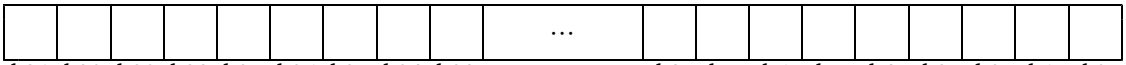
iアプリ Java
ゲームプログラミング

○　第９回　キーパッド状態の取得　○

キーパッド状態の取得

携帯電話のボタン(キーパッド)の状態は、キャンバスクラスのgetKeypadStateメソッドで取得します。このメソッドを呼び出すと、それぞれのキーが押されているかどうかをint型の値で返します。Javaのint型は３２ビットあり、それぞれのキーに割り当てられたビットが押されていれば１になり、押されていない場合は０になります。



たとえば、キー０に対応するビットは、最下位のビット０(b0)で、キー１に対応するビットは、その隣のビット１(b1)というように、それぞれのビットにキーが割り当てられています。
どのキーが押されているか(または押されていないか)を調べるには、論理演算やシフト演算を行います。キーに対応した値がDisplayクラスで定義されており、「1 << (Display.キーの値)」といったように、この値をそのままシフト演算に用いることが出来るようになっています。
下記の表のとおり、キー０は「Display.KEY_0」、キー１は「Display.KEY_1」とわかりやすい名前がつけられています。

キー	キーの値	値	備考
数字０	KEY_0	0x00	b0
数字１	KEY_1	0x01	b1
数字２	KEY_2	0x02	b2
数字３	KEY_3	0x03	b3
数字４	KEY_4	0x04	b4
数字５	KEY_5	0x05	b5
数字６	KEY_6	0x06	b6
数字７	KEY_7	0x07	b7
数字８	KEY_8	0x08	b8
数字９	KEY_9	0x09	b9
＊	KEY_ASTERISK	0x0a	b10
＃	KEY_POUND	0x0b	b11
◁	KEY_LEFT	0x10	b16
△	KEY_UP	0x11	b17
▷	KEY_RIGHT	0x12	b18
▽	KEY_DOWN	0x13	b19
選択/決定	KEY_SELECT	0x14	b20
ｸﾘｱ	KEY_CLEAR	0x20	getKeypadState(1)

▲よく使うキーとその値

キーイベント

getKeypadStateメソッドでは、押されているかどうかを調べることができますが、ゲームでは、「押された」「離された」という状態を知りたいときがあります。これらの状態を取得するメソッドはありませんが、Canvasクラスでは、イベントという形で知らせるようになっています。

キーが押されたり離されたりするたび、イベントが発生し、自動的にCanvasクラスのprocessEventメソッドが呼ばれるようになっています。

－Canvas#processEvent－

```
public void processEvent(int type, int param)
  type  - イベントのタイプが渡されます
  param - イベントのパラメータが渡されます。ない場合は0が入ります
```

typeには、起きたイベントにより、下記の値が入ります(定義はDisplayクラス)。

KEY_PRESSED_EVENT	キーダウンイベント
KEY_RELEASED_EVENT	キーアップイベント
MEDIA_EVENT	メディアイベント
RESET_VM_EVENT	リセットイベント
RESUME_VM_EVENT	レジュームイベント
TIMER_EXPIRED_EVENT	タイマーイベント
UPDATE_VM_EVENT	アップデートイベント

キーに関連するのは「KEY_PRESSED_EVENT」「KEY_RELEASED_EVENT」で、その場合paramにgetKeypad Stateメソッドでも使ったキーの値(KEY_xx)が入ります。
たとえば、上下左右キーが押されたときに処理を行うためには、下記のようになります。

```
public void processEvent(int type, int param) {
    if(type == Display.KEY_PRESSED_EVENT) {
        switch(param) {
        case Display.KEY_UP:
            /* 上方向キーを押した時の処理 */
            break;

        case Display.KEY_DOWN:
            /* 下方向キーを押した時の処理 */
            break;

        case Display.KEY_LEFT:
            /* 左方向キーを押した時の処理 */
            break;

        case Display.KEY_RIGHT:
            /* 右方向キーを押した時の処理 */
            break;
        }
    }
}
```

デフォルト状態のprocessEventメソッドは「何もしない」ようになっているので、継承したクラスでオーバーライド(上書き)し、定義を変更する必要があります。

processEventメソッドは、キー以外にも、タイマやレジュームなど、さまざまなイベントが起こるたびに呼ばれています。

課　　題

(１)キーパッドの状態によって、キャラクターを移動させましょう。

・ヒント

```
// キーパッド状態取得
int key = getKeypadState();

// 移動
if((1 << Display.KEY_LEFT & key) != 0)   // 左
```

(左が押されている)

(２)キャラクターの移動範囲を画面内だけにしましょう。

・ヒント１

座標は、画像の左上を指しています。

・ヒント２

ｘ座標の最小値は、原点の0です。この値未満になるということは、原点より左側にいるということになり、画面外にいることになります。この場合は、ｘ座標を0にし、画面左に戻します。

・ヒント３

ｘ座標の最大値は、画面右端の480になりそうですが、実は違います。ｘ座標は、キャラクターの左端の座標なので、ここが480の場合は、すでに画面外に出ています。キャラクターの右端(左端 + 幅)が480を越えているかどうかを調べるか、キャラクターの左端が、480から幅を引いた値(480 - 幅)を越えているかを調べます。越えている場合は、ｘ座標を480から幅を引いた値にします。

・ヒント４

ｙ座標も同様に考えます。