aibo さんと ビジュアルプログラミング まなんじゃぉ～ パート2

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（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。

LESSON 2

イベントをマスターしちゃいましょう

● イベントとは？

ブロックに「イベント」と「aibo のイベント」というものがありますね。LESSON１では、新しい言葉を覚えてもらい、その言葉を認識すると処理がスタートするように、こんなブロック定義を作り、実際に追加した言葉の認識を確認してみました。

これらは、イベントと呼ばれております。ちょっとだけむずかしい説明となりますが、プログラムの世界ではよく見かける言葉です。パート１の２時限目でコンピュータのプログラムについて簡単に説明をしました。コンピュータはいろいろな処理が順番に行われ、プログラム通りに処理されることで、ゲームや WEB サイトの動画を見たり音楽を再生できていますが、使うに時には、必ず人の操作が行われています。例えば、画面のタップ、キー入力、マウスクリック、時間が来たらスタートなど、普段何気なく使っていますよね。その操作が行われたら、反応して表示したり、動作したりしていますよね。

コンピュータの世界では、これらの操作を普段は待ち受けをし、操作をされたら、開始をしたり、続きを行ったりしながらたくさんのプログラムで処理をこなしています。これらを一般的イベント処理と呼ばれております。

もう一度、イベントブロックを見てみましょう。

イベントブロック

Aibo イベントブロック

「〇〇したら・・〇〇を・・・」となっていますよね。つまり、何かが起こったら何かをすること・・の開始の条件を待っているブロックとなります。

LESSON2では、このイベントを少し頑張って覚えてみたいと思います。でも、皆さんはパート１で旗マークのブロックやLESSON１の認識ワードですでに分かっているよ・・と思いますが、ちょっとだけ、こんなことをやってみたいと思います。

イベントブロックの「【スペース】キーが押された時」を見てみましょう。

キーボードの数字や文字、カーソルキーの方向キー、スペースのリストが表示されていますね。

このブロックを使うことで、「〇〇キーが押されたら・・・」の処理開始の定義ができそうです。次のブロックを作ってみてみましょう。ここで作った内容はあとで使いますので、そのまま作ってみましょう。

① 新しい変数【オーナさんが示した向き】を作ります。

② 先ほどのイベントブロックと作った変数を次のように並べてみましょう。イベントのボタンは、キーボードのカーソル矢印キーとなる　左・右を使います。

また、①で作った変数への設定する値は、今回「右」「左」を入れてみましょう。パート１の５時限目の変数の説明をしましたが、変数には数字や文字も設定ができます。今回更に漢字を設定してみましょう。

③　準備ができたら、キーボードの左右の矢印キー【←】【→】を押して、「オーナさんの示した向き」をクリックして、値が何になっているか確認してみましょう。

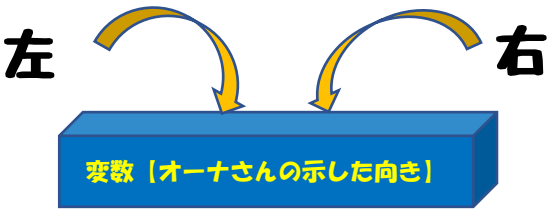
【←】左矢印キーを押したら・・・

【→】右矢印キーを押したら・・・

③　左右キーに反応して変数【オーナさんの示した向き】の値が変わったことが確認できますね。つまり・・・

キーボードの左右キーが押されたら

イベントブロックで処理開始



変数【オーナさんの示した向き】に【左】や【右】が設定されます。

④ すでにお気づきかと思いますが、左右キーはいつでも押せばイベントとして呼び出されて、処理されています。

イベント処理は実はとても重要でプログラミングでは必ず必要なものとなります。一方、いつでも処理ができてしまうことで、処理したくない場合、ガード処理なども考慮しなければならないのがイベント処理のむずかしいところで、プログラムの予期しない結果になる場合があります。

さて、イベントブロックの動作がなんとなく、わかってきたところで、ＬＥＳＳＯＮ１で作った「あっち向いてホイ」に追加して、オーナさんと勝負ができるようにしてみましょう。

ＬＥＳＳＯＮ１では、認識ワードでアイボさんに次の言葉が認識されると左右に向いてくれるプログラムを作りました。
「あっちむいてほい」、「あっちむいて」、「ほいほいほい」
ＬＥＳＳＯＮ２では、先ほど確認したイベントを組み合わせをして、オーナさんと勝負してみる処理を追加してみましょう。

これまではブロック並べながら処理の流れを理解してきましたが、複雑なプログラムやイベントが沢山あると、頭の中で整理ができなくなったり、また思ってもいない結果が発生することがあります。
今回はフローチャートと呼ばれるものをちょっとだけ作りながら、考えていきましょう。

【遊び方ルール】

①アイボさんに認識ワードを使って話しをする
②認識ワードを認識すると吠えて、左右どちらかを向く
③左右キーを使ってオーナさんの指の向きを教える
④同じ向きだったら、アイボさんの負け、違う向きだとアイボさんの勝ち
　勝ち（歌う）、負け（悲しむ）でアイボさんが勝敗を振る舞いで表現する

【事前準備】

① 認識ワード設定
「あっちむいてほい」、「あっちむいて」
「ほいほいほい」を認識ワード１で登録

② 変数を事前に作成
【アイボさんの向き】
【オーナさんの示した向き】

③ アイボさんを指示待ちにする

【イベント処理（左右キー処理）】

左右キー【左】が押されたら

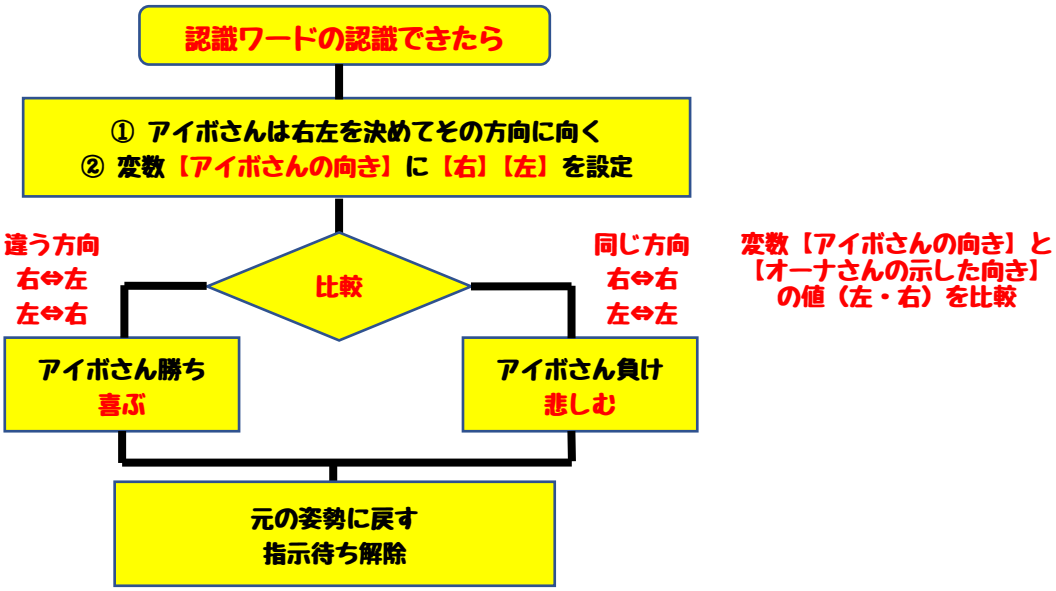
変数【オーナさんの示した向き】に
【左】を設定

【イベント処理（左右キー処理）】

左右キー【右】が押されたら

変数【オーナさんの示した向き】に
【右】を設定

【認識ワード処理】「あっちむいてほい」「あっちむいて」「ほいほいほい」を認識したとき



先ほどフローチャートに基づき、プログラムを作ってみましょう。
作り方はいろいろとあります。一例として参考にしてください。

あらかじめに WEB で設定した認識ワードを変数に登録。アイボさんを指示待ちにする

左右キーが押されたら、新しく作った変数【オーナさんの示した向き】に【左】【右】を設定

①認識ワードの言葉を認識できたらスタート
②認識ワードが認識できたので一回ほえる
③乱数１～２でアイボさんの方向を決める
　乱数１の時は、【右】を向く
　また変数【アイボさんの向き】に右を設定
　乱数２の時は、【左】を向く
　また変数【アイボさんの向き】に左を設定

④変数【アイボさんの向き】の値と
【オーナさんの示した向き】の値を比較
⑤同じ向きであれば、アイボさんの負け
　アイボさんが悲しむ ふるまいをする
⑥違う向きであれば、アイボさんの勝ち
　アイボさんが楽しむ ふるまいをする
⑦指示待ちの解除を行う

実際に実行してみましょう。どうでしょうか？

ん？　なんかおかしいことに気が付いたかと思います。
　　アイボさんが【右】に向いた
　　オーナさんが、アイボさんに向かって【右】のキーを押した
　　向かいあっているので違う方向なのに負けの扱いに・・・

そうですよね・・向かいあっているので、右と右、左と左だと違う方向を向いていることになります。本当であればアイボさんが勝ったこととなるのにプログラムでは負けにしてしまっています。

アイボさんは自分自身の向いている方向は右ではなく【左】

アイボさんは自分自身の向いている方向は左ではなく【右】

オーナさんはアイボさんと向き合っているので【左】

オーナさんはアイボさんと向き合っているので【右】

プログラムは、【左】【左】、【右】【右】を負けとしましたが、この通り、このパタンだとアイボさんは逆向きなので、アイボさんの勝ちとなります。

　　あっ、フローチャートの時に気が付いていましたか？

修正の仕方は、いろいろとありますね・・

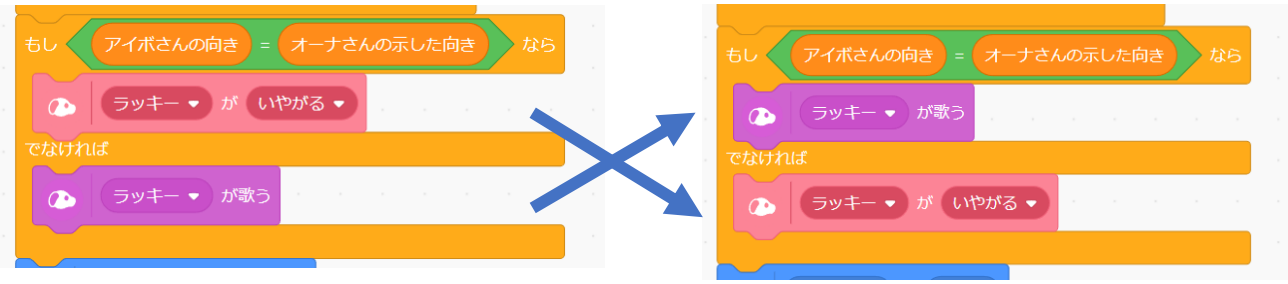
■修正ケース１
イベント処理でオーナさんの左右キーを変数【オーナさんの示した向き】の右、左の設定を逆にする

■修正ケース２
アイボさん向きを決定するとき、変数【アイボさんの向き】の右、左の設定を逆にする

■修正ケース３
アイボさんとオーナさんの向き判定処理でイコール【＝】の時の条件の制御処理の内容を変更する

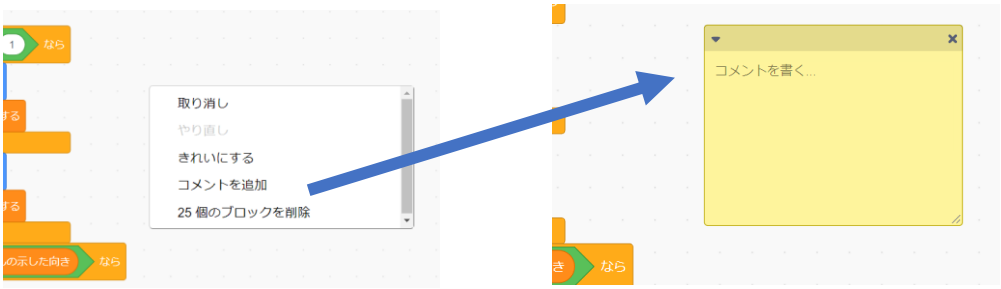
どれも可能です。ただ、なぜ？そのような条件や設定にしたかを後で見返ししたときに、頭の中で悩まなければならなくなりますので、できる限り後で見たときに理解しやすい方法に修正した方法がよいですね。修正方法は皆さんでいろいろと試してみてください。

今回は以下のように、条件判定後の振る舞いを逆にしてみました。



複雑なプログラムになると、なぜそうしたっけ？ がだんだん多くなり、そのたび、頭の中でシミュレーションしていくことが多々あります。そうなんです、必ずあります！
そこで、なぜそうしたのかを忘れないようにコメントを残しておく方法を覚えましょう。

ブロックを配置している場所で右クリックを押すと、つぎのようなメニューが表示されます。【コメントを追加】を選択するとコメントを記録する付箋紙のようなものが表示されますので、後で見てわかりやすい説明を残しておくとよいですね。



また、特定の部分についてコメントを残しておきたい時は、該当するブロック上で右クリックすると、次のようにそのブロック部分にコメント追加ができます。また、ブロック指定の場合、自動的に引き出し線が追加表示されます。どの部分の説明かをあとで見たときに把握しやすくなります。是非、活用していきましょう。

では、改めて修正およびコメントを追加したプログラム全体を見てみましょう。

ＬＥＳＳＯＮ２のおさらいをしておきましょう。

① イベントの動作について確認をしました。
② また、イベントはいつでも処理が開始されとても便利です
③ イベントはコンピュータの世界は不可欠な処理です。
④ イベントをいくつかつかって、ＬＥＳＳＯＮ１のあっち向いてホイのプログラムを拡張して、オーナさんの向きと判定の処理を追加してみました。
⑤ ブロックで作る前にフローチャートを使って全体のイメージを作って確認することをしました。
⑥ あたまで考えた処理で大丈夫だと思っていたら、想定外の結果となりました。左右判定において、向きあっているため同じ方向を示した場合、逆向きとなる考えが不足していました。
⑦ ⑥の修正する方法はいくつかありますが、後で見てわかりやすい箇所を修正しました。
⑧ ブロックだけだとわかりにくいので、便利なコメントを活用していきましょう。

ＬＥＳＳＯＮ２ いかがでしたでしょうか？

パート１から見ていただいた方は、あっち向いてホイがものすごくゲームらしいものになったか・・・違いがわかりますね。
パート１はちょっとだけ遠回りをしながら、ブロックの使い方や制御、振る舞いを学んできましたので、すでに基本的なブロックの使い方などは理解できるかと思います。今回更にイベント処理を覚えましたので、更にいろいろなプログラム世界が広がったかと思います。イベント関連もいろいろと試してみてくださいね。