■#08_手続き型処理とポーリング処理

今まで、コンピュータがどうやって処理しているのか勉強してきました。その中で

「基本的には CPU は順番にしか処理できない」

と言う話をしました。実際には並列処理やマルチコアなどやっているのですが、そもそもその並列処理などはどのように実現しているのでしょうか？

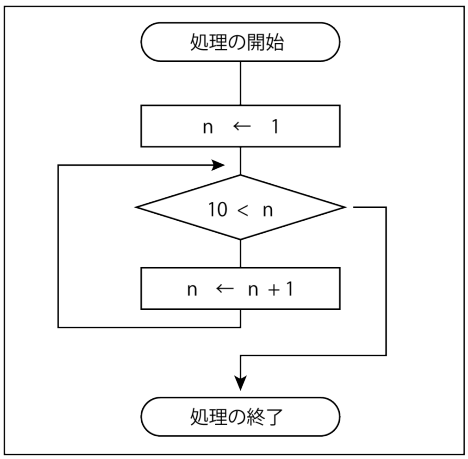
今日は、その事について勉強してみましょう。

1. 巡回型処理「ポーリング」

プログラミングの勉強をする上で、みなさん 「フローチャート」 と言う言葉聞いたことあるでしょうか？アルゴリズムの時間にも少し話あり、問題にも出たはずですがこのプローチャートを使って話をしてみます。

1.1. 一般的な「フローチャート」

フローチャートは 「流れ図」 とも呼ばれています。手続き型の処理をする上で、プログラム全体の流れを図形化できるため、広く使われていた図です。



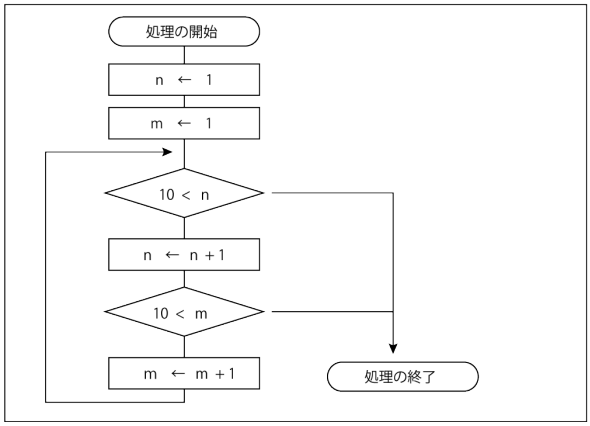
左にあるのがそれで、

1. 変数 n に 1 代入
2. 変数 n が 10 以上になったら終了、
3. そうで無ければ変数 n に 1 を加算
4. 2.に戻る

という一般的な処理になります。このように上から下に向かい流れるように処理することを 「手続き型処理」 と言い、プログラミングの基本になります。しかし見ての通り、この処理を行っている間他の処理はできません。

1.2. 複数の処理をするためには？

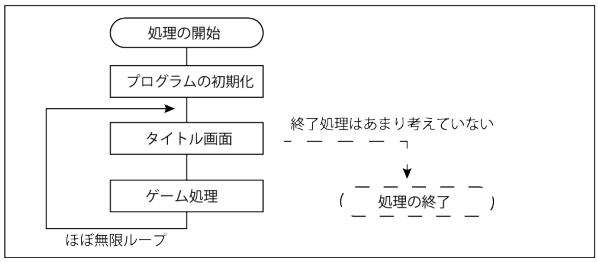
複数の処理をするためには、このループの中にいくつかの処理を組み込めば大丈夫そうですね。



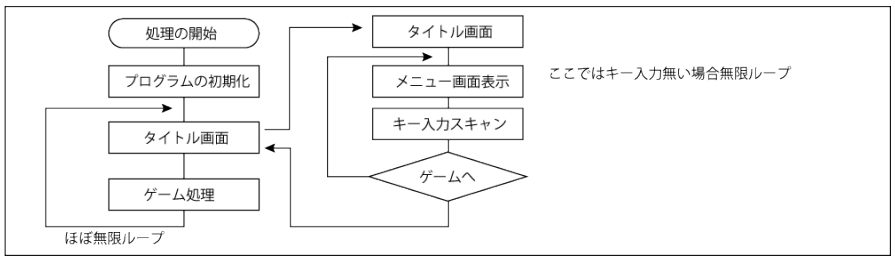
これで、変数 n の処理以外に、変数mの加算処理が組み込むことができました。このようにループの中で順番に処理することを 「ポーリング」 と言います。

2. ゲームでの処理

ゲームでは、ゲームが終了してもタイトルに戻りますので、プログラムは動いています、終わるときは電源を切ったり、プロウラム全体を終了したりすることになりますので、基本的に無限ループになります。

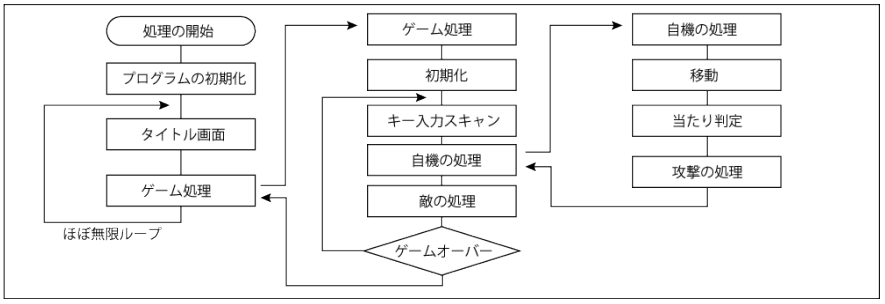


皆さんがプログラミングする環境で、処理を記述する際、よく「Update 関数」と呼ばれる関数があると思いますが、これがループの中で毎回呼び出される関数になります。また、四角い処理が「関数」になると考えて下さい。



すこし大雑把な図ですが、各処理でもこれは同じで、上の例だと、タイトルの処理の中でもループがありますね。ゲームの処理はどうでしょう。

敵も同様な考え方になります。



このように、処理ごとに作業を分け、関数化し、処理順番を組み立てることで、ゲームが動きます。

3. じゃ、全部ポーリングでよくね？

と、思いますよね、実は動作させる対象が多くなると、この処理では煩雑になる可能性があります。「手続き型プログラミング」 は処理の基本ですが、想定外の処理が入った場合や、思った以上に各処理に時間がかかった場合全体の動作が止まってしまうことになります。

次回では、その解決策について学習します。

〇チェックポイント・キーワード

・手続き型処理
・フローチャート
・ポーリング処理
・処理の不均一
・タイムスライシング

図は かわいいフリー素材いらすとや より引用