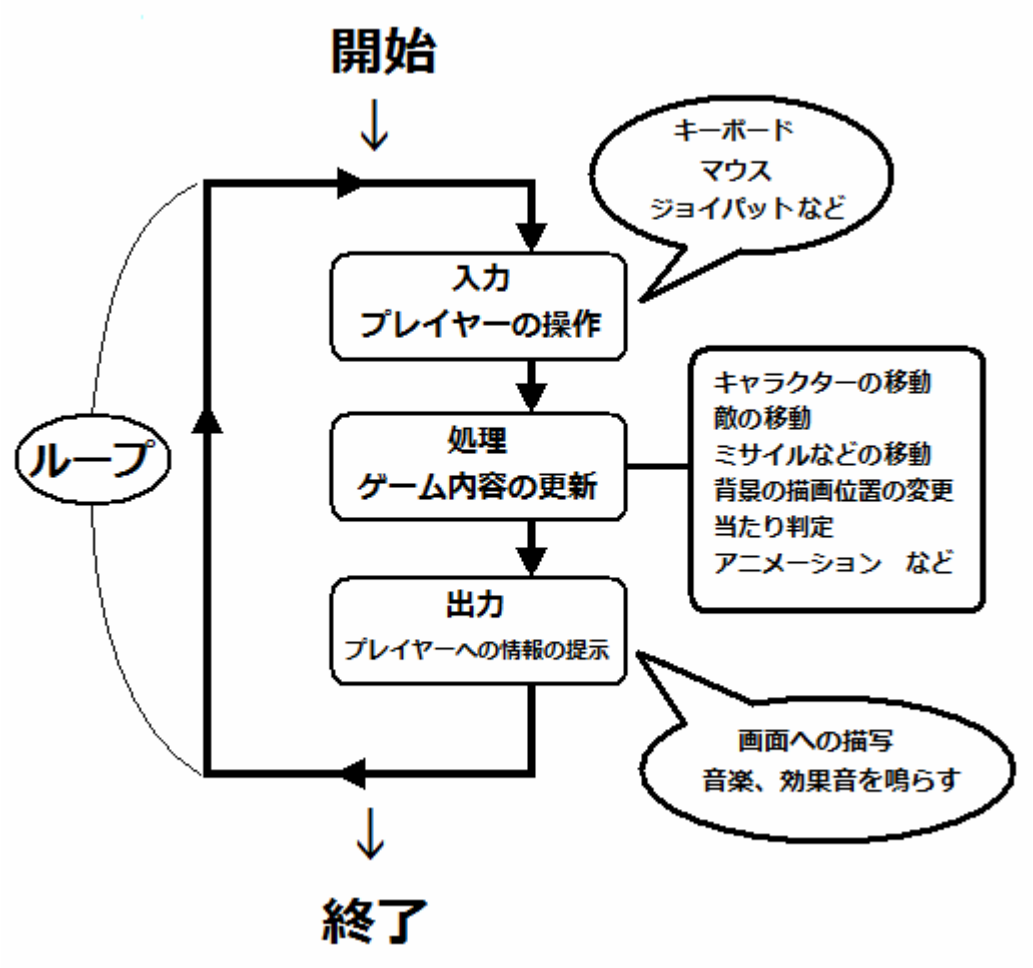
ゲーム製作勉強会　ループ編

ループとは同じ処理をくりかえすこと。（例：for 文や while 文）

ゲームは常にループしています。

ゲームを終了させない限り、処理を行い続けます。

これをゲームループといいます。



ゲームループはプレイヤーとのコミュニケーションを円滑にするための、仕組みのことです。

ＦＰＳについて

通常、ゲームはＦＰＳによって管理されてます。

ＦＰＳ：（ Frame Per Second :単位秒あたりのフレーム数）

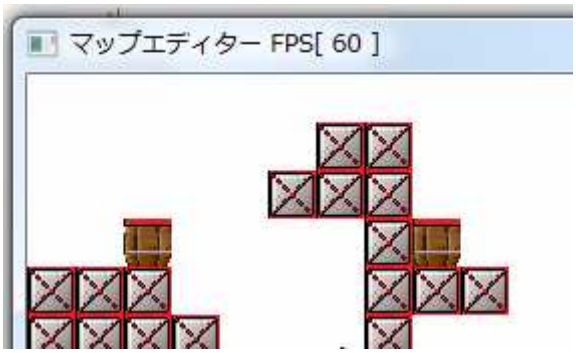
動画のなめらかさなどを表す指標。1 秒間に何枚の画像を表示しているかを示します。

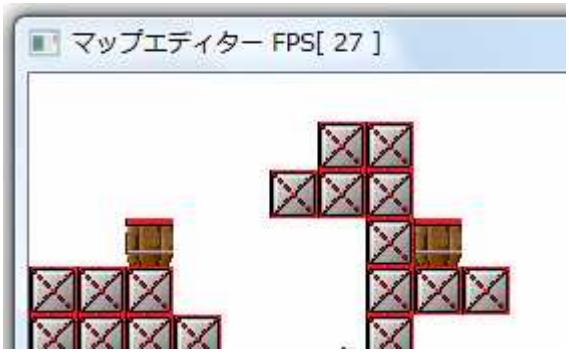
１フレームというのは１ループのことです。

ゲーム内では、一秒間に６０回ループしています。

通常のゲームでは１秒間に 60 回の処理を行うようにしています。なぜなら、ディスプレイ（画面）の更新が 1 秒間におよそ 60 回であるからです。これをリフレッシュレートといいます。

多くの処理をしたりたくさんの描画をしたりすると､処理が重くなりＦＰＳの値が小さくなります。これを処理落ちといいます。





ゲームでＦＰＳ管理を行わないと・・・

パソコンの処理速度によってゲーム自体の速さが変わってしまいます。
つまり、環境によって全く違った難易度のゲームになってしまいます。
（処理速度が速い　→　ゲームスピードが速くなる）
対策として、ＦＰＳの上限を固定するなどの方法があります。

プログラムで見るゲームループ

プログラムの流れ

```
main()
{

      初期化(データの読み込み)

      //ゲームループの開始
      while(1)
      {
          入力(キーボード、パッド)
          ゲーム処理
          出力(画像、音)
      }

       終了処理
}
```

シューティングゲームの場合

```
main()
{
    初期化(自機、敵、弾、ステージなど)

    ///ゲームループ
    while(1)
    {
     (1) 入力
     (2) 自機移動
     (3) 自機からの弾の発射
     (4) 敵の移動、弾の発射
     (5) 弾の移動(自機、敵)
     (6) 自機の弾が敵に当たっているか調べる
     (7) 敵の弾が自機に当たっているか調べる
     (8) 画面に出力、音を鳴らす
    }
    終了(スコアなどの保存など)
}
```

基本的に(1)～(8)までの処理を繰り返します。

もう少し具体的な例

```
main()
{
    初期化(データの読み込み)
   //ゲームループ
    while(1){
         PDRECT Src,Dst;
         static float  x = 128,y = 480 -128;

            // 右を押していたら右に進む
         if(PDInput::GetKeyData( DIK_RIGHT ) ){
                 x = x + 1;
             }

         // プレイヤーを描画する
         Src.Set(0,0,128,128);
         //Src.Set(画像をどこから切り取るかX,Y,画像をどんな大きさで切り取るかX,Y);
         Dst.Set(x,y,128,128);
         //Dst.Set(画像をどこに表示するかX,Y,画像をどんな大きさで表示するかX,Y);
         PDSprite::Draw(TestGraph,&Src,&Dst);
     }
     終了処理
}
```

この例は部内で使用しているＰＤライブラリを使っています。

<発展>

ゲームの状態にはゲームメイン、タイトル、メニュー、オプション、ポーズなどがあり、それぞれに対する処理が必要です。

ゲームの状態を表す変数を作って処理を分けると良いです。

```
void Game()
{
         static int GameState = 0;
         if(GameState==0){
                  //タイトル画面を表示
         }else if(GameState==1){
                  //メニュー画面を表示
        }else if(GameState==2){
                  //ゲーム画面を表示
         }else if(GameState==3){
                  //ポーズ画面を表示
         }
}
```