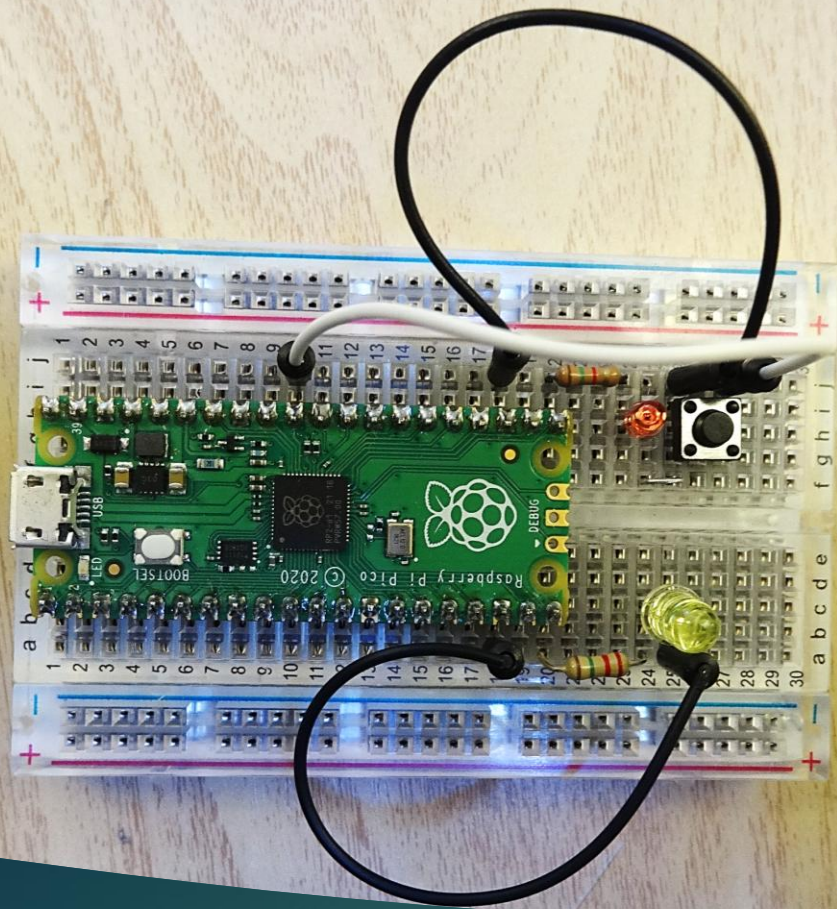




2025年1月度 集合活動 プログラミングゲーム

2025年 1月 19日 日本宇宙少年団 厚木分団



プログラミングとは？

プログラミングとは？

- ▶ プログラミングというとコンピューターを動かすための内容を作ることと言えます
- ▶ プログラミングの考え方はコンピューターを動かすだけではなく、人に何かをお願いする時などにも応用できます
- ▶ 今日は、ゲームを通してプログラミングの考え方を学んでいきます
- ▶ なお、皆さんはコンピューターを使わないで学びます

コンピューターはどうやって動かすの？

- ▶ 市販のパソコンやタブレットなどは、電源を入れたら、基本動作のために必要なプログラムが動作するので、すぐに使える状態になっています
- ▶ 売られているマイコンボードの場合は、電源を入れても何も動きません。なぜでしょうか？
- ▶ なぜなら、マイコンボードには何もプログラムが書かれていないからです。
- ▶ マイコンボードの場合は、やってほしいことをプログラムに書くことで、初めて使える状態になります。
- ▶ コンピューターは何かプログラムを与えないと何もしないというのが本来の姿です

プログラムには何を書くの？

- ▶ マイコンボードを動かす場合は、プログラムで、やりたいことを細かく指示する必要があります。
- ▶ コンピューターは自分自身に何がどのようにつながっているのかを把握していませんので、次のような情報を与えて動かします
 - ▶ コンピューターのどこにつながっているのか？
 - ▶ つながっているものが入力（スイッチなど）か、出力（ブザーなど）か？
 - ▶ 入力からどのような信号が来た時に反応するのか？
 - ▶ 出力を動かすためにどのような信号を出すのか？
 - ▶ 出力を出すための条件、出力の時間、入力の待ち時間など
- ▶ パソコンの場合はある程度基本のプログラムの中に、細かい処理の情報を持っているので、つなぐだけで動かせるようになっています。

プログラムの例（実際に動かしてみます）

```
1 from machine import Pin
2 import time
3
4 Led1 = Pin(15, Pin.OUT)
5 Led2 = Pin(16, Pin.OUT)
6 sw = Pin(26, Pin.IN, Pin.PULL_UP)
7 Lst = 0
8 val = 1
9
10 while True:
11     val = sw.value()
12     if val == 0 :
13         if Lst == 0 :
14             Lst = 1
15         else :
16             Lst = 0
17     Led2.value(Lst)
18     Led1.on()
19     time.sleep_ms(100)
20     Led1.off()
21     time.sleep_ms(100)
22
```

- 0.1 秒おきに黄色の発光ダイオードを点滅させる
- 0.2 秒おきにスイッチの状態を読んで、赤色の発光ダイオードを付けたり、消したりする

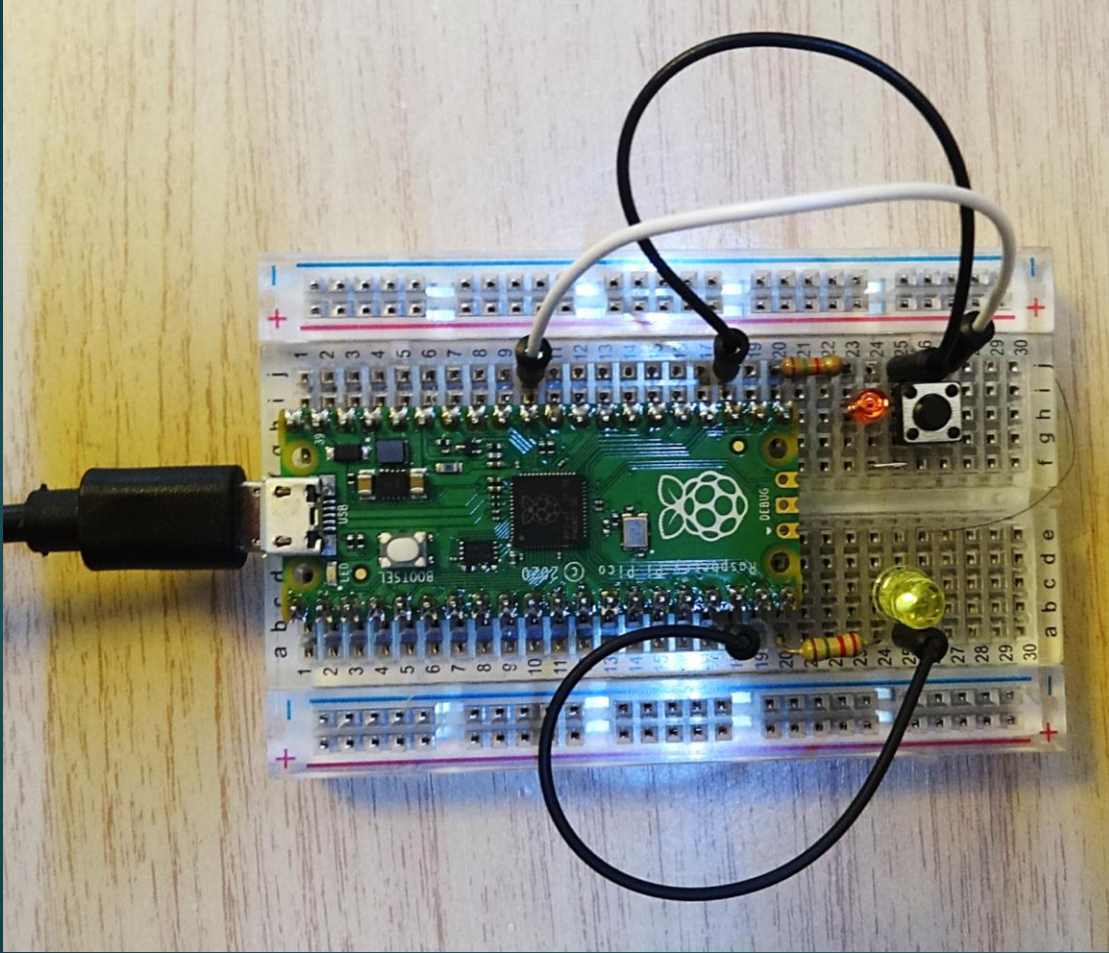
マイコンのどの端子に何がつながっているのかを表している
(入力[IN]: スイッチ、出力[OUT]: 発光ダイオード)

同じ処理を止めるまで繰り返す（止めないと無限に動く）

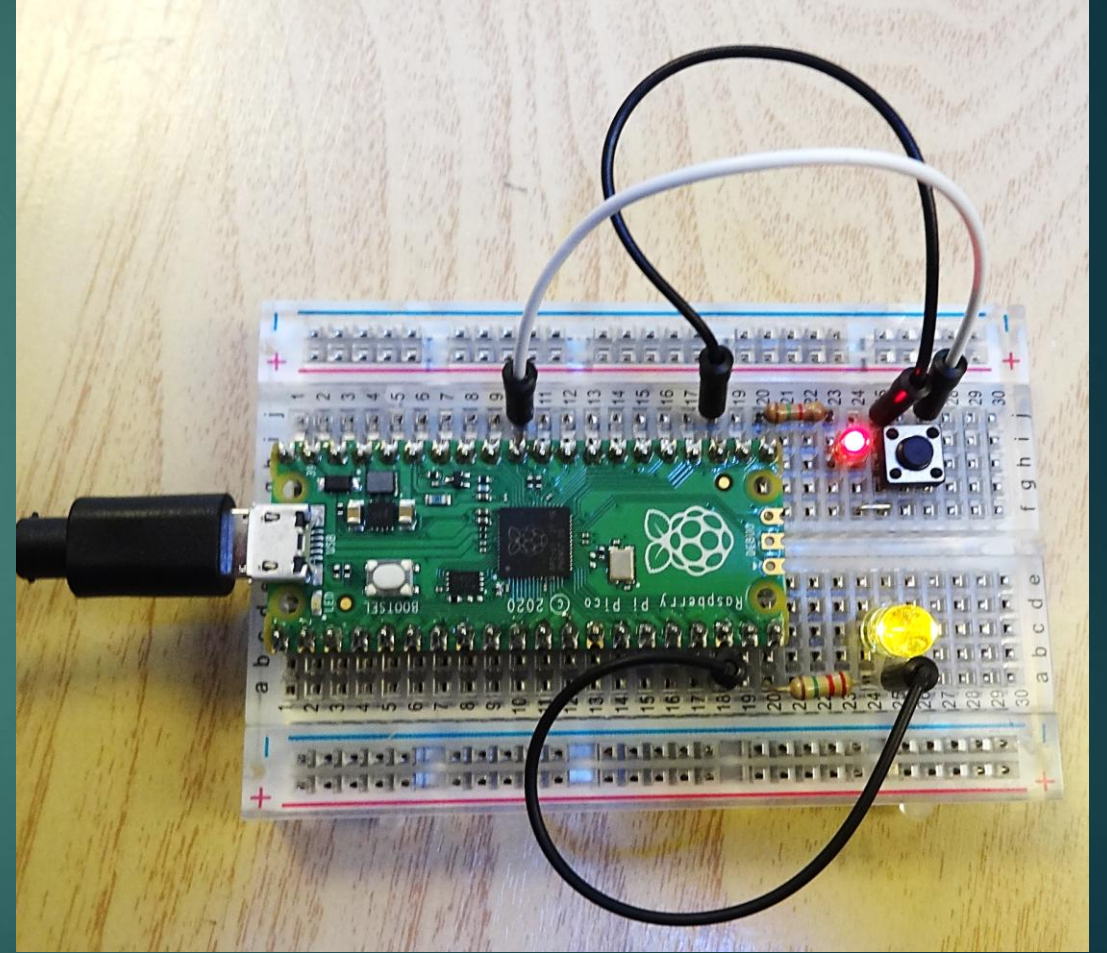
スイッチの状態を検出して、スイッチが押されていたら、赤色の発光ダイオードの状態を前の状態と変える

黄色の発光ダイオードを0.1秒おきに点滅させている
(0.1秒の間点灯、0.1秒の間消灯)

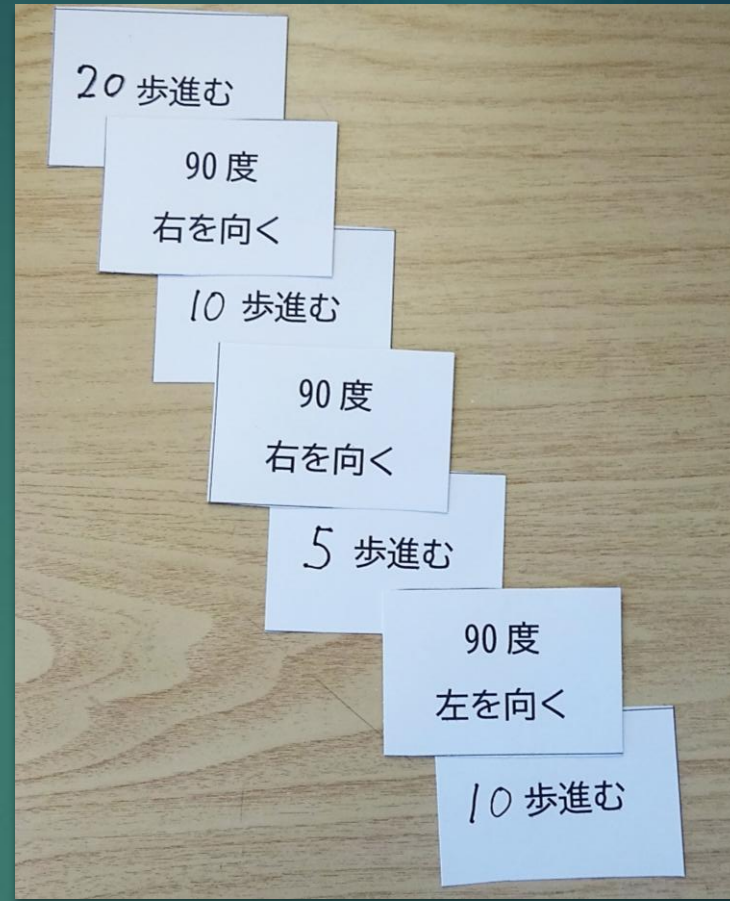
プログラムが動いているときと動いていないときの違い



プログラムがないとき



プログラムが動いているとき



プログラミングゲーム

コンピューターを使わないで、プログラムのしくみを学びます

今回のゲームの想定

- ▶ 皆さんは、コンピューターで動くロボットになったつもりでこのゲームに取り組みます
- ▶ コンピューターを動かすためのプログラムは紙のカードの形で与えられます
- ▶ カードを処理の順番に並べます
- ▶ そのプログラムに沿って、実際に動いてみて正しく動くことができるか試してみましょう
- ▶ 作ったプログラムでほかの人（ロボット）も動くのかどうか確認しましょう

ゲームのルール①

- ▶ ゲームとしてはロボット（人）を動かし、ものを動かすミッションを行います
- ▶ ロボットの移動は、床のカーペット 1 枚を 1 歩とします
- ▶ 歩くという指示が出たときは前にだけ動くことができます
- ▶ 右を向く、左を向くという指示はその場で回転します
- ▶ ものをつかむ、ものを離すという指示は指定のカードに沿って動きます
- ▶ 机やいすといった障害物は動かせません。障害物にぶつかったらどうすると脱出できるか、考えてください。

ゲームのルール②

- ▶ カードの中には「ぶつかるまで前に進む」というものがありますが、これは障害物にぶつかるまでは、歩数にかかわらず前に進めます
- ▶ カードの中には「ぶつかったら跳ね返る」というものがありますが、これは障害物にぶつかったら、これまで進んだ方向と逆を向き、動けるようにします
- ▶ 歩数については、カードに書き込んで使っても構いませんが、目的地にただどり付けるように、正しく数えてから記入します
- ▶ 1回のミッションの中では、一度使ったカードを、何回も使うことはできません。1回使ったカードは、後ろに回してください。

お願い事項

- ▶ ミッションカードは複数用意していますので、一人1枚選んでください
- ▶ ミッションにトライするときは、手を挙げて知らせてください
 - ▶ ミッション実施中は、じゃまをしないように待っているかコースの外にいてください
- ▶ 歩数を書けるカードは多めに準備していますので、必要な時はもっていつてください
- ▶ 保護者の方は、基本その場にいてください。動くときは、コースのじゃまにならないようにして移動をおねがいします。

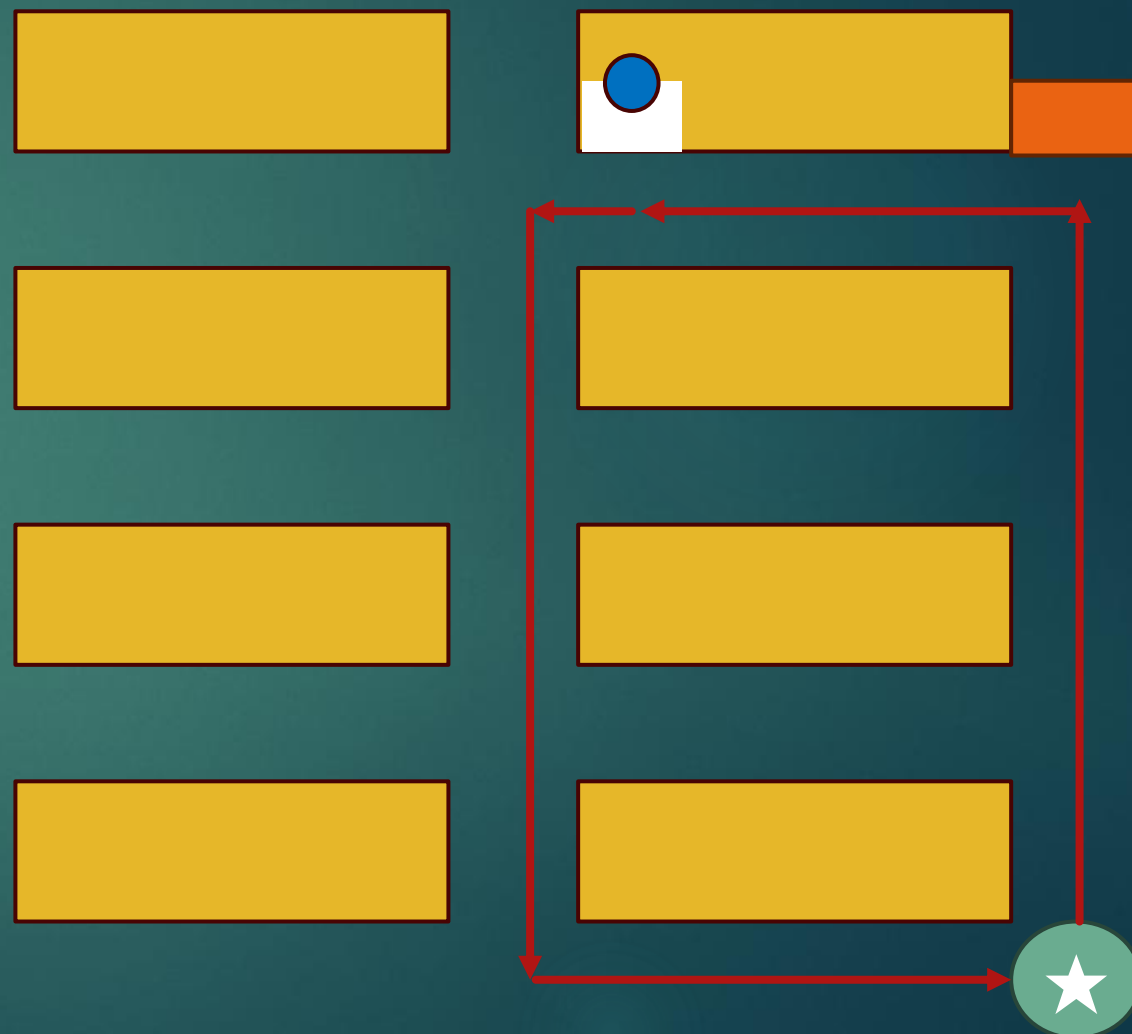


サンプルミッション

リーダーが見本を見せます

見本のミッション

1. 部屋の後ろからスタートします
(★印)
2. 障害物（いす）に当たるまで前に進みます
3. 机の端の白色のかごのところまで移動します
4. 白色のかごの青いボールを取ります
5. ボールを持って通路に出ます
6. 通路を頑張ってスタートに戻ります





皆さんでミッション
に取り組みましょう

ミッション達成したら、
リーダーに渡して試して
もらいましょう

ミッション No. 3

1. 透明のかごのところまで行く
2. 透明のかごから、青色のボールを取る
3. ボールを持って茶色のかごのところまで行く
4. 持っているボールを茶色のかごに入れる
5. 茶色のかごからオレンジ色のボールを取る
6. ボールを持って白色のかごのところまで行く
7. 持っているボールを白色のかごに入れる
8. スタート地点にもどる

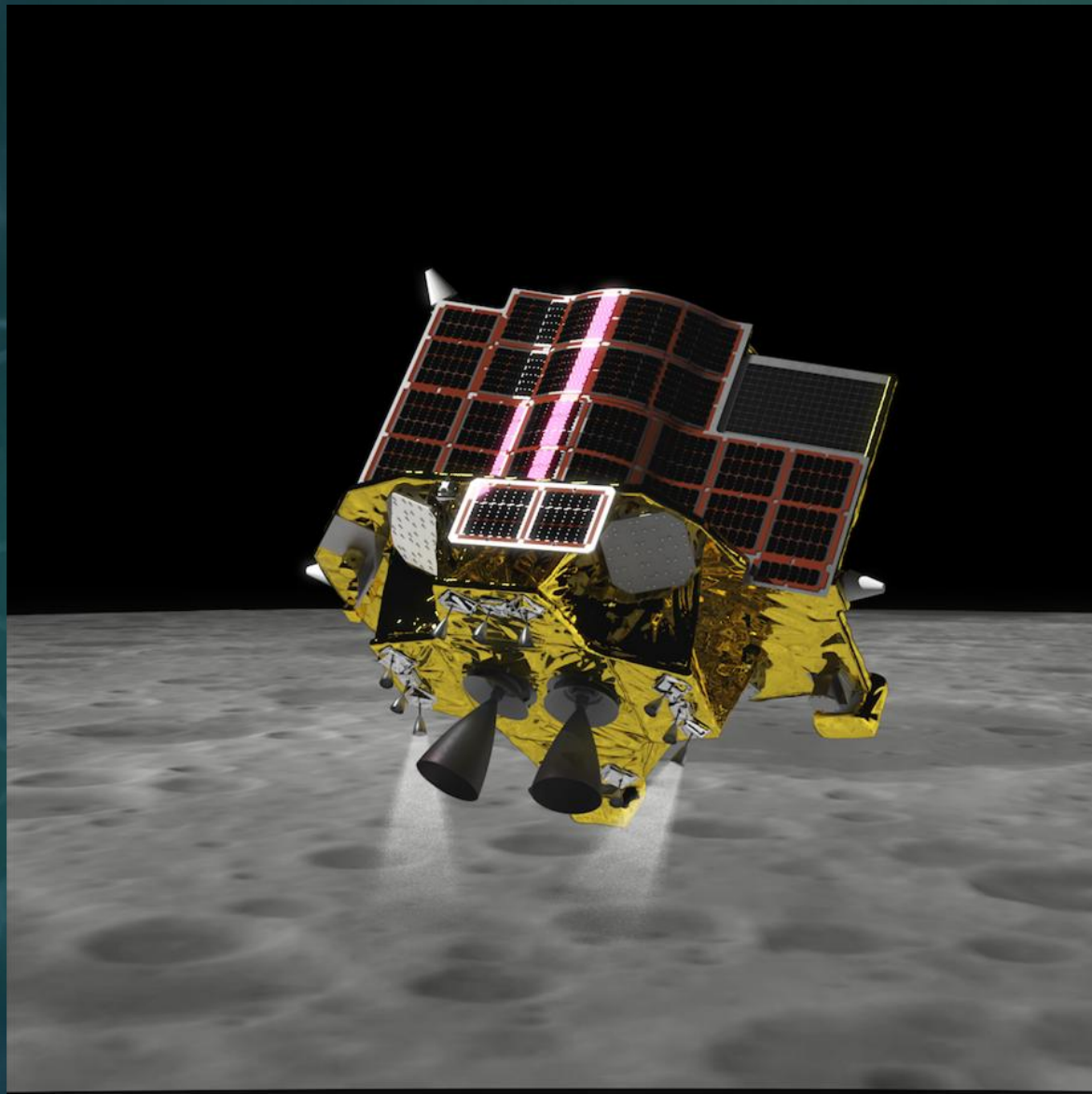
ミッション No. 10

1. 透明のかごのところまで行く
2. 透明のかごから、黒色の箱を取る
3. 箱を持って茶色のかごのところまで行く
4. 持っている箱を茶色のかごに入れる
5. 茶色のかごから緑色のボールを取る
6. ボールを持って白色のかごのところまで行く
7. 持っているボールを白色のかごに入れる
8. スタート地点にもどる

プログラミング ゲームの感想

今回の感想

- ▶ 与えられたミッションはクリアできましたか？
- ▶ できなかった人は何が課題だったのでしょうか？
- ▶ 自分がクリアできたミッションのカードをリーダーに渡したら、クリアできたのでしょうか？
- ▶ リーダーがクリアできない場合は、何か不足があるはずなので、理由を考えてみましょう



宇宙開発と プログラミング

宇宙開発とプログラミング

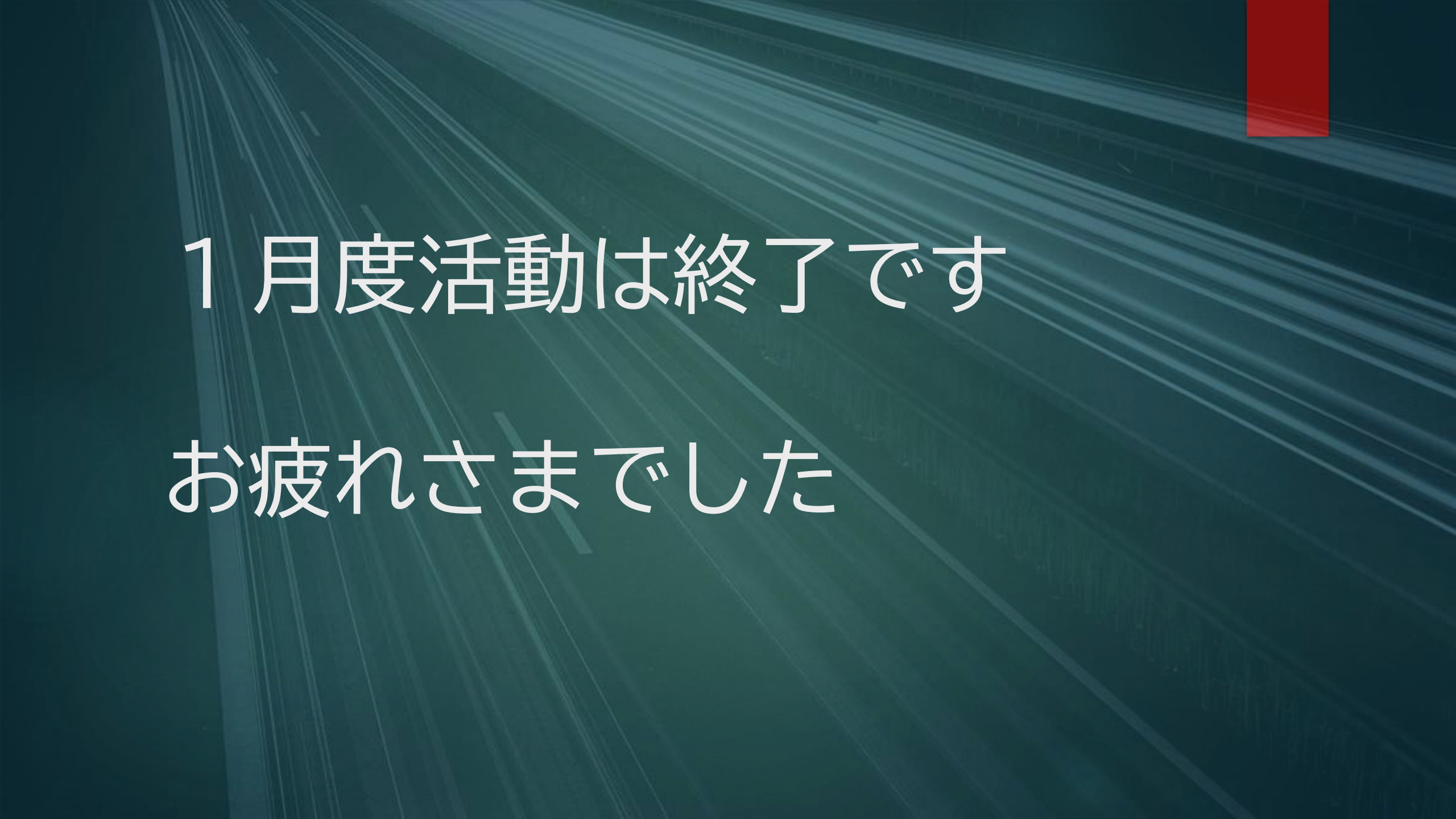
- ▶ 分団活動ではプログラミングを取り入れています
- ▶ 宇宙とプログラミングは何か関係があるのでしょうか？
- ▶ プログラミングとその考え方は、宇宙開発でも生かせる部分があります
- ▶ 今回は2つの事例で説明します

① 無人探査ミッション

- ▶ 月などを調査する探査機に関しては、着陸の指示は地球から送りますが、細かい操作は地球からは行わず、探査機のコンピューターがロケットの状態を把握して細かい制御を行っています
- ▶ 月や惑星から、地球に電波が届くまで数秒～数分かかります
- ▶ 地球から指示を出して動かそうとした場合、探査機からの状態に合わせた指示を送っても、指示が届くまでの間に探査機は前の指示に沿って動いています。
 - ▶ そのため急な変更を行っても反映されるまである程度の時間がかかってしまう
- ▶ 細かい指示を行っても、指示が届いて処理されるまでの時間、探査機が進んでしまいますので、実際には、指示がなくとも探査機側で考えて、探査機の動きを制御して、確実に着陸できるような仕組みを持っています

② 有人宇宙飛行ミッション

- ▶ 人が乗ってロケットを飛ばす場合、ロケットの操作を間違えると乗っている人の命にかかわる可能性があります
- ▶ 宇宙船を飛ばす際には確認すべきこと、行すべき作業がたくさんあり、作業がもれたり、順番が違ってしまうと、人の命にかかわることもあります
- ▶ 宇宙飛行士やロケットの管制官があやまった作業を行わないように、宇宙船やロケットの操作の手順については細かく書かれた指示書があります
- ▶ 指示書の中で、必要なことが、正しい順番で細かく書かれているので、それに沿って作業を行うことで安全な飛行を行うことができます
- ▶ 指示書を書くときは、プログラミングの考え方が役に立ちます

The background features a dark teal color with numerous light blue and white diagonal streaks that create a sense of motion or light trails. In the top right corner, there is a solid red rectangular block.

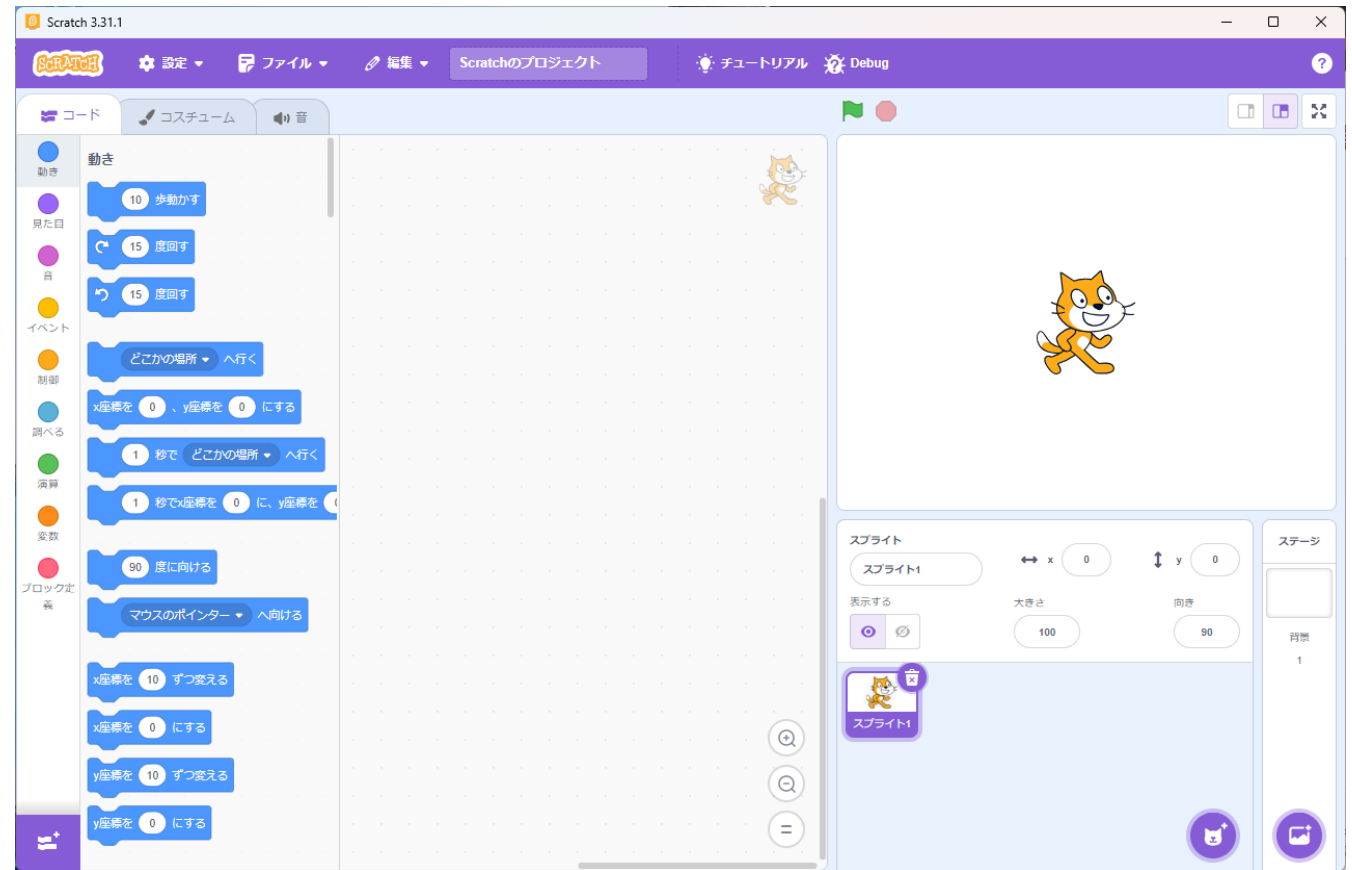
1 月度活動は終了です
お疲れさまでした



以降、リーダー向け補足資料

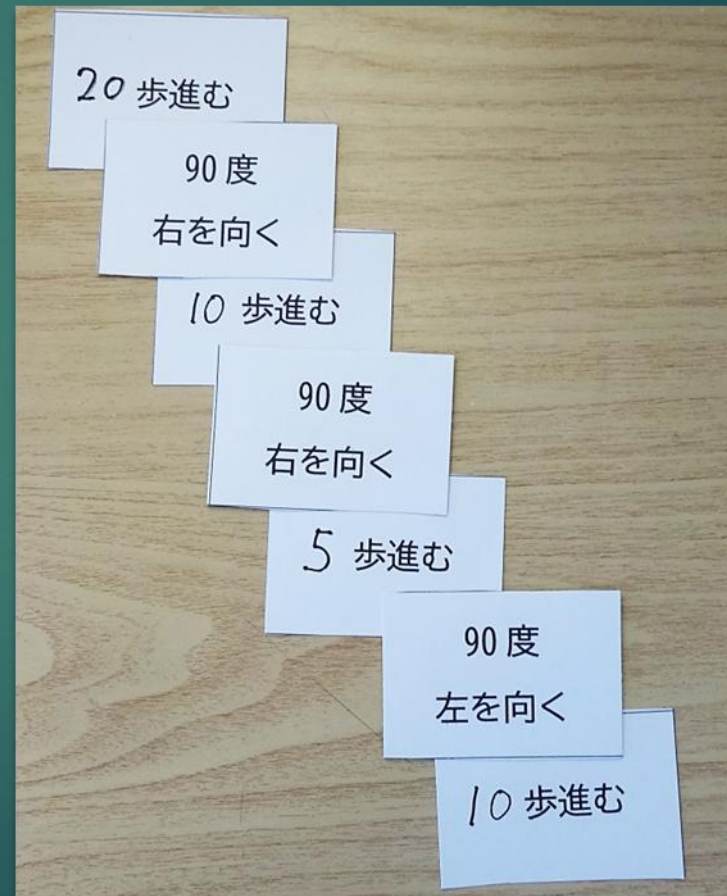
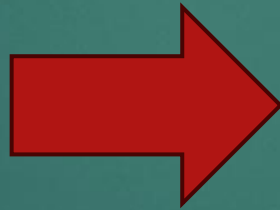
活動のヒント： 考え方（1/2）

- ▶ 今回の活動ですが、子ども向けのプログラミング学習でよく使われている「Scratch」（スクラッチ）のプログラムの考え方を紙のカードに置き換えたものです



活動のヒント：考え方（2/2）

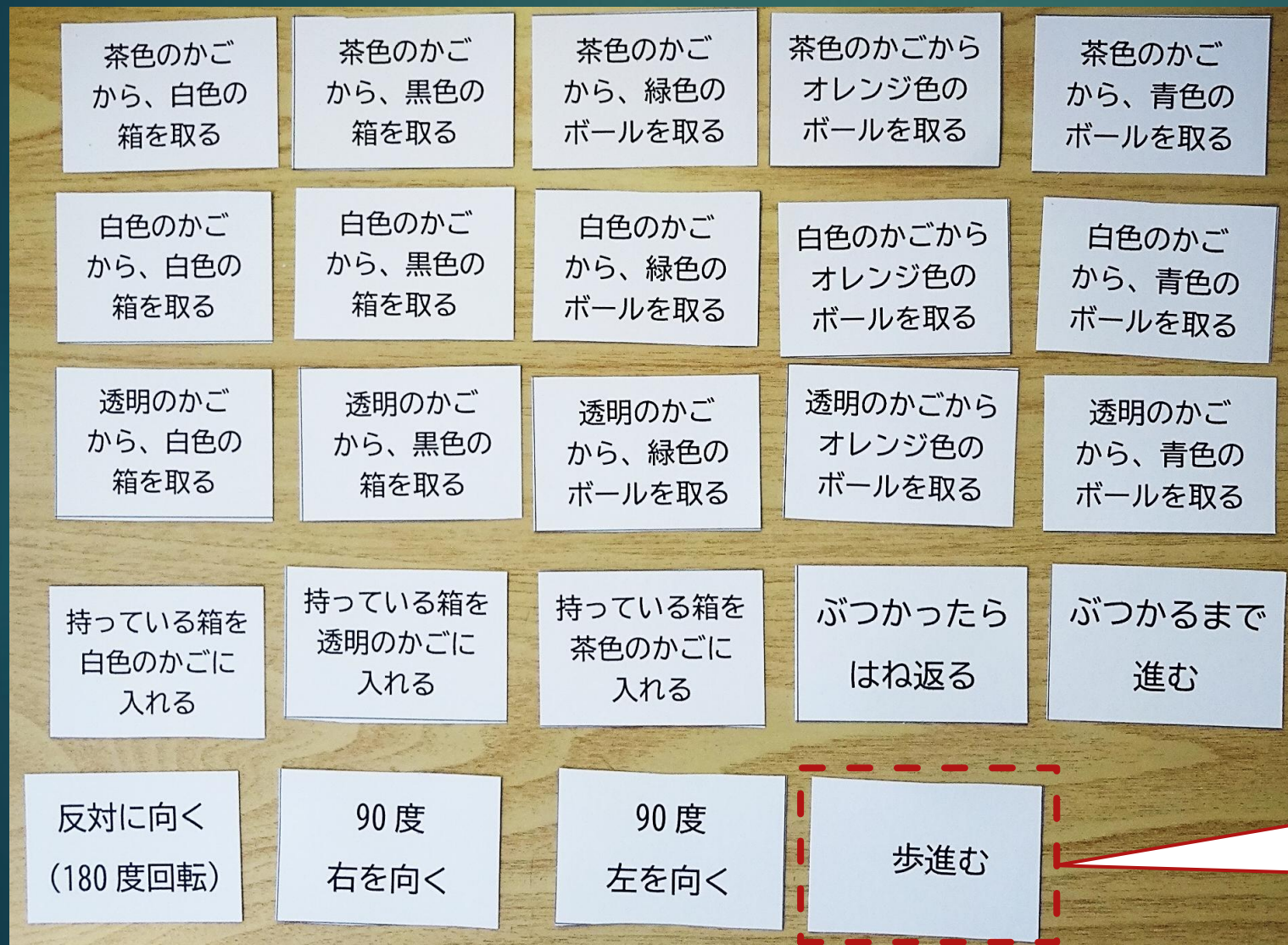
- ▶ Scratchでは左の図のようにブロックを並べてプログラムを作りますが、今回の学習では右の写真のようにブロックを紙のカードに置き換えています



活動のヒント：カード準備

- ▶ Scratchのブロックを参考にプログラム用のカードを準備します
- ▶ カードは、画用紙など少し厚手の紙に印刷するとよいです
- ▶ 場所の広さなどでも変わるので、移動距離は各自で書き込んでもらうようにします
- ▶ それ以外のカードは、すべてパターンを決めておきます
- ▶ カードは多めに準備してもよいです
 - ▶ 少なめにするほど難易度が上がりますが、ミッションシートの内容も加味して枚数を決める必要があります

実際に準備したカード



ミッションカードに書かれた内容を達成できるように、これらのカードを組み合わせ、プログラムを作る

このカードは、多めに準備して、必要に応じて歩数を書いてもらう

活動のヒント：ミッション準備

- ▶ やってもらうこととしては、部屋のどこかから、別の場所にものを移動させるというミッションがわかりやすいと思います
 - ▶ 移動元、移動先に入れ物（かごや箱）を用意することで、問題のバリエーションを増やすことができます
- ▶ 団員にいろいろ考えてもらうように、回答は1つに絞れないようなルート設定にしてもよいと思います
- ▶ 他の人の行動を見てまねることを減らすよう、ミッションのパターンは団員の数（またはチームの数）より多く準備した方がよいです
- ▶ 当日、障害物を置いたりしても問題のないようにミッションを考えます

ミッションカードの例

ミッション No. 10

1. 透明のかごのところまで行く
2. 透明のかごから、黒色の箱を取る
3. 箱を持って茶色のかごのところまで行く
4. 持っている箱を茶色のかごに入れる
5. 茶色のかごから緑色のボールを取る
6. ボールを持って白色のかごのところまで行く
7. 持っているボールを白色のかごに入れる
8. スタート地点にもどる

ミッション No. 3

1. 透明のかごのところまで行く
2. 透明のかごから、青色のボールを取る
3. ボールを持って茶色のかごのところまで行く
4. 持っているボールを茶色のかごに入れる
5. 茶色のかごからオレンジ色のボールを取る
6. ボールを持って白色のかごのところまで行く
7. 持っているボールを白色のかごに入れる
8. スタート地点にもどる

実際のミッションの組み合わせシート

ミッションカード作成メモ									
	最初の手順		2 番目の手順			3 番目の手順			
ミッション番号	目的地	取るもの	目的地	戻すもの	取るもの	目的地	戻すもの	取るもの	ゴール
No. 1	茶色のかご	緑のボール	白のかご	緑のボール	青のボール	透明のかご	青のボール		スタート
No. 2	白のかご	オレンジのボール	透明のかご	オレンジのボール	緑のボール	茶色のかご	緑のボール		スタート
No. 3	透明のかご	青のボール	茶色のかご	青のボール	オレンジのボール	白のかご	オレンジのボール		スタート
No. 4	透明のかご	白の箱	白のかご	白の箱	緑のボール	茶色のかご	緑のボール		スタート
No. 5	白のかご	黒の箱	透明のかご	黒の箱	青のボール	茶色のかご	青のボール		スタート
No. 6	白のかご	緑のボール	透明のかご	緑のボール	黒の箱	茶色のかご	黒の箱		スタート
No. 7	茶色のかご	オレンジのボール	白のかご	オレンジのボール	黒の箱	透明のかご	黒の箱		スタート
No. 8	茶色のかご	青のボール	白のかご	青のボール	白の箱	透明のかご	白の箱		スタート
No. 9	茶色のかご	白の箱	透明のかご	白の箱	オレンジのボール	白のかご	オレンジのボール		スタート
No. 10	透明のかご	黒の箱	茶色のかご	黒の箱	緑のボール	白のかご	緑のボール		スタート
No. 11	透明のかご	緑のボール	茶色のかご	緑のボール	オレンジのボール	白のかご	オレンジのボール		スタート
No. 12	透明のかご	オレンジのボール	茶色のかご	オレンジのボール	白の箱	白のかご	白の箱		スタート
No. 13	白のかご	青のボール	透明のかご	青のボール	黒の箱	茶色のかご	黒の箱		スタート
No. 14	白のかご	白の箱	茶色のかご	白の箱	青のボール	透明のかご	青のボール		スタート
No. 15	茶色のかご	黒の箱	白のかご	黒の箱	白の箱	透明のかご	白の箱		スタート

活動のヒント：ルール作り

- ▶ カードに書く歩数の考え方はそろえた方がよいです
 - ▶ 会場のタイルやカーペットシート1枚分=1歩とするなど
- ▶ 回転の方法や跳ね返るときの動きなども定義します
 - ▶ 再現性を高めるため勝手に解釈を変えることがないようにする
- ▶ 小学校低学年だと、角度で書いてもわからないので、説明をうまく考える必要があります
 - ▶ 写真で説明したり、リーダーが見本を示すような形がよいと思います
 - ▶ 必要に応じてミッションシートにフリガナを振るなどの工夫も必要かと思います
- ▶ 後でほかの人が再現できるよう、カードの使いまわしをなしにしています
 - ▶ もう一回上から順にたどることで、ほかの人も同じように動けます
- ▶ コースの障害物・入れ物はイベント中は動かさないようにします
 - ▶ 保護者・見学者も障害物に見立てるので、場所を変えないように依頼します

イベントの進め方

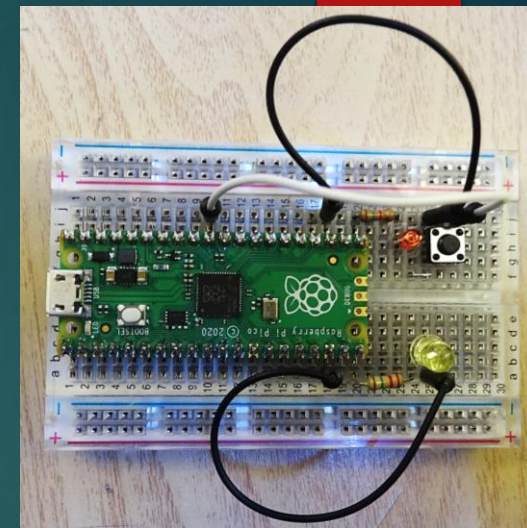
- ▶ まずは、ゲームがどのようなものか説明します
- ▶ 次にルールの説明を行います
 - ▶ リーダーが見本を示しながらルール説明を行います
- ▶ そのうえで、ミッションカードを配って、プログラムを考えてもらいます
 - ▶ 考える時間はしっかりとってください（30分～1時間）
 - ▶ 準備ができたら団員から声をかけてもらうようにします
- ▶ 完成したプログラムを（作った人が）実際に試してもらいます
 - ▶ 想定の通り、動くかどうか確認します
- ▶ うまくできたら、ほかの人（リーダーでも可）に同じプログラムで動けるかどうか試してもらいます

前段のデモについて

- ▶ デモは、コンピューターはプログラムを書かないと何もしないということを目の前で見て、理解してもらうことが目的です
- ▶ 前段のデモはマイコンボードを準備して、まずはプログラムを書かない状態で電源を入れどうなるか見てもらいます
- ▶ そのあとで、プログラムを書いてから再度マイコンボードを動作させ、変化を確認してもらう形にしています
- ▶ 今回は、厚木分団のほかの活動（次世代プロジェクト会議中級編）で使っているマイコンボード、プログラムを利用しています

デモで使った機材（一例）

- ▶ マイコンボード：Raspberry pi Pico
 - ▶ 発光ダイオード：赤と黄色を使用
 - ▶ スイッチ：タクトスイッチを使用
 - ▶ その他：電流制限用の抵抗、ジャンパー線、ブレッドボードを準備
-
- ▶ プログラミング環境
 - ▶ プログラム言語：Micro python
 - ▶ プログラム作成：パソコンに「Thonny」をインストールして実施
 - ▶ Raspberry pi picoとパソコンをUSBケーブルでつないで書き込みを実施
 - ▶ プログラミング環境は他でもよいですが、その場合は説明資料をその環境に合わせたものに置き換えます



```
Thonny - C:\Users\sunao\新しいフォルダー\LED_SW_test.py @ 23:1
ファイル 編集 表示 実行 ツール ヘルプ

LED_SW_test.py
1 from machine import Pin
2 import time
3
4 led1 = Pin(15, Pin.OUT)
5 led2 = Pin(16, Pin.OUT)
6 sw = Pin(26, Pin.IN, Pin.PULL_UP)
7 lst = 0
8 val = 1
9
10 while True:
11     val = sw.value()
12     if val == 0 :
13         if lst == 0 :
14             lst = 1
15         else :
16             lst = 0
17         led2.value(lst)
18         led1.on()
19         time.sleep_ms(100)
20         led1.off()
21         time.sleep_ms(100)

Shell
MicroPython v1.24.1 on 2024-11-29; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
MicroPython v1.24.1 on 2024-11-29; Raspberry Pi Pico with RP2040
Type "help()" for more information.
>>>
```

パソコンだけでのプログラムの有無説明

- ▶ 廃棄予定で、起動ディスクのデータを削除しているパソコンがある場合は、そのパソコンの電源を入れると、起動不可の画面が出ます（左写真参考）
- ▶ それをもってプログラムが動かないことの説明ができます
- ▶ パソコンの通常の起動画面（右図）と比較してみます

