

第 1 回講義概要

1 Scratch の基本操作

(1) 簡単なプログラムの作成

命令ブロックの説明のあと、図 1 のプログラムを作成させ実行。更に図 2 のプログラムを作成させ実行。なぜ、図 1

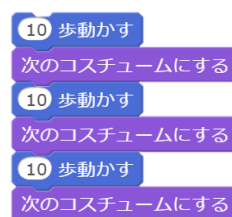


図 1

より図 2 の方が良いのかを考えさせる。最後に図 3 のプログラムを作らせる。この指導の中でプログラムの実行・停止などの操作方法について、コスチュームについて説明する。

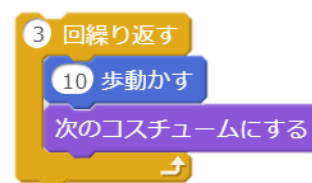


図 2

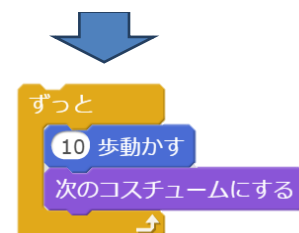


図 3

(2) 複製・削除・拡大・縮小の使い方について説明

スプライトを使って、それぞれの機能について説明する。

(3) 新しいスプライトの追加

多くのスプライトがあることを確認させる。またステージへの取り入れ方について説明する。

2 座標軸について説明

(1) 座標軸の説明は、特に低学年の児童には難しい。しかし、これを指導することで、Scratch の学習に広がりが出てくる。図のように (x, y) で説明するより、端的に上下は 180~-180 で左右は -240~240 であると話す方がよい。



児童にスプライトの動きから確認させ、上下左右の x と y の値を調べさせる。この説明には学年に応じて対応する必要がある。

(2) 例題として、ボールが下にあった場合、そのボールを上げるにはどのようなプログラムが考えられるか。

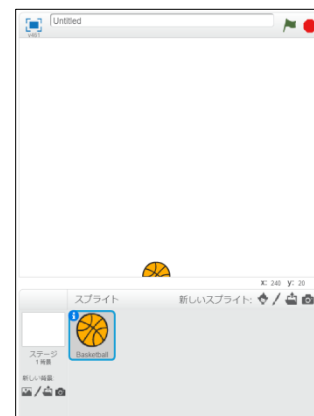
手続きとして、

① ボールを下に置く。

② ずっと、y 座標を 10 ずつ変える。

※ここでは PC を使わずに、白板に書きしっかり考えさせる。

プログラム作成は次回実施する。



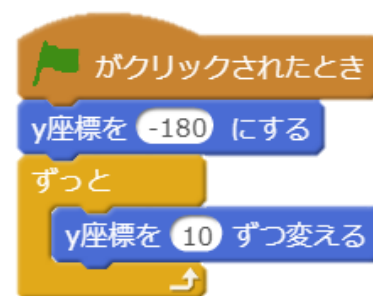
第2回講義概要

1 ボールを使い基本的な動きを理解

(1) 前回の基本操作の復習

(2) 前回2の(3)より、下にあるボールを上げるにはどのようなプログラムになるか。右図を作成。

ここでは y 座標の変化について、-180、-170…-10、0、10、170、180 と具体的に白板に書き、その上でその変化をプログラムで表現させる。児童の中には「y 座標を-180 にする」を「ずっと」の中に入れる場合もある。なぜうまく行かないのか考えさせる。

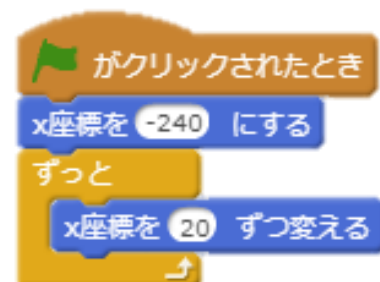


(3) 逆に、上にあるボールを下に落とすには、どのようなプログラムになるか考えさせ実行させる。数字を変えるだけでできることを理解させる。

※低学年の児童には、マイナスの概念が難しいようである。ここでも再度座標軸について略図を書き説明する。

(4) 更に、左から右へ、右から左へボールを転がすにはどのようなプログラムになるか考えさせ実行させる。

※ここまできると、y 座標を x 座標に変えればできるとに気づく。



※ボールの大きさを変えたり、「見た目」より色に効果を入れたりすると面白い。4つのボールの動きを楽しむことができる。

※座標軸について説明すると、実際に動きを確認しながらプログラミングができ、次へのステップにつながる。

※ボールの速さを変えることもできる。

(5) 次回は制御について説明する。

第3回講義概要

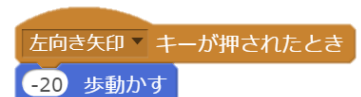
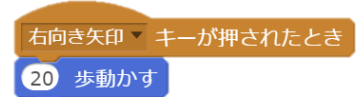
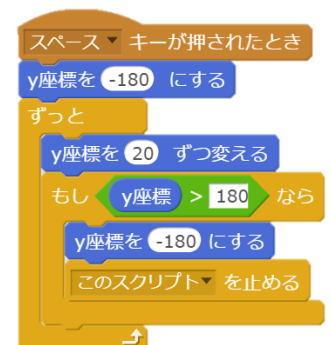
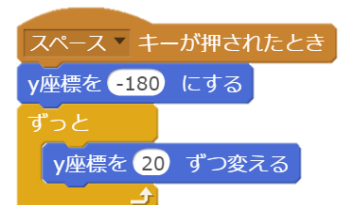
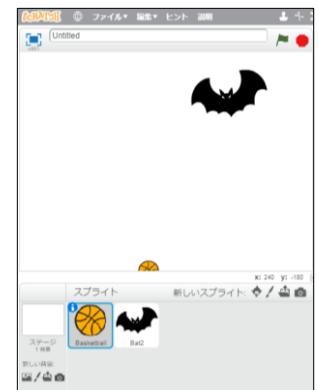
1 ボールを制御

- (1) 前回のボールの動きについて復習
- (2) 今回、右図にあるように下からボールを打ち、コウモリに当てるシューティングゲームを作成する。
- (3) 右図のように「グリーンフラッグがクリックされたとき」を使用せず、「スペースキーが押されたとき」にする。これを使用することでゲームらしくできる。しかし、このプログラムを実行してもボールは元に戻ってこない。なぜ戻ってこないのかを理解させ、次にどうすればよいのかを考えさせる。
- (4) そこで右図のように「もし y 座標が 180 より大きいなら」の条件文を加える。その中に「y 座標を-180 にする」、「このスクリプトを止める」を入れる。このようにすることでボールは一旦下に戻って止まり、スペースキーが押されるたびに打ち上げられるようになる。

※「y 座標=180」より「y 座標>180」の方がよい。ボールの速さを変えても対応できるからである。不等号についても説明する。

- (5) 更に、右図のようにカーソル移動キーにも役割を持たせることで、ボールを左右に移動させることができる。

※条件文の理解は低学年には難しいようである。何回も座標軸を書き説明する。



2 コウモリを制御

- (1) コウモリを「新しいスプライト」から追加する。
- (2) 「0.3 秒待つ」を入れることで、コウモリの動きを一瞬止めることができる。「ランダムな場所へ行く」だけではなく、「次のコスチュームにする」、「y 座標を 0 から 170 までの乱数にする」を入れることで、コウモリの動きと出現が制限できる。

※乱数の概念が難しいが、丁寧に説明する必要がある。



第4回講義概要

1 ボールがコウモリに当たった時の処理

(1) 前回の復習

条件文の利用、乱数についてもう一度丁寧にプログラムを入力しながら説明する。

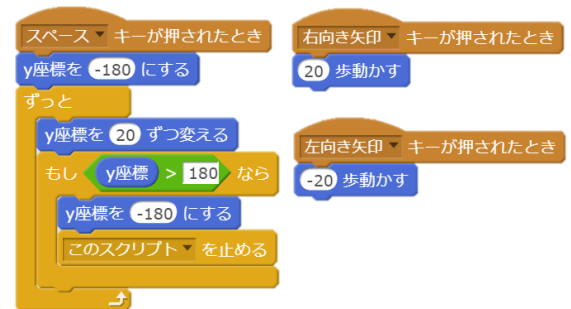
(2) 最初の図はボールに対するプログラムである。

前回と同じプログラムである。

右図はコウモリに対するプログラムである。ここではコウモリに対して最初に「前に出す」、「スコアを0にする」を宣言する。「スコアを0にする」は合計などを求める時の定石である。もし「ボールに触れた」なら、「メッセージ1を送る」、「スコアを1ずつ変える」を入れる。メッセージ1は図の「メッセージ1を受け取ったとき」に引き継がれ「当たった!!」と言う。

※もし「ボールに触れた」ならの条件文は、4種類ほどの書き方が考えられる。次ページにまとめている。

※「スコアを0にする」はコウモリのプログラムに入れる。



2 当たった時その回数をカウントする

(1) 最後の図のように、「データ」に変数の追加をする。名前は自由であるが分かりやすいものがよい。

(2) 「変数を作る」をクリックすると、「すべてのスプライト用」、「このスプライトのみ」の選択が表示される。コウモリの複製を作り、2・3匹飛ばす場合には、他のコウモリにもこの変数の適用がある方が有効であるので、「すべてのスプライト用」を選択した方がよい。

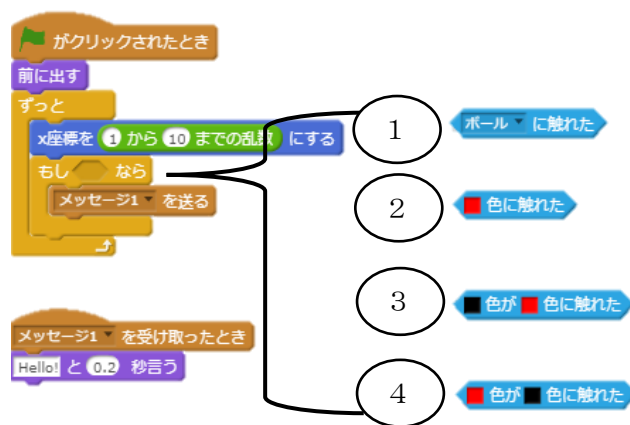
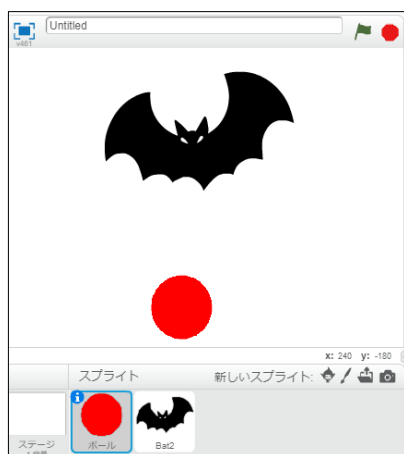


3 ここまででシューティングゲームは終了である。更に背景を変えると面白い。

補足

- 1 図のような大きなコウモリと大きなボールを作り、下記の条件文「もし< >なら」の部分に何を入れるとメッセージ 1 が出現するかについて調べた。

	コウモリを「前に出す」と宣言	ボールを「前に出す」と宣言
1	表示された	表示された
2	表示された	表示された
3	表示された	表示されない
4	表示されない	表示されない



※ なお、ボール、コウモリとも小さくした場合、当たっているのにメッセージが表示されない場合が多々ある。このようなものだと理解し進めるしかない。「前に出す」と宣言しているが、こうした方が良く思う。なお、右図のように「メッセージを送る」を用いなくても直接「〇〇と〇〇秒言う」を利用することもできる。



2 コウモリの制御

「ランダムな場所へ行く」、「y 座標を 0 から 170 までの乱数にする」でコウモリの出現を制御した。このプログラムも逆にすると上手く働かない。なぜそうなのかを考えさせる。

また、「ランダムな場所へ行く」を利用したが、y 座標と同じように乱数を x 座標に取り入れることもできる。

演習 1

第 1 段階

図のようにリンゴが落ちてくる。そのリンゴをカゴで受け取るプログラムを作成しなさい。なお、リンゴの落下速度、カゴの動きなどは適切に設定するものとする。

第 2 段階

リンゴを取った回数を表示するようにしなさい。

「キャッチ!」と音が鳴るようにしなさい。

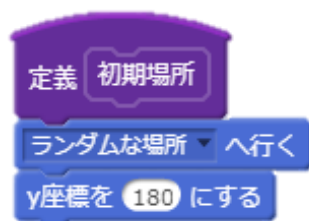
第 3 段階

20 秒以内でゲームを終了するように制限時間を設定しなさい。

第 4 段階

最後に、自分の工夫を追加しなさい。

※「スクリプト→その他→ブロックを作る」
で、リンゴの位置決めに 1 つのブロックとして定義することができる。



演習 2

第 1 段階

図のようにスペースキーを押すたびに、右側から転がってくるボールに対して、カエルが跳びはねるようにプログラムを作成しなさい。

なお、カエルが跳びはねる高さ・速さ、ボールの速さなどは適切に設定するものとする。

第 2 段階

カエルがボールに当たったなら、カエルが「ゲッ!」と表示するようにしなさい。

また、音が鳴るようにしなさい。

第 3 段階

「ゲッ!」の回数を表示するようにしなさい。

第 4 段階

20 秒以内でゲームが終了するように制限時間を設定しなさい。

第 5 段階

カエルが跳びはねた頂点で、カエルの色が赤色になるようにしなさい。

(下に降りたときは元の色に戻る)

第 6 段階

最後に、自分の工夫を追加しなさい。



演習 3

第 1 段階

図のように、ヒトデが左側（x 座標-230~-20 の範囲）に、大きさ（10%から 50%の大きさ）でランダムに現れる。（クローンを利用する。次ページ参照）

右側には矢を用意して、出てくるヒトデに矢を放つプログラムを作成しなさい。なお、ヒトデが上昇する速度、矢の速度などは適切に設定するものとする。

Point :10%のヒトデは上に上がると削除されず残る場合がある。y 座標>180 ではなく、y 座標>175 程度にすればすべて削除される。

第 2 段階

当たればヒトデの色が変化するようにしなさい。

第 3 段階

当たった回数を表示するようにしなさい。

第 4 段階

20 秒以内でゲームを終了するように制限時間を設定しなさい。

第 5 段階

最後に、自分の工夫を追加しなさい。



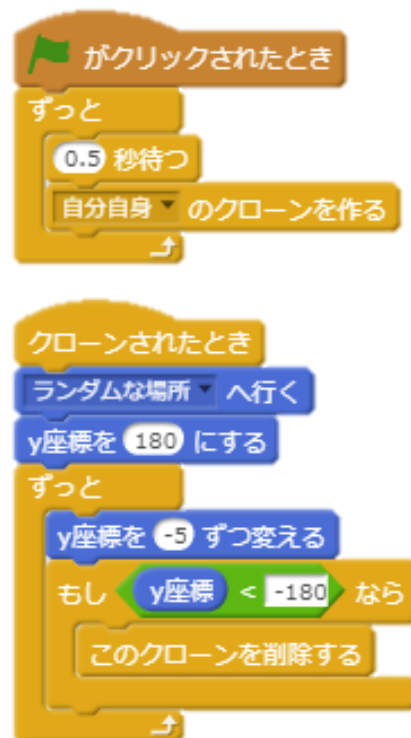
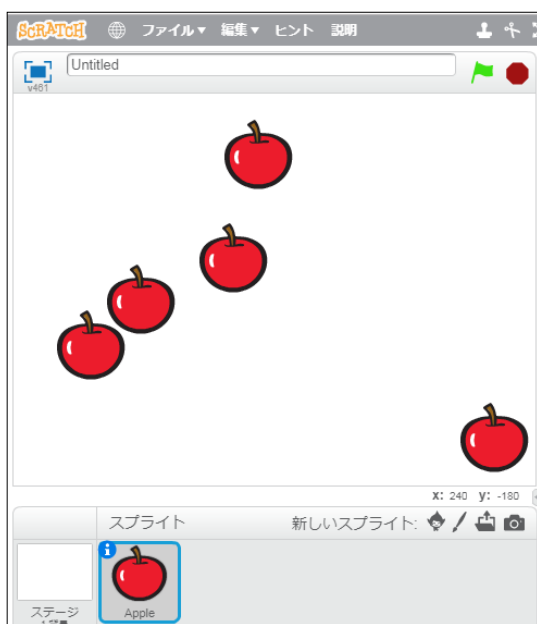
クローンの利用

リンゴが落ちてくるプログラム

- ①グリーンフラッグの「0.5秒待つ」でリンゴの数の調整ができる。
- ②上記クローン作製のプログラムを受け、「クローンされたとき」の処理を次に作る。
- ③最後には「このクローンを削除する」を入れる。

※クローンされる元のリンゴは消えているか？ 残っていないか？確認すること。

残っている場合の対処方法は？ 「隠す」、「表示」を利用する。



演習 4

第 1 段階

図のように、ブロック（自作する）が上からランダムに落ちてくる。（クローンを利用する）
左側にあるロケットで、出てくるブロックに当てるプログラムを作成しなさい。なお、ブロックの下降速度、ロケットの速度などは適切に設定するものとする。

第 2 段階

当たれば何かに変化するようにしなさい。（変化については自分で工夫する）

第 3 段階

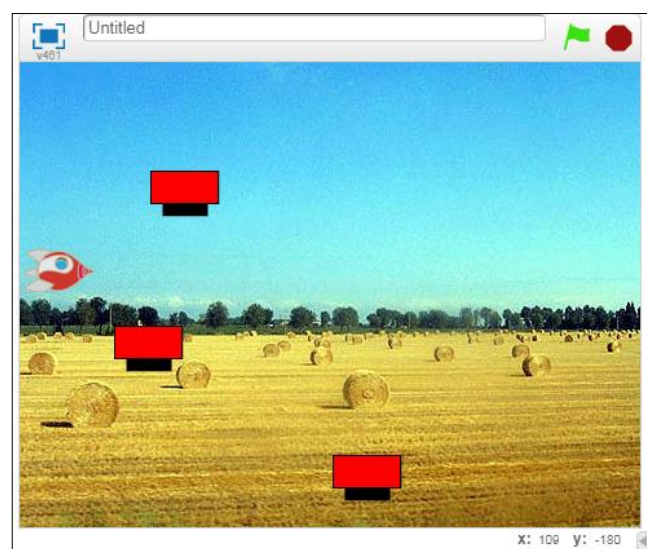
当たった回数を表示するようにしなさい。

第 4 段階

20 秒以内でゲームが終了するように制限時間を設定し、終わりになれば背景として「おわり」と表示するようにしなさい。

第 5 段階

最後に、自分の工夫を追加しなさい。



演習 5

1 クローンの利用

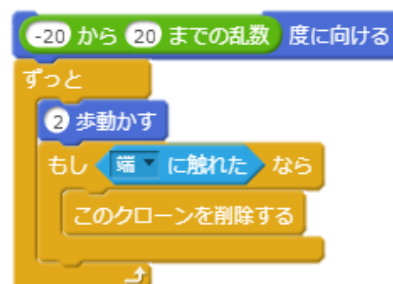
クローン技法を利用して、「海底から噴き出す泡」のプログラムを作成する。

- ①泡は「新しいスプライトを描く」から作成する。
- ②泡が発生するとき「-20 度～20 度に向ける」を入れる。※泡の動きに変化を持たせる。
- ③泡が端に触れたらクローンを削除する。
- ④泡同士触れたらその泡を削除する。



※遊び心：魚を入れて、その動きにも変化を加えるようにする。

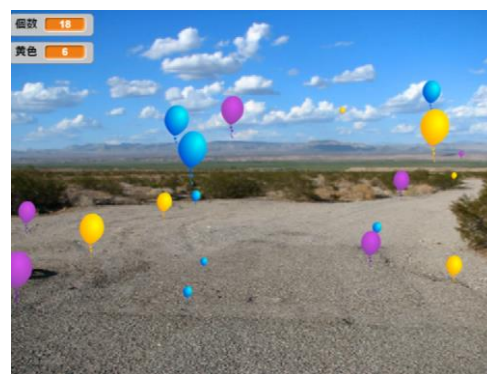
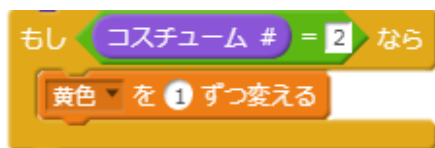
Point：図のように、「○度に向ける」の利用では、「ずっと」の中で、「y 座標を○ずる変える」ではなく、「○歩動かす」を使うとうまく行く。



2 クローン技法を用いて、砂漠で風船を発生させよう。

- ①風船の数を数え、200 個を超えたとき止める。
- ②黄色の風船の数を数える。
- ③風船の発生する場所は「y 座標を-100」とする。
- ④風船の大きさは「1 %～40%までの大きさ」とする。

Point：黄色の風船を数える時に利用する。



第12回講義概要

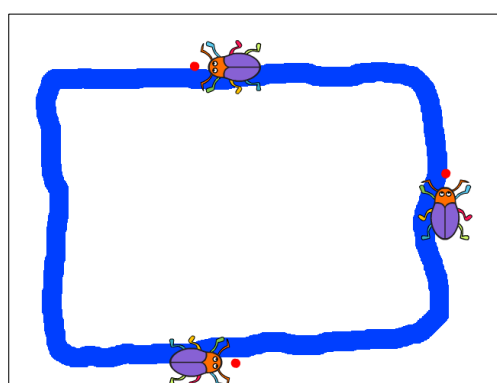
1 虫が決められた道を歩く。

第1段階

虫の前にセンサーの役割をする小さな点を付ける。ここでは条件文でそうでなければの分岐も加える。

第2段階

色々なパターンの道を描き、センサーの働きを観察する。

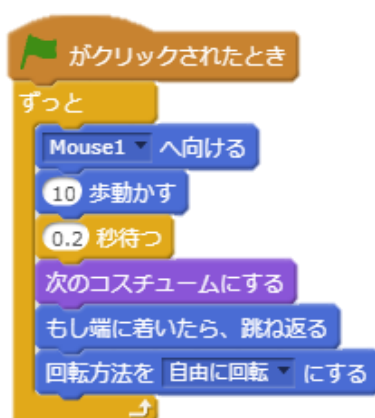


※このプログラムの応用として、mbot で実演する「8の字」運動に利用されている。

2 ネズミを追っかけるネコ

(1) 〇〇へ向けるの利用である。組み合わせることで面白いプログラムができる。

(2) 下のプログラムはネコのものである。ネズミのプログラムは自分で考える。



第 13 回 講義 概要

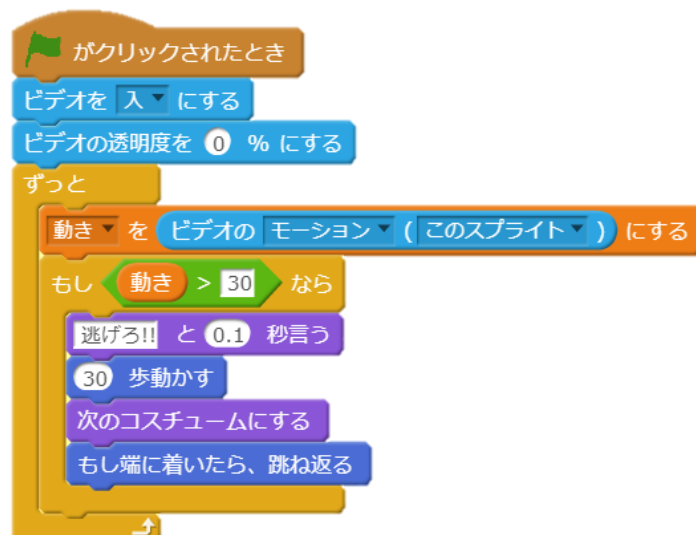
1 Web カメラの利用

手を近づけると音を鳴らす。



2 Web カメラの利用

手を近づけるとネコがにげる。



「動き」変数を作る。

ビデオの モーション (このスプライト)
ビデオ画像上のモーションや向きを検知します。

使用方法:

- ウェブカメラが必要です。

ビデオの モーション (このスプライト)

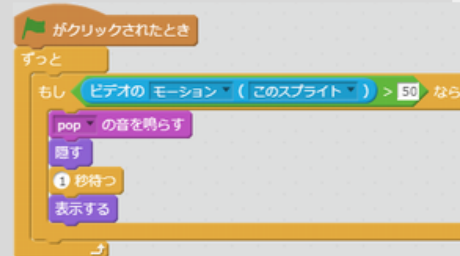
を使用して現在のスプライトの下ビデオ中のモーション（動き）の量を確認します。

- ビデオの モーション (ステージ)** を使用してビデオ画像全体のモーション（動き）の量を確認します。

- ビデオの 向き (このスプライト)** を使用して現在のスプライトの下ビデオ中の向きの量を確認します。

- ビデオの 向き (ステージ)** を使用してビデオ画像の向きの量を確認します。

スクリプトの例:



Scratch ヒント集より引用

第 14 回講義概要

Scratch 最終課題

これまで Scratch を利用して多くのプログラムを作ってきた。その最終課題として自分で好きなプログラムを作成しなさい。

- ・ゲーム的もの
- ・風景的なもの
- など

プログラミングの手順

- 1 紙にアイデアを列挙する。
- 2 どんな種類にするか。
- 3 スプライト、背景、キー操作などをどう組み入れるか。
- 4 アイデアを整理する。処理の流れを考える。変数をどう使うのか。
- 5 プログラムを作る。
- 6 テストする（動作を確認する） → 直す → テストする → 直す → . . .
- 7 他の人から意見を聞く。
- 8 完成

これまでの学習・演習で、自分で簡単なゲームを作ることができるようになった。

また、**自分で学習しなかった命令を加えていくこともできるようになった。**

以後の学習では、micro:bit、mBot を使いロボットを自由に動かすことに挑戦する。

用語

- ①アンプラグドコンピューティング（コンピュータを使わない）
- ②ビジュアルコンピューティング（Scratch のようなもの）
- ③フィジカルコンピューティング（物理的なもの、micro:bit、mBot を使う）

第 15 回講義概要

他のプログラミングツールの利用

- ・ **アルゴリズムを使ってみる。**

アルゴリズム体験ゲーム・アルゴリズムは、プログラミングの基本となる論理的思考をゲーム感覚で習得するための課題解決型ゲームソフトである。

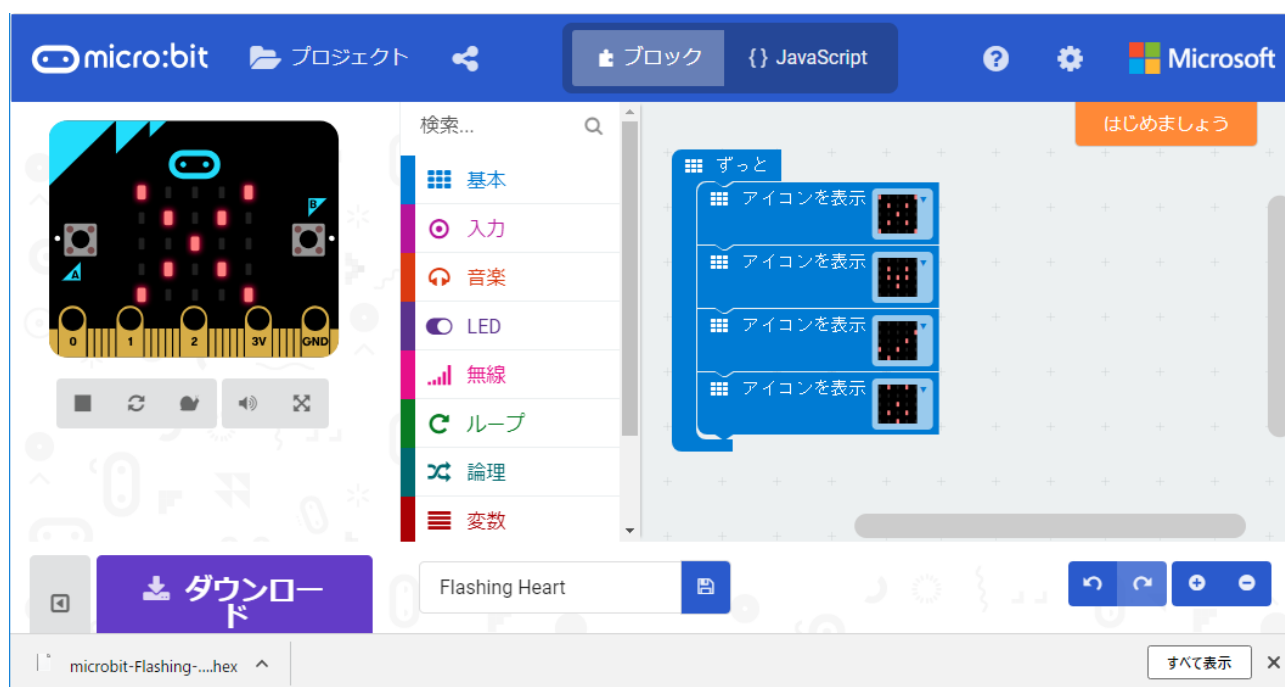
- ・ **Google Blockly を使ってみる。**

Google が提供するビジュアルプログラミング言語で、パズルのようにブロックを組み合わせることで、プログラムを作ることができる。

第 16 回 講義概要

- ・ MakeCode for micro:bit¹を使い、micro:bit で表現する。

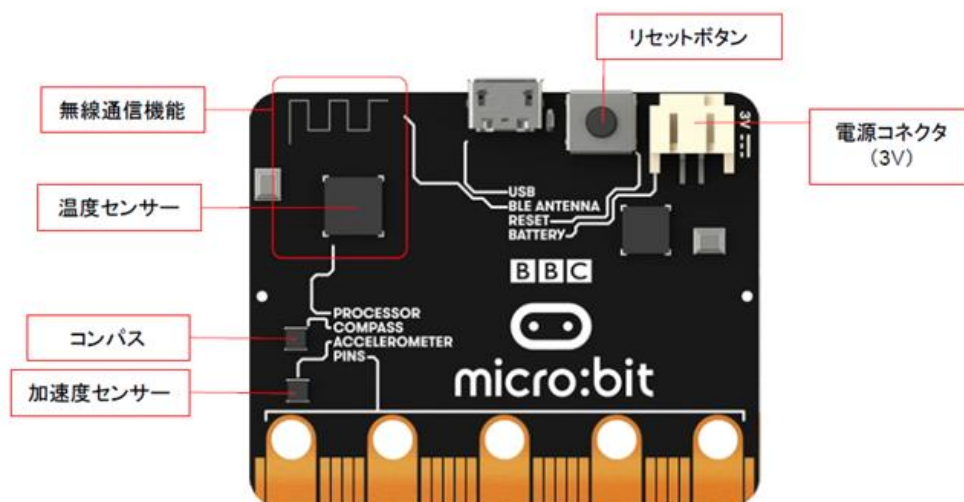
これまでの Scratch での学習をベースに、自分で表示してみたい図・文字などを作成して micro:bit で動きを確認する。



各部の働き

- 【LED】プログラミングできる 25 個の LED があり赤色に光る。光センサーとしても機能する。
- 【ボタンスイッチ】ボタンスイッチが二つある。左がボタン A、右がボタン B です。
- 【端子】金色の部分は端子と言う。センサーなどを追加して使う。5 つの穴には 0、1、2、3V、GND と名前がついている。0、1、2 はタッチセンサーとして使うことができる。

¹ Windows10 用の [MakeCode for micro:bit](#) がリリースされた。
MakeCode for micro:bit は、ワンクリックで micro:bit にアップロードできる。。



【温度センサー】 温度を測定できるセンサーがついている。

【加速度センサー】 本体をゆさぶった時に実行するプログラミングを組むことができる。

【コンパス】 本体の向いている方角を調べることができる。

【無線】 本体間でメッセージの送受信ができる。

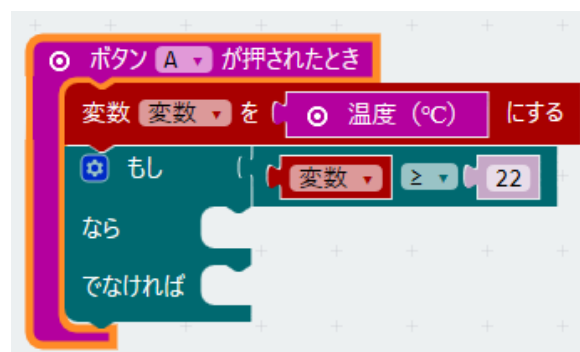
課題

第 1 段階 見本の通り、「バツ」「小さいハート」「チェック」「びっくり顔」を micro:bit に順に表示させる。

第 2 段階 ボタン A が押されたとき、「バツ」「小さいハート」を表示して消す。
ボタン B が押されたとき、「チェック」「びっくり顔」を表示して消すようにしなさい。

第 3 段階 ボタン A+B の両方が押されたとき、室内の温度を表示しなさい。
(基本→数を表示(0)→0 に温度を入れる)

第 4 段階 ボタン A が押されたとき、温度が 22 度以上の場合には、「うれしい顔」「warm」と表示、そうでなければ、「バツ」「cold」と表示するようにしなさい。



- ①デスクトップ上にある「scratch-microbit-1.0.hex」を micro:bit にアップロードする。
- ②Scratch Link を起動
- ③Scratch(3.0 ベータ版)を起動
- ④画面左下の拡張機能から micro:bit を選択し接続

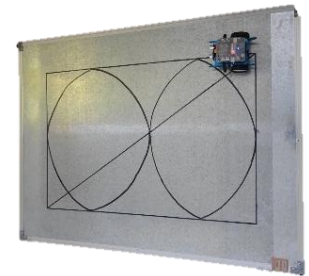
- ①カゴを A ボタンが押されたとき左側に、B ボタンの時は右側に移動するようにする。
- ②リングがカゴに触れたとき、micro:bit に「◇」のマークを 0.2 秒表示させる。
- ③止める時は、micro:bit を前に向ける。
- ④自分の工夫を入れる。

18

第 18 回講義概要

ロボット（mBot³）を使う

① 上下・左右矢印キーで動かす。



第 1 段階 右図の長方形、直角三角形に沿うように走行させる。後向きにも走行させる。

第 2 段階 速さを適時変更させる。

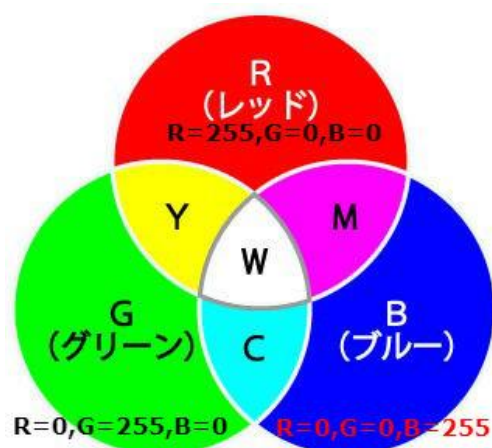
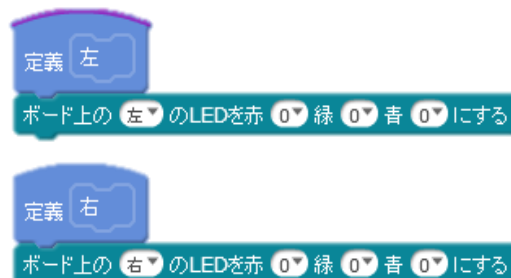
第 3 段階 自由に走行させる。

³ 参考文献：『Makeblock 公式 mBot で楽しむ レッツ！ ロボットプログラミング』
著者：久木田寛直 FOM 出版 2017/3/29

第 19 回講義概要

②LED ライトを光らせる。

- ・ 赤 (R) ・ 緑 (G) ・ 青 (B) ・ 黄 (Y)。左右に消灯のブロックを定義する。



第 1 段階 0.5 秒おきに、「赤 (R) ・ 緑 (G) ・ 青 (B)」の順番に 5 回色が変わるようにする。

第 2 段階 上記に「赤 (R) ・ 左消灯 ・ 緑 (G) ・ 右消灯 ・ 青 (B) ・ 全て消灯」を 5 回する。

第 3 段階 変数をつくり、赤がだんだん暗くなるようにする。

第 4 段階 左側の赤が徐々に暗くなるのと同時に、右側の青が徐々に明るくなるようにする。

前回に続いて

第 5 段階 前進する時には赤色を、停止する時には消灯する。

後進では緑色、停止には消灯。

左向きにする時には青色を、停止する時には消灯する。

右向きでは黄色、停止には消灯するようにする。

第 6 段階 自分の工夫で、動きや色の変化を追加する。

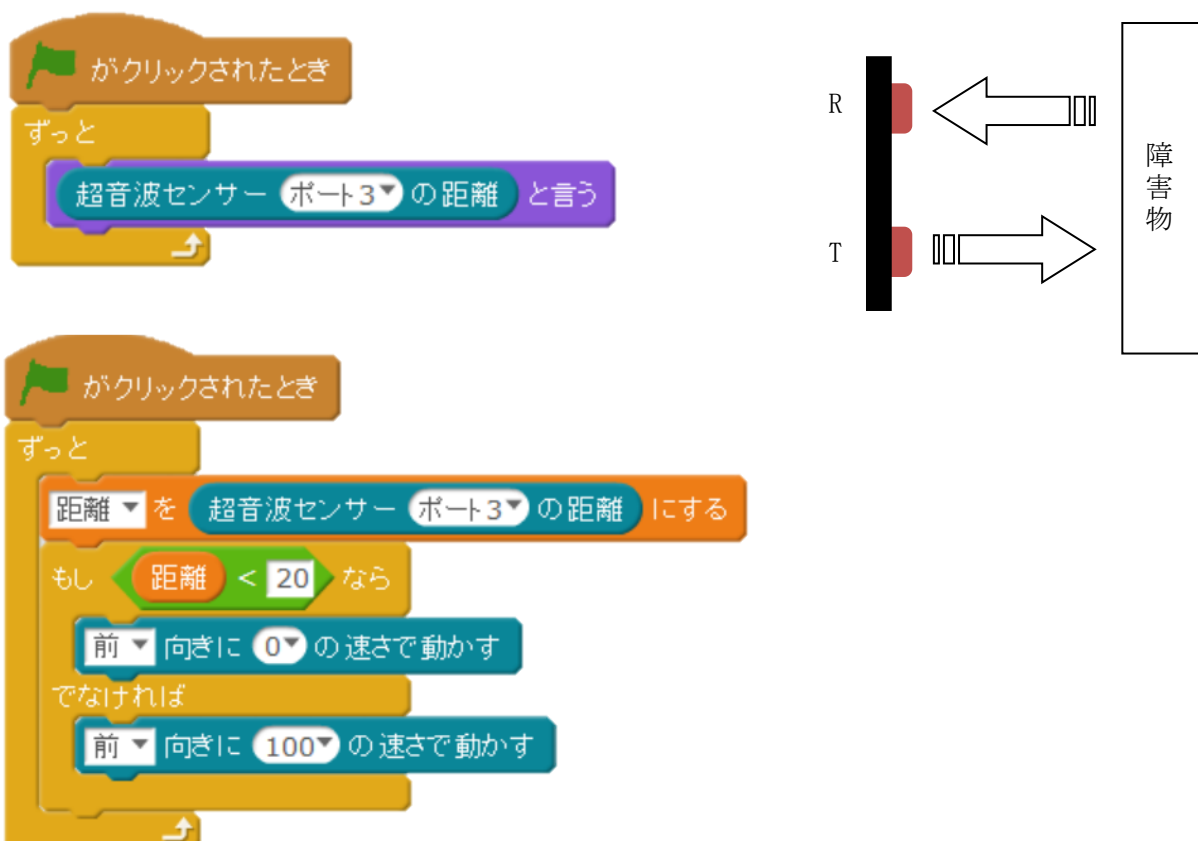
mbot は Scratch の追加機能として利用できるのも、事前に Scratch から学習しておく、使用する上で非常に有効的である。画面上の物が動くだけでは物足りなく感じるが、mbot を利用することで車を制御できるようになる。学習教材に幅ができる。

第 20 回講義概要

③超音波センサーを使う。(例：コウモリの目の働き)

正面から左目「T」(トランスミッター：送信機) 右目「R」(レシーバー：受信機)

単位は cm で、mbot の測定範囲は 3cm～4m



第 1 段階 天井までの高さを測定する。

第 2 段階 障害物を用意して、20cm 前まで走行させ停止させる。

第 3 段階 障害物を用意して、15cm までは全てが青色になり、止まった時には全てが赤色になり、その 3 秒後に赤色が消灯する。

第 4 段階 画面上のスプライトも一緒に走行させ、障害物の前で止める。

第 21 回講義概要

⑤障害物をよけながら走らせる。

乱数を使って左・右に動かす。



第 1 段階 上記のプログラムを入力して実行する。(障害物の適当に配置)

第 2 段階 障害物の 10cm 前でブザーを鳴らし、赤色の LED ランプで危険を知らせる。

第 3 段階 更に、右向きになったら青色をする。左向きになったら緑色にする。

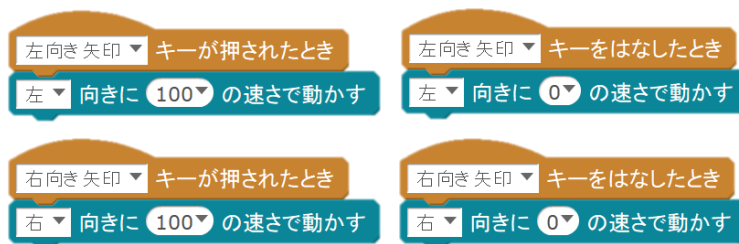
第 22 回講義概要

⑥ 高速回転



第 1 段階 上記のプログラムを入力して実行し、左向きにも回転するようにする。
100 を 255 に変える。

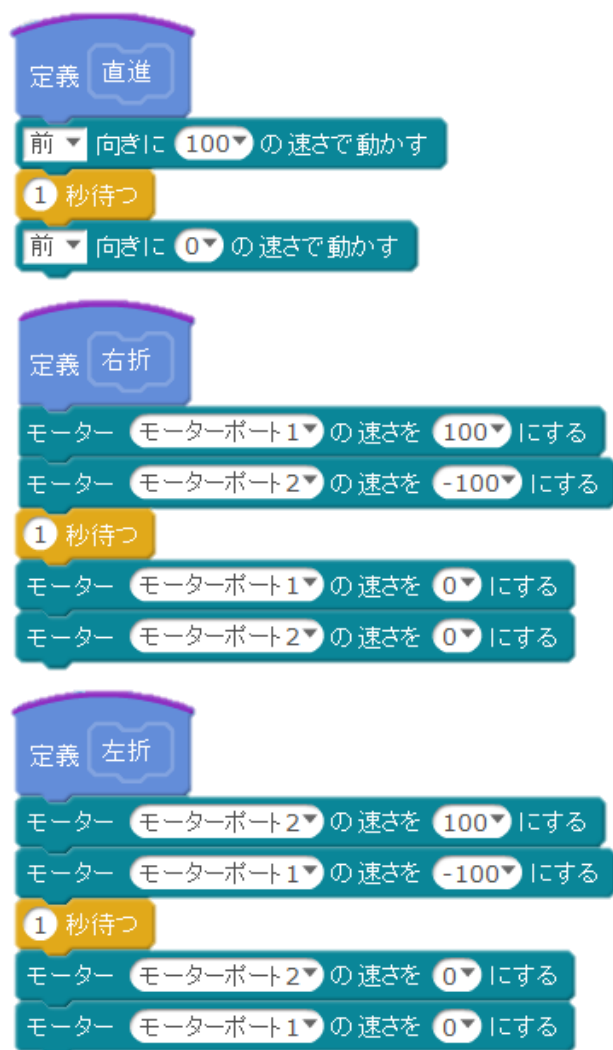
第 2 段階 以前使用した下記のプログラムとの違いを比較・検討する。



第 23 回講義概要

プログラムした通りに mbot を動かす。

図のように定義されたプログラムを利用して、タイルコースを走行させる。



⑦上記の①～⑥までの機能を組み合わせて、mBot を楽しく走行させる。