

前回のおさらい

- ・条件分岐

これを使って、クリックゲームの難易度調整

- ・リスト（配列）

これを使って、スプライトにじゃんけんをさせるプログラムの作成

- ・コメント機能

今回の新しい内容…なし

- ・複雑な条件分岐

1. 条件分岐の組み合わせ

前回講義の最後に見せた、じゃんけんの勝敗をつけるプログラムを考えてみる。

【前提】

乱数を1～3で発生して、1＝グー、2＝チョキ、3＝パー、とする。

相手の手、という変数に上記の値を入れる。

自分の手は、スプライトをそれぞれ作成し、

スプライトをクリックした時に自分の手、という変数に1～3を入れる。

今回はスプライト（ネコ）が、勝ち／負け／あいこ、かどうかを記入する

（自分が勝ち／負け／あいこ、ではないことに注意する）

もし、自分の手が1（グー）の場合

相手の手が1（グー）	
相手の手が2（チョキ）	
相手の手が3（パー）	

もし、自分の手が2（チョキ）の場合

相手の手が1（グー）	
相手の手が2（チョキ）	
相手の手が3（パー）	

もし、自分の手が3（パー）の場合	
相手の手が1（グー）	
相手の手が2（チョキ）	
相手の手が3（パー）	

上記の図を使って、条件が複雑な場合は整理すると良い。
来週はフローチャート（流れ図）を使うが、そちらを利用するのも良い。

ところで、Scratch の条件分岐はこんな感じだった。



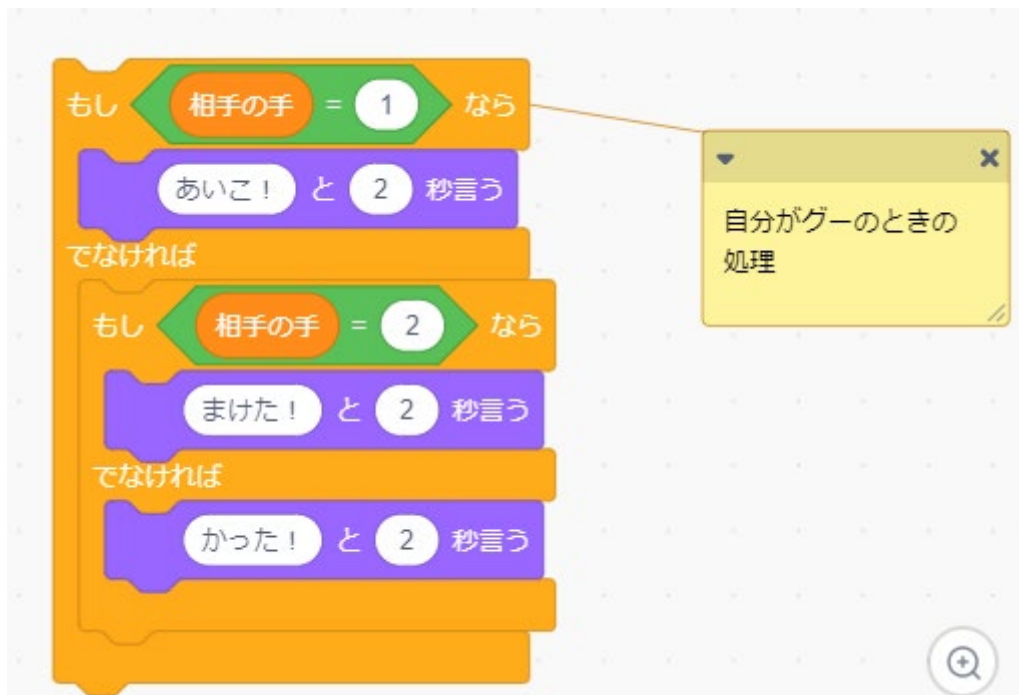
※六角形の部分は条件が入る

要は2つしか分岐できないので、これを組み合わせる必要がある。



このような感じで、まずは自分の手が1（グー）かどうかを判断する。
でなければ1以外、すなわち2（チョキ）か3（パー）なので、自分の手が2（チョキ）かどうかを判断する。2でもなければ3（パー）になる。

同じようにして、今度は相手の手が1（グー）かどうかを判断。



このようにして2段階で処理することが大事である。

同じように、自分がチョキの時、パーの時を作成してみよう。（本日の課題）

- ・さらに複雑な条件分岐

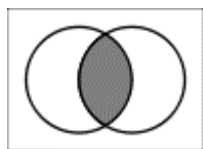
条件分岐はもっと複雑なものがある。

条件が2つ以上ある場合を考えてみる。

条件 A：眼鏡をかけている人、条件 B：男性 とする。

【AND と OR、そして NOT】

こちらは AND。論理積とも呼ばれる。



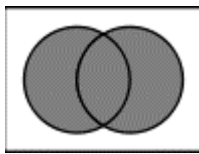
両方の条件を満たす場合（条件 A、かつ、条件 B）。

上の例だと、眼鏡をかけている男性、だけが該当する。

Scratch では右図のような「かつ」を使用し、条件を 2 か所入れることで使用する。



こちらは OR。論理和とも呼ばれる。



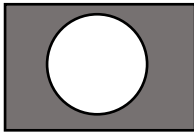
どちらか片方の条件を満たせば OK (条件 A または 条件 B)。

先ほどの例だと、眼鏡をかけている方 (男女とも) と、男性全員があてはまる。

Scratch では右図のような「または」を使用し、こちらも条件を 2 か所入れる。



最後に NOT。否定とも呼ばれる。



条件の「反対」を表す。

条件 C が、80 以上だとしたらその反対になるので 79 未満になる。Scratch では右図のよう

な「～ではない」を使用する。



この 3 つはほかのプログラミング言語でもよく使われ、論理演算と言う。

これを組み合わせることで、全ての条件を判定することが出来る (らしい)。

【少し難しいけど、知っておいた方がよい話】

コンピュータの中は 2 進数 (0 と 1) で出来ています。

イメージ的には電灯の ON (1) / OFF (0) ですが、厳密に言えば電圧の Hi/Low になります。

この 0 と 1 を使って、コンピュータは比較判断をしています。

数学で命題、というのがあったかと思いますが、真 (True)、偽 (False) を使ったようにコンピュータ内でも同じように操作しています。0 を False、0 以外を True にしている言語がほとんどです。

では、先ほどの AND で考えてみます。

条件を満たす = True(1)、条件を満たさない = False(0)

条件 A	条件 B	AND
0 (満たさない・偽)	0 (満たさない・偽)	0 (満たさない・偽)
0 (満たさない・偽)	1 (満たす・真)	0 (満たさない・偽)
1 (満たす・真)	0 (満たさない・偽)	0 (満たさない・偽)
1 (満たす・真)	1 (満たす・真)	1 (満たす・真)

掛け算をすれば AND と同じになるので、論理積と呼ばれます。

同じように OR、も考えてみましょう。

条件 A	条件 B	OR
0 (満たさない・偽)	0 (満たさない・偽)	0 (満たさない・偽)
0 (満たさない・偽)	1 (満たす・真)	1 (満たす・真)
1 (満たす・真)	0 (満たさない・偽)	1 (満たす・真)
1 (満たす・真)	1 (満たす・真)	2 (満たす・真) →1

最後の行の結果は足したら 2 になるのですが、0(False・偽)、0 以外(True・真)なので真になります。

この理論で実はコンピュータ内での加算処理なども行っています。

(長くなるので割愛します…)

・難しい話ばかりではなんなので

スプライトや背景を「自分で描いた絵を利用する」などが出来ます。



スプライトの箇所から、一番上の「スプライトをアップロード」を選択します。

そうするとファイルの場所を聞いてきます。

jpg、gif、png、svg、bmp などの画像ファイルの主なものを網羅しています。

ご自身でキャラクター作成するのも良いかと思えます (愛着がわきます)。

インターネットからファイル取得するのもありますが、個人的使用の範囲内をお願いいたします。

(画像にはそれぞれ著作権があります・フリー素材等をきちんと探して利用しましょう)

また、スプライトや背景は Scratch の画面でも描くことができます。

描いたものを右クリックすることで、保存も出来ますよ。

是非、こちらも挑戦してみてください。

