

## 条件分岐 (つづき)

### ■ 無限ループと if, break

サンプルプログラム kotil.c をまず実行して出力結果を確認せよ。白い口が指定した角度に少しずつ移動しながら描かれるはず。

```
#include <eggx.h>
#include <math.h>

int main() {
    int win, d;
    double r,x,y,dx,dy,w;

    printf("Direction? (0-360) : ");
    scanf("%d",&d);
    win=gopen(400,400); /* 描画ウィンドウを開く */
    winname(win, "koti"); /* 名前をつける */
    r= 3.14159265 / 180.0 * d ; /* radian 単位に変換 ( 30° は 1/6π ) */
    /* 四角の幅を設定 */
    w=20.0;
    /* 初期位置を 200, 200 に設定し、増分値を算出 */
    x=200.0;
    y=200.0;
    dx=w * 0.5 * sin(r);
    dy=w * 0.5 * cos(r);

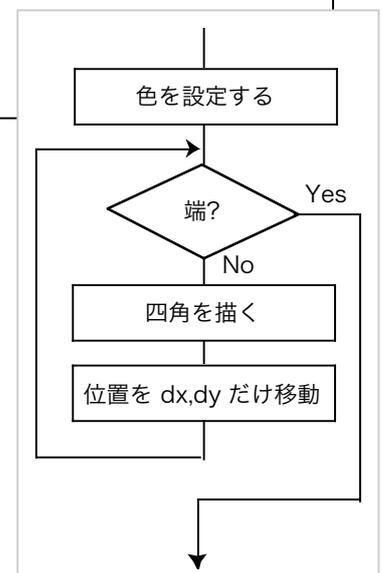
    newpen(win, 1); /* 色を白に設定 */
    while(1) {
        if( (x<0.0)||((x+w)>400.0)|| (y<0.0)||((y+w)>400.0) ) { /* 端に到達 */
            break; /* while ループから脱出 */
        };
        fillrect(win, x, y, w, w);
        x+=dx; y+=dy;
        usleep(50000); /* 少し待つ */
        gclr(win); /* 画面を消去 */
    };

    drawstr(win,300.0,30.0,16,0.0,"FIN"); /* 300,30 の位置に 16 サイズの文字 */
    ggetch(win); /* キー入力を待つ */
    gclose(win); /* 描画ウィンドウを閉じる */
    return(0);
}
```

このプログラムの考え方：

1. x,y という変数で座標位置を覚えておき、
2. dx, dy という変数で移動距離を横軸、縦軸方向ともに覚えておき、
3. それを使い、少しずつずらして口を描き、
4. 上下左右のどこかの端に到達したら終了する。
5. 終了したら FIN と目印を出す
6. キーが押されるのを待つて終了

毎回画面を消去(gclr関数)している点に注意。



前回同様のポイントは：

- while() によるループ（教科書 pp.130）があるが、それは条件節に 1 と書かれた無条件ループ（無限ループ）である。（教科書 pp.133）
- 右端に到達したかどうかをif文（教科書 pp.112）によって毎回確認し、ループを終了（脱出）させる。
- while() ループからの脱出には break を使う。（教科書 pp.144）
- usleep() によって 50ms 待ち時間が入っている。（そうしないと速すぎて面白くない）

今回新しく加わったポイントは：

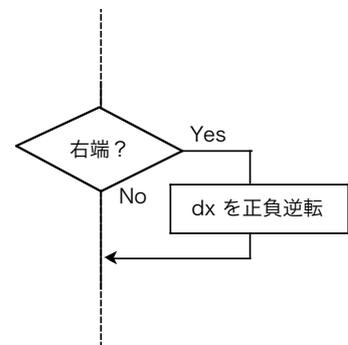
- 画面を毎回消去するために glcr() 関数を使っている。
- 終わった事を示すために文字を表示させる。  
`drawstr(win, 300.0, 30.0, 16, 0.0, "FIN");`  
は、300,30 の位置に、16 の大きさと、0度の傾きで FIN という文字列を描く、という意味。
- if 文に or 条件の論理演算子「||」がある。「どれかひとつでも条件が成立すれば」というもので、教科書 pp.122 参照。

### ■ 課題1.

描いている口が壁に当たったら跳ね返るようにしてください。

考え方：

- 少しずつ右に進む場合は dx は正の値、左に進む場合は負の値になっているはず。
- dx が 1.4 だったとして、いつか右端に到達した場合、dx は -1.4 になれば跳ね返っているように見えるはず。
- 縦軸方向についても同じ。



永遠に終わらないプログラムになるかも知れないが、それで構わない。

強制的に実行を終了させるときは、コマンドプロンプトが出ているウィンドウを表に出して、Control-C（Control キーを押しながら C キーを押す）で停止する。

できあがれば講師に見せてOKが出たら課題提出システムで登録してください。

### ■ 課題2.

跳ね返るたびに口の色を変えるようにして下さい。

考え方：

- 色を保持する変数をひとつ作成し、初期値を与え、跳ね返るたびに色を変更していけば良いでしょう。

可能なら 5 回跳ね返ったら終了するようにしてください。

できあがれば講師に見せてOKが出たら課題提出システムで登録してください。

受講生に提案があります。

プログラムの一行一行を、どういう意味があって、どういう振る舞いをするものなのか理解しながら進めて下さい。前回、色を変えて下さいと指示したら

`newpen(win, 1-15)` や `newpen(win, 1,2,3,4,5...,15)` などと書いて試している受講生がいました。

プログラムは願望を書くものではありません。相手がコンピュータであることを忘れずに。コンピュータに完全に理解できる方法で動作を明確に指示しなければなりません。