

NINTENDO DS ゲームプログラミング

第4回 メインループ

メインループ

- ・ゲームの仕組みはセルアニメーションと同じ
- ・1秒間に数十回程度のループを行う
- ・ループ内でゲーム固有の処理を行い、画面に描画する

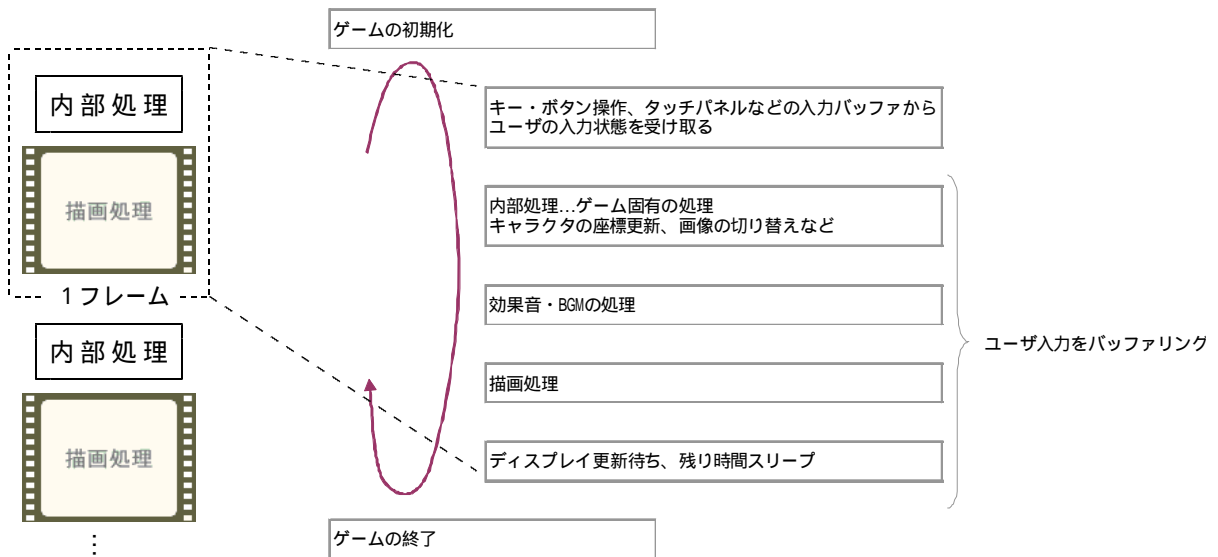
概要

ゲームプログラムは、キャラクタや背景など、いろいろな「もの」の状態を時間にそって更新し、画面に描画するプログラムと考えることができます。これを非常に単純化すると、

- ・「もの」の状態を更新(内部処理)
- ・「もの」を画面に描画(描画処理)

の2つにまとめることができます。この処理の繰り返しメインループになります。ループ1回ぶんを「1フレーム」と呼びます。1フレームごとに、キャラクタの位置や状態、背景やそのほかの表示状態などを更新し、その状態を反映して描画を行う、という処理を繰り返しています。

簡易図



フレームレート

- ・1秒間あたり何度画面が更新されるかを表す
- ・単位はfps(Frames Per Second)
- ・ディスプレイではリフレッシュレートと呼び、単位はHz

実例

- ・NDSは最大60fps = 1ループは約16ms。
各種処理に10ms 6msスリープ 各種処理に20ms スリープなし、処理落ち発生
- ・コンシューマ、アーケードは30または60fps
- ・携帯アプリは20~30fps
- ・映画は24fps、テレビアニメは8fps

メインループを作成しましょう。

(1)以下の各種関数を作成します。

- ・初期化.....Init関数。初期化のための処理を記述します。
- ・内部処理...Exec関数。キャラクタの座標やゲームの状態を変更するための処理を記述します。
- ・描画処理...Draw関数。内部処理で変更された状態を画面に表示するための処理を記述します。
- ・終了処理...Term関数。終了のための処理を記述します。

上記の関数をarm9_main.cの適切な場所に追加しましょう。

- Init関数 -

```
//-----
// Program initialize
//-----
void Init(void) {
}
```

- Term関数 -

```
//-----
// Program terminate
//-----
void Term(void) {
}
```

- Exec関数 -

```
//-----
// internal processing
//-----
void Exec() {
}
```

- Draw関数 -

```
//-----
// draw processing
//-----
void Draw() {
}
```

(2)main関数を以下のように変更します。なお、プログラム中の"?"は適切な語句にしてください。

- main関数 -

```
//-----
// Program entry point
//-----
int main(void) {
//-----

    ????();

    while (1) {
        ????();
        ????();
        swiWaitForVBlank();
    }

    ????();
    return 0;
}
```

(3)すべて完成したら、makeを行いエラーがないか確認し、プログラムを実行しましょう。