

**aibo (アイボ) さんと  
ビジュアルプログラミング  
まなんじゃお〜!**



**ご注意**

**この資料はSONYさんのアイボ (ERS-1000) を簡単にプログラミング体験できる「aibo ビジュアルプログラミング」の使い方やサンプルなどをアイボオーナー (ハピラキ) が自身のマニュアルのために勝手に作成したものです。この内容についての保証、お問い合わせ、配布、販売などはご遠慮願います。**

**作成 2022年8月 時点のものです。**

## 10時限目



### アイボとゲームしたいですね！その3

#### ● 動作を確認するって大変ですね～

アイボさんとの「あっちむいてほい！」で勝敗確認はいかがでしたでしょうか？ 少しはゲームらしく楽しめましたか？

最初は、アイボさんの肉球が押されたかの判断するタイミングが分からなかったかと思いますが、何度か繰り返しているうち、だんだんアイボさんが確認するタイミングがつかんでくるかと思えます。更に、いろいろな動作や【調べる】ブロックを使いながら、それぞれのコツ（タイミング）をぜひつかんでくださいね。

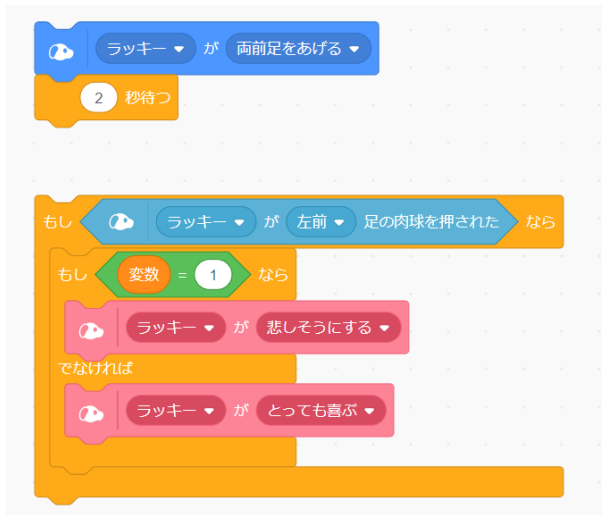
ところで、何度か繰り返しているときに、**何もしてくれなかった**ことありませんか？ これは、**前足の肉球の押すタイミングが合わなかったり、また、短かかった**とき、この状況になっているかと思えます。

前回のプログラムで【2秒待つ】のおまじないを入れた説明をしました。



ために「肉球を押さない」ようにしてみましょう。すると、何もしないで元の状態に戻っているかと思えます。なぜでしょう。

もう一度流れを見てみましょう。



⇒アイボさんが両前足を上げる  
 ⇒2秒待ちますが・・・  
 ここで肉球を押さなかった場合  
 ⇒ 右前足の肉球チェック

肉球が押されていないので  
 この処理はされませんね・・・

つづいて

⇒ 左前足の肉球チェック



肉球が押されていないので  
 ここも処理はされませんね・・・

そのまま【立つ】もとに戻る。

つまり、肉球を押すための待ち時間として【2秒】をセットしたのに、押さなかった場合、または、押したのに確認ができなかった場合に、後の処理の「前足の肉球が押されたかのチェック」が素通りされてしまったために、何もしてくれない状況となります。

どうしたらよいでしょうか？  
 考えてみましょう～

## ●肉球がおされるまで待つことができれば・・・

### 課題

- ① 両足を上げた後、すぐに次の処理に進むので2秒待ち入れた
- ② 肉球を押さなかった場合、判断できず何もしない状況となる
- ③ 改良点として、肉球が押されたかを待つ必要がある

ちょっとむずかしくなりましたね～、確認する方法はいろいろとあるかと思いますが、いくつかやり方を考えてみたいと思います。

## ■確認する方法のパタン

### 【パタン1】

肉球が押されているかを確認し続け、右または左が押されたら、今回新しく作る変数（オーナさんが左右どちらを指したかを保存する）に値を入れます。値が入ったことが確認できるまで次の処理に移らないようにします。

### 【パタン2】

肉球が押されるまで、とにかく待つ。こちらのほうが楽そうですが、学習のため、今回はパタン1を考えます。なお、パタン2は 応用編で、別な作り方と合わせて説明します

## ●新しい変数をもう一つ追加してみましょう

ここで新しいことを1つ覚えましょう。これまで【変数】は値を保存するための魔法の箱として使ってきました。でも、プログラムでは、たくさんの情報を保存するため、覚える種類数の分、変数の箱が必要となります。新しい変数を作る方法をぜひ覚えてください。

### (1) 変数のブロックから「変数を作る」を押してください



初期状態では、変数はそのまま【変数】と名前がついている、魔法の箱が1つだけです。

### (2) 変数のブロックから「変数を作る」を押してください



- ① 「新しい変数」の入力画面が出ます。
- ② 分かりやすい変数を作ります。
- ③ 今回「オーナの左右の向き」としました。自由に決めてくださいね。

### (3) 変数のブロックに新しく追加されました。

もともとあった【変数】に新たに【オーナの左右の向き】という別の変数を追加しました。ブロックの部分をクリックすると【変数】と今回作ったもの、どちらかが選択できるようになります。





いままで使ってきた【変数】と呼ばれる箱  
アイボさん側の乱数の値を入れてきました



今回【オーナの左右の向き】の変数を追加し  
て、オーナの左右方向を値（1または2）を  
保存します。

【パターン1】の流れを少し考えてみましょう。

肉球が押されているかを確認し、押されたらオーナさんの示  
した方向の値を保存。確認できるまで次の処理に移らない

次の処理に移らない様にするには、【制御】ブロックでできそうで  
すね。更にいくつかやり方がありそうです。

ここで変数に保存する値のルールを作りたいと思います。

- ① アイボさんの左右は乱数をつかって1か2で決めました。  
1の時は【左】、左の方向を向くようにしました。  
2の時は【右】、右の方向を向くようにしました。
- ② オーナさんの左右方向も同じように値(ルール)を決めます。  
オーナさんが左向きを指定：変数【オーナの左右の向き】  
の値を「1」とします。  
オーナさんが右向きを指定：変数【オーナの左右の向き】  
の値を「2」とします。つまりアイボさんと同じ2の値

## 【サンプル1】 どちらか押されるまで繰り返す

繰り返しを抜けるのは、値が1か2が設定された時



⇒変数【オーナの左右の向き】を0

⇒【オーナの左右の向き】が1か2にならない間は処理をし続ける

- ① 左前足の肉球が押されたとき【オーナの左右の向き】=1
- ② 右前足の肉球が押されたとき【オーナの左右の向き】=2

## 【サンプル2】 左右どちらか押されるまで繰り返す



サンプル2は、左前足か右前足のどちらかが押されるまで、待つようにしています。厳密には、流れのとおり肉球チェックが2回行われますので、正しい方法ではありませんが数秒間肉球を押すようにしていますので、この内容でも同じように処理できます。

今回はサンプル1を使っていきましょう。なお、同じ場所にサンプル2を置き換えても問題ありません。

つぎに、オーナさんの左右の方向を保存している変数【オーナの左右の向き】に左右どちらかが押されたかの値 1(左)または2(右)が保存されましたので、以降の処理も変更したいと思います。

以降の処理、前回作ったプログラムを見直ししてみましょう。

```
もし ラッキー が 左前 足の肉球を押された なら
  もし 変数 = 1 なら
    ラッキー が 悲しそうにする
  でなければ
    ラッキー が とっても喜ぶ
もし ラッキー が 右前 足の肉球を押された なら
  もし 変数 = 2 なら
    ラッキー が 悲しそうにする
  でなければ
    ラッキー が とっても喜ぶ
ラッキー が 立つ
```

⇒ここはチェック済となるので不要ですね

⇒今回、新たに【オーナの左右の向き】の変数を作りましたので、アイボさんの向きが保存されている【変数】と比較するだけでチェックできそうですね。

⇒ここもチェック済となるので不要ですね

⇒こちらも同様です。

作り変えてみたものがこちらとなります。

ん？ これだけ?? 検証してみましょう。

```
もし 変数 = オーナの左右の向き なら
  ラッキー が 悲しそうにする
でなければ
  ラッキー が とっても喜ぶ
```

⇒アイボさんとオーナの向きをチェック

アイボさん左(1)：オーナ左(1)アイボさん負け

アイボさん左(1)：オーナ右(2)アイボさん勝ち

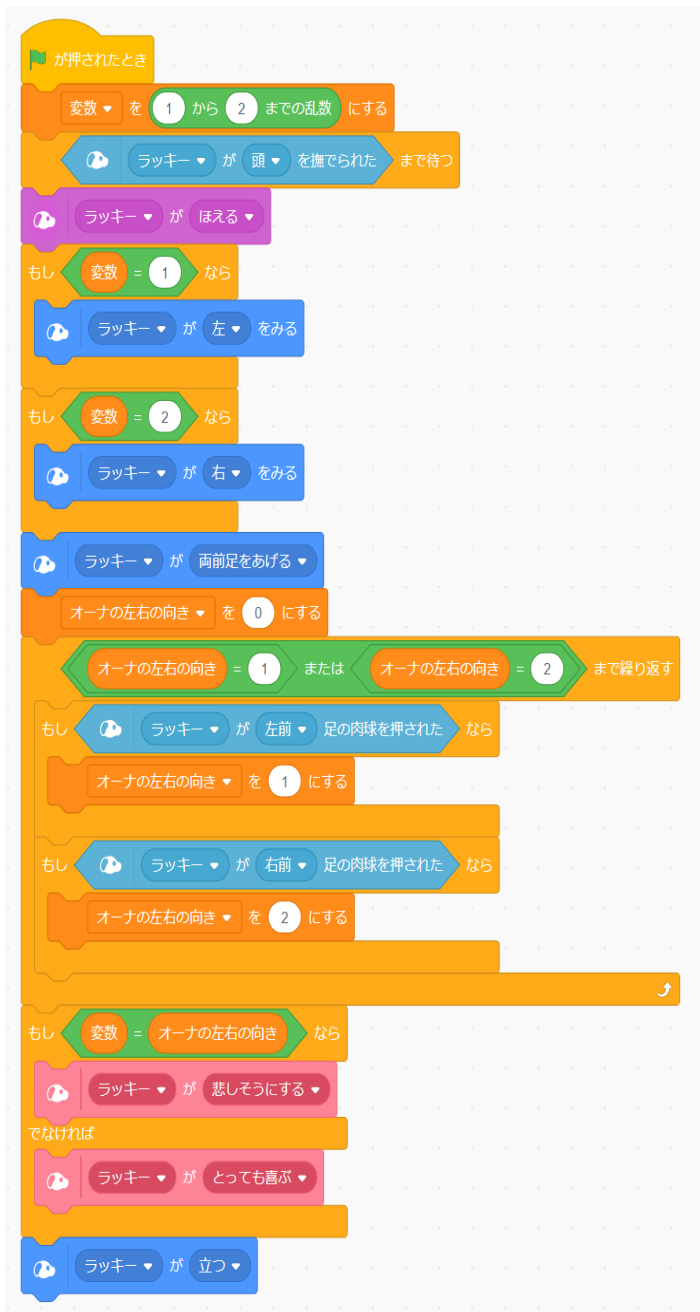
アイボさん右(2)：オーナ左(1)アイボさん勝ち

アイボさん右(2)：オーナ右(2)アイボさん負け

これだけで、OKのようですね・・・



全体を組み立ててみたいと思います。



### プログラムの開始

- ⇒【変数】を乱数1～2に設定する
- ⇒頭を撫でられるまで待つ
- ⇒撫でられたので ほえて知らせる
- ⇒【変数】が1ならば
- ⇒左を向く
- ⇒【変数】が2ならば
- ⇒右を向く

⇒オーナさんに教えてもらいます

- ⇒新しい変数【オーナの左右の向き】を0にして、1か2にならない間は繰り返す
- ⇒左足の肉球が押されたら【オーナの左右の向き】の値を1（左）に設定
- ⇒右足の肉球が押されたら【オーナの左右の向き】の値を2（右）に設定

### アイボさんが向いている方向

- 乱数できめた値 1：左、2：右と
- オーナが決めた左右の値
- 左肉球を押した場合、左：1
- 右肉球を押した場合、右：2
- を比較して 喜ぶか悲しそうにする。

かなり難しくなりましたが、いかがでしょうか？

あせらず、ひとつひとつ 確認しながらやってみましょう

**10時限目はこれで終了です。**

おさらいしましょう。

- ・ 9時限目で作成したプログラムでは肉球が押されなかった時の対処が必要
- ・ 肉球が押されるまで、待つように繰り返し処理を追加
- ・ 変数を新たに追加して、アイボさんとオーナさんの向きも新たに保存して、アイボさんとオーナさんの値を比較するように変更
- ・ アイボさんの左右の向き、オーナさんの左右の向きを比較して、同じ向き（値）の場合は「悲しそう」に、また違う向き（値）の場合は「喜んで」もらう

いかがでしたでしょうか？

10回を通して、ビジュアルプログラミング初心から、簡単なゲームを作るところまで学んできました。まだまだたくさんのブロックがありますので、ぜひ、いろいろと試してみましょう。

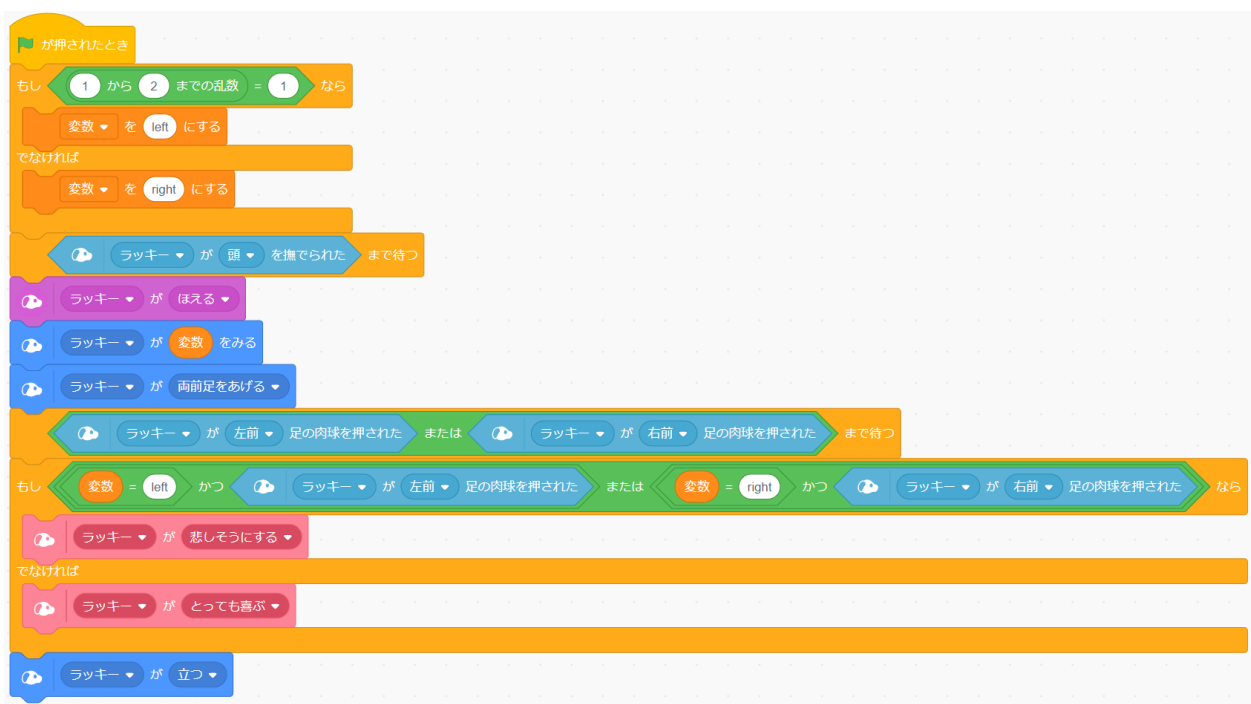
**この「faibo（アイボ）さんとビジュアルプログラミング まなんじゃお〜！」は、パート1として一旦終わりにしましょう。**

**新たに「faibo（アイボ）さんとビジュアルプログラミング まなんじゃお〜！ パート2」として、更に動きのあるブロック、イベント、メッセージ、リスト、更にブロック定義などを使っていきたいと思えます。**

## 応用編

応用編として、別のサンプルを掲載します。

ゲームの内容は変わりませんが、プログラムの作り方が大きく異なっていることがわかるかと思います。



### ■ちょっと解説



乱数を1, 2のうち1であれば  
変数=「left」という文字をいれる  
1でなければ、つまり2だと  
変数=「right」という文字をいれる



変数の値に基づき右、左を向く



- ① 左右の前足肉球が押されるまで待ちます。
- ② 【もし〜なら】は以下の条件が成立したかのチェックします  
[変数=left(乱数1)] かつ 左前足が押された  
または  
[変数=right(乱数2)] かつ 右前足が押された
- ③ 条件が成立した場合  
つまり、アイボさんとオーナさんの指した向きが一緒だったら  
アイボさんは同じ向きを向いたため「悲しそうに」をする
- ④ 条件が成立しなかった場合  
つまり、アイボさんとオーナさんの指した向きが違ったら  
アイボさんと違う向きだったので「とっても喜ぶ」をする

今回学んだ内容と同じ処理になったかと思います。  
かなり、難しくなりましたが、こんな作り方もできるということを知っていただくだけで問題ありません。

繰り返し、繰り返し、いろいろと作っていくことで慣れていきますので、是非、沢山のアイデアでプログラミングしてくださいね。