

ものづくり基礎工学

情報工学分野

J2. 動きのあるグラフィックス

前回のまとめ（1）

- ◆ 基本的な図形を描いた
長方形（正方形），三角形，楕円（円），線
塗りつぶしや線のみ
- ◆ 座標で位置 (x,y) を指定した
数学の座標と同じ
図形によって指定が若干異なった
- ◆ 色の指定をした
光の3原色「RGB（赤，緑，青）」で

前回のまとめ（2）

- ◆ 書いた順番にプログラムが実行される
- ◆ ソース中では改行などが無くても実行は可能

見づらいので改行，タブなど
を利用し見やすく

+

さらに“コメント”を付ける
(何をしている部分か)

```
/* ここに文 */
```

プログラム
は誰にでも
分かる様に
書くことも
重要

前回のおさらい（課題1）

- ◆ リスト2.3.1（kadai1.cpp）を実行してみよう。左右の目の位置が対称になるように調整してみよう
- ◆ ヒント：左目のx座標を調整する。
（右目のx座標の符号を逆にしたものを左目に利用すればよい）

今回すること

- ◆ 変数を利用する
代入, 四則演算

プログラミング特有の表現

- ◆ 条件文
if や if~else if

ウィンドウを移動させても表示されていたのはそのおかげ

- ◆ 実は, 前回皆さんが作ったプログラムは「**繰り返し実行**」されている. 座標が固定なので, わからなかっただけ.

20[ms]ごとに実行されている

- ◆ この繰り返しを利用し, 座標を変化させてやれば?

まずはやってみよう

リスト2.3.2 (kadai2.cpp)

```
void userdraw(void)
{
```

最初に変数を宣言する

記憶する箱を用意。ここでは整数(int型)を入れる箱

```
int stepx;
```

箱の名前をつけ準備 (実は中身はわからない)

```
static int movex=0;
```

箱の名前をつけ、さらに0を入れておく。

```
stepx=2;
movex=movex+stepx;
```

箱の中身は次の繰り返しのときに残らないが、これを書くと残しておいてくれる

```
glTranslatef(movex,0.0, 0.0);
```

値の代入

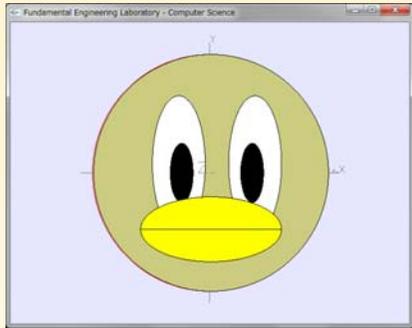
- stepxに2を入れる
 - 今のmovexの中身とstepxの中身を足してmovexに入れる
- 1回目：2=2+0 movexが増えている
2回目：4=2+2

以下で描く図形の移動
(x軸方向, y軸方向, z軸方向)
ここではx軸方向にmovex移動という意味

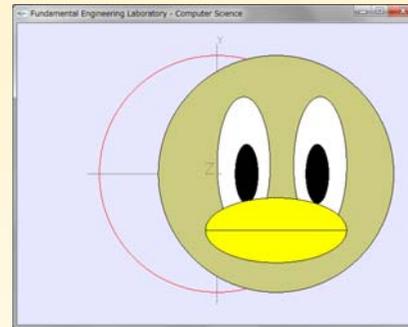
以下は顔の描画の部分

知らないうちに繰り返し実行されている

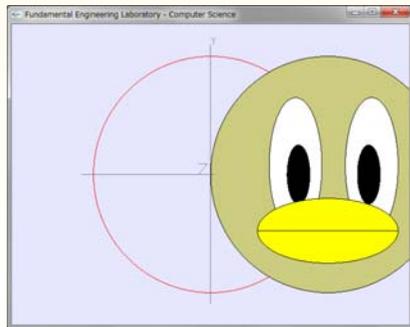
実行してみよう



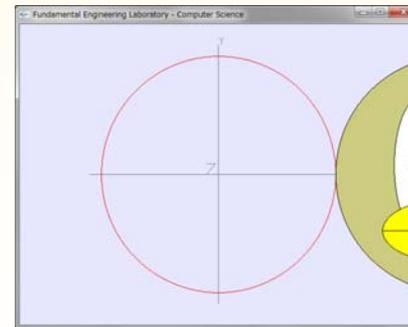
- $stepx=2$
- $movex=2$



- $stepx=2$
- $movex=100$



- $stepx=2$
- $movex=200$



- $stepx=2$
- $movex=400$

最後は見えなくなる

ここでのおさらい

- ◆ 最初に宣言が必要
型の指定や箱の名前をつける
- ◆ 最初に利用する前に何らかの値を代入する
→宣言しただけではゴミが入っている
- ◆ 変数は数値と同じように利用できる
- ◆ 四則計算に利用できる：数値だけでも可
+：加算 -：減算 *：乗算 /：除算
計算の規則は数学と同じ，()も使用できる
- ◆ 代入とは「右辺の（計算された）値を左辺に入れる」
○ $x=x+25$ や $x=y$
× $x+25=y$ や $12=y$

移動方向を変えてみよう（課題2）

◆ プログラムの

```
glTranslatef(movex, 0.0, 0.0);
```

のmovexの場所を変更してみよう

```
glTranslatef(0.0, movex, 0.0);
```

```
glTranslatef(0.0, 0.0, movex);
```

のように変更してみよう。

movex=2の数値も変更してみよう。

画面から消える！

- ◆ 移動の方向が一定(moveが単調に増加)で描画範囲を超えてしまう.
- ◆ 超えないようにするためには
ある程度動いたら
 - (1) 止める
 - (2) 元の位置に戻す
 - (3) 方向を反転させる
- ◆ もし「動いた量 (movex)」が「ある条件」になったら「こうする」という処理を利用する.

実行してみよう（右で跳ね返る）

リスト2.4.3 (kadai3.cpp)

他にも
<, ==, !=
などが使える

```
void userdraw(void)
{
    static int stepx=2;
    static int movex=0;

    if(movex>50){
        stepx=-2;
    }

    movex=movex+stepx;

    glTranslatef(movex,
```

もし(movexが50より大きくなったら)
stepxの中身を-2にする（元々は2）

すなわちmovexが

①2,4,...,50と増えて行く。

②52になったら、次の繰り返しの時にif文の条件が成り立ち、stepx=-2になる。

③するとこの式に来た時に

50=52-2

48=50-2

46=48-2

と、どんどん減っていく

以下は顔の描画の部分

知らないうちに繰り返し実行されている

if ~else ifを利用し左右で跳ね返る ように改良（課題3）

- ◆ 実行すると、右で跳ね返るものの消えてしまう。
これを左右両方で跳ね返るようにしよう。

```
if (movex>50){  
    stepx=-2;
```

条件（1）
右で跳ね返る

```
}else if(条件式を記入){  
    そのときの処理を記入;  
}
```

条件（2）
左で跳ね返るよう
にするには

ヒント：

右で跳ね返るのと同じような処理。

（movexがあるところまで減ると、再度stepxが増えるように）

実行してみよう (投げあげ後に落下)

リスト2.4.5 (kadai4.cpp)

```
static int stepy=10; /* 最初の中身は10 */
```

```
static int movey=0; /*最初の中身は0 */
```

```
stepy=stepy-1;
```

• stepyは繰り返すたびにどんどん小さくなる
9,8,...,1,0,-1,... 単調な減少

```
movey=movey+stepy;
```

• moveyは
上に動くとき：だんだん動きが遅くなる
下に動くとき：だんだん動きが早くなる

```
glTranslatef(0.0, movey, 0.0);
```

以下は顔の描画の部分

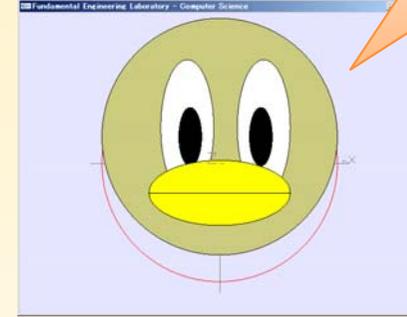
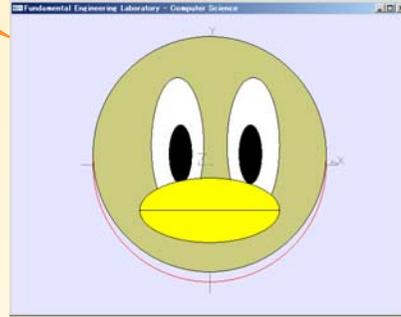
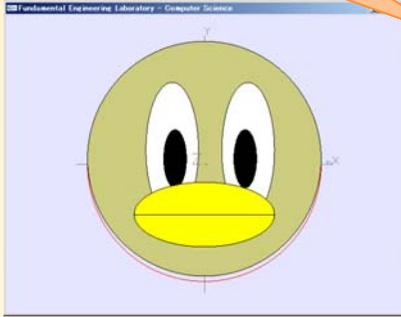
移動を y 軸に

知らないうちに繰り返し実行されている

実行してみると

最初の動き
は速い

だんだん
遅く



最初

- **stepy=9**
- **movey=9**

2回目

- **stepy=8**
- **movey=17**

9回目

- **stepy=1**
- **movey=45**

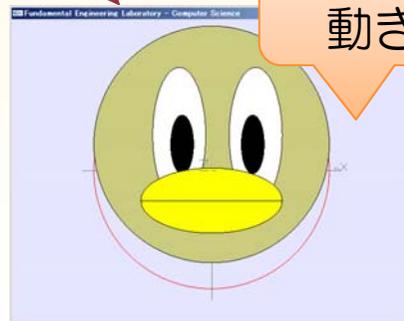
止まる



10回目

- **stepy=0**
- **movey=45**

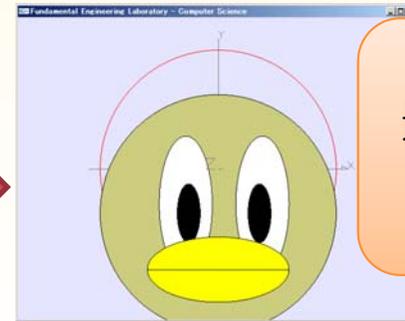
下に
動き出す



11回目

- **stepy=-1**
- **movey=44**

最後は見え
なくなる



下で跳ね返るように 改良してみよう（課題4）

リスト2.4.6 (kadai4.cpp)

```
static int stepy=10; /* 最初の中身は10 */  
static int movey=0; /* 最初の中身は0 */
```

```
stepy=stepy-1;  
movey=movey+stepy;
```

```
if (movey<-100) {  
    stepy=??????;  
    movey=-200-movey;  
}
```

```
glTranslatef(0.0, movey, 0.0)
```

以下は顔の描画の部分

知らないうちに繰り返し実行されている

①moveyが-100よりも小さくなった場合
例えば, movey=-108になったとすると

②stepyの値を反転（増減（移動方向）
を反転）させる。
stepyは条件を満たすと
正→負へと反転を繰り返す

③moveyの値も調整
（y=-100で対称的な跳ね返りとなる
ようにしている）
例えばmovey=-108となっていた時,
movey=-200-(-108)=-92となる

表を使って数値の変化を確認

- ◆ リスト2.4.6 (kadai4.cpp) の `movey=movey+stepy;` 行に着目し、表2.4.3を埋めて完成させよ。
- ◆ 注意：if文の中に入る条件と実行される式に気をつけること。

◆ ヒント：

◆ 1回目	9	0	9
◆ 2回目	17	9	8
◆ 3回目	24	17	7
◆ 10回目	45	45	0
◆ 27回目	-108	?	?
◆ 28回目	-76	?	?
◆ 35回目	8	?	?
◆ 50回目	23	?	?

お昼の休憩時間に向けての注意

- ◆ PCは電源を切ると、作成したデータも消えますので、必ずUSBメモリに保存しておくこと。
- ◆ 自分の荷物・貴重品は置いたままにしないこと。
- ◆ 12:50までに戻る。各自午後の準備をしておく。

課題（跳ね返りの拡張）

- ◆ 課題5：kadai4.cppをコピーし、kadai5.cppとする。改良し、上下左右に弾むようにプログラムを修正してみよう。

※ヒント：上下は重力が働くが、左右は働かない。
課題4と課題3の処理を合わせる。

- ◆ 課題6：kadai5.cppをコピーし、kadai6.cppとする。改良し、上下左右前後に弾むようにプログラムを修正してみよう。

※ヒント：前後の動きは、課題3と同様の動き。
すなわち、xの動きと同じようにzの動きを追加する。

発展課題

- ◆ 課題7：リスト2.5.1 (kadai7.cpp) を実行してみよう。
 - 「glRotatef(angle,0,0,1);」の(angle,1,0,0)や(angle,0,1,0)とするとどうなるか確認せよ。
 - 「*」キーを押すたびにどうなるか確認せよ。
 - 「l」キーが押されたら左回転,
「r」キーが押されたら右回転
するようにプログラムを改良せよ。

※ヒント：課題3の時と同様に、2つの条件の処理で「l」「r」を利用する。

回転方向によってangleの増減考える。
(符号を考えて見ると分かる)

チャレンジ課題

- ◆ 課題8：時間になったら提示します。

※早く進んでいる人は、テキストを見るなどして、自分でCGを作ってみましょう。