

この時間のねらい：① キーボードの操作により、実行画面での操作をできるようになろう。

前回はマウス操作による反応のさせ方を学びました。今回はキーボードを操作することで、処理を行えるようにしたいと思います。

前回と同様にキーボードの情報を格納するための変数をまとめましょう。

keyPressed	____が押されているかどうか
key	____の情報を格納
keyCode	____の名前を格納

キーボード操作について、実行するためのプログラムは下の画像のようになります。

```

1  if(keyPressed){
2      if(key == 'アルファベット一文字を記述') {
3          /* 処理内容1 */
4      } else if(key == CODED) {
5          if(keyCode == "特別なキーの名前を記述") {
6              /* 処理内容2 */
7          }
8      }
9  }
```

見た感じはどうでしょうか。マウスの反応のさせ方と似ている部分があります。探してみてください。

ここで、特別なキーの名前を記述するところに入れるものを以下にまとめておきます。

←半角スペースを空けた状態です。	スペースキーに反応
UP	上矢印キーに反応
LEFT	左矢印キーに反応
RIGHT	右矢印キーに反応
DOWN	下矢印キーに反応
ENTER	Enter キーに反応
SHIFT	Shift キーに反応

\_\_\_\_\_に関しては注意が必要です。他はそこまで難しくありません。矢印キーなどはゲームにて操作を行う際に、使う人も多いと思いますので覚えておくといいでしょう。

以上のように記述すると、キーボードを使用して処理を行うことができます。例えば、FPS などでは、WASD キーにて移動をさせることが多いのではないのでしょうか。私は FPS はやらないのでわかりませんが、友達がやっていたものはそのようになっておりました。このような設定を行うためには、以上の変数などを使用することで、設定を行うことができます。また、音ゲーとして、有名(と思います。)な osu! に関しても、基本的にはキーのためこのような設定がされています。

それでは今回の演習問題に入ります。

#### 演習 1

いつものようにウィンドウ表示と円を表示させましょう。

このとき、前回学んだ `setup()`関数と `draw()`関数を用いて記述を進めてください。

(これらの関数を利用しないと条件文は実行されないので注意です。)

#### 演習 2

1 で表示した円を、キーボードの矢印キーを利用して、動かすことができるように改良しましょう。また、背景に円の軌跡が残らないように一文を加えてください。

#### 演習 3

前回、マウスのクリックを行った際に色が変わったのと同じように、以下の条件に合うように条件文に書き足してください。

押すキー	図形の色
y	黄色
b	青色
r	赤色
押さない	白色

押すキーの決め方は安直ですが、色を英語にした時の頭文字です。

まあ、それぐらいの方が簡単だし楽なので、皆さんも変数を決める際には、そのように決めて十分だと思います。

前回と同じで押していない状態のとき、白色になることにも注意しましょう。

#### 演習 4

後ほど Web 上に解答を上げますが、一応、先ほどのプログラムは保存しておくといわれています。この問題は新しいファイルを使用するので、上のファイルから新規を選び新しく書くことができるように準備してください。

準備ができたなら、ウィンドウを作成し、それにキーボードで押した場所のアルファベットを表示させてください。