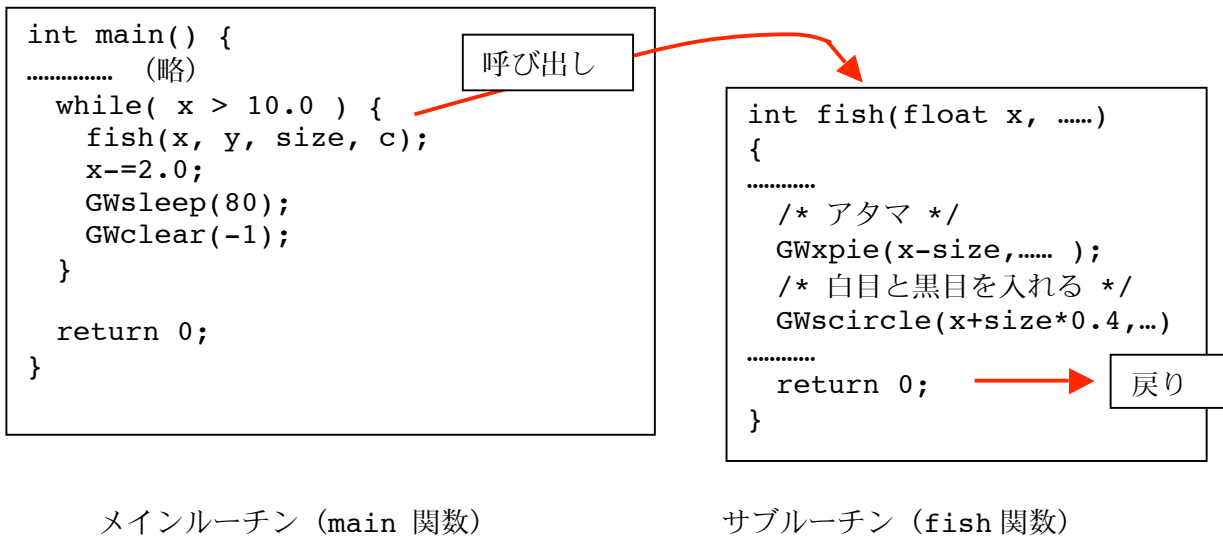
○ fish2.c サカナを右から左へ移動させる

ループによって x 座標位置を少しずつ移動させてサカナを描き直します。

すぐに描き直すと早すぎるので、毎回 `GWsleep(80);` によって少し (80 ミリ秒) 待ち、`GWclear(-1);` によって画面を消去してからループするようにしています。

x の値をチェックし、左に寄ったら (例では x が 10 以下になったら) ループを終了します。

○ fish3.c サカナを描く部分をサブルーチンとして fish() 関数を別に作る



処理の流れ :

- ・ プログラムは main() の上から順に実行される。
- ・ main の途中で fish() 関数を呼び出すと、
- ・ fish 関数に処理が移る。
- ・ サカナを描いて、
- ・ return でメインルーチンに処理が戻る
- ・ そのまま main の処理を続行する。

○ fish4.c 乱数を使って登場位置 (上下) をランダムに決める

random( ) 関数は、呼び出すたびに 0 から非常に大きな整数のなかから、ばらばらの数を返します。つまり `i=random( ) % 100;` とやれば、整数変数 i には 0 から 99 までのランダムな値が設定されます。( % は余り (剰余) です。)

ただし random( ) 関数を呼び出す前に、srand( ) 関数にランダムな数値を入れて一度だけ呼び出す必要があります。この srand( ) 関数に与える引数を「seed (種)」と呼んでいます。つまり random( ) 関数は srand( ) に与えられたタネをもとに、それ以降乱数を発生させ続けているわけです。

サンプルでは srand( ) 関数に入れる seed として、何か scanf で入力させています。

○ fish5.c 乱数を使って登場位置 (上下) を毎回変える

左端に到達したとき、x を右に戻すだけでなく、y の次回登場場所を乱数で決める。このために while によるループを `while(1) { }` による無限ループとしました。プログラムは終了しません。

○ fish6.c 配列を使って多数のサカナを登場させる

サンプルでは 10 匹のサカナをだしてみました。

配列については別紙を見て下さい。この例では従来 x, y が縦横の座標位置を決めていましたが、それを `float x[10], y[10]` として x[0]から x[9], y[0]から y[9] の 10 セットずつ用意しました。for() によるループで、それらをつつ繰返して処理しています。