スロットマシンを考えよう！

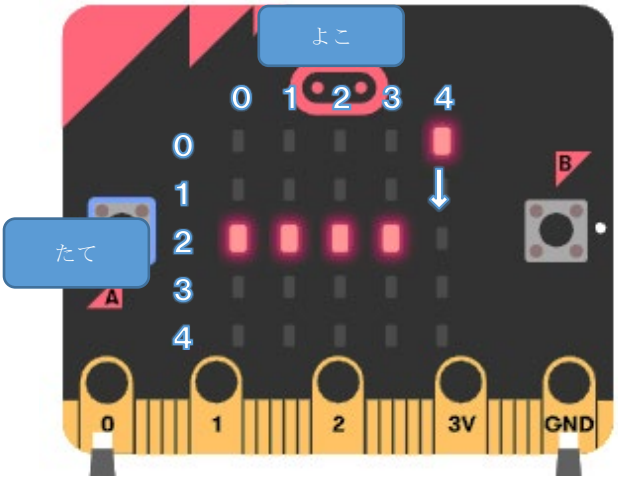


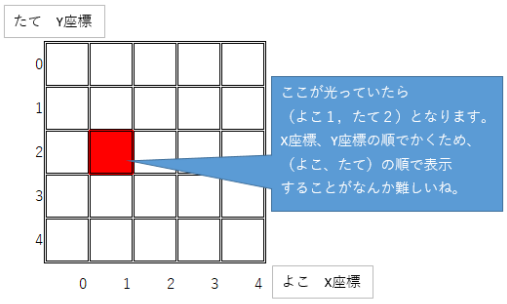
図１　マイクロビットの LED の番号

マイクロビットの LED が横に一列揃ったら当たりというスロットマシンゲームを、算数の知識を活用してつくってみましょう。

1.どうしたら、上から順番(じゅんばん)に LED が点灯するようになるの？！

図1を見てみましょう。マイクロビットには LED が5×5の25個が付いています。これらには、番号が振(ふ)られており、一番左上が、<u>よこが0、たてが0で(0,0)</u>と表すことが多いです。また、
<u>よこは X</u>、<u>たては Y</u> で示すことが多いので覚えておきましょう。

よこ0のたて0から下に順番に LED を表示させて、まるで光が振ってくるようなプログラムをつくってみましょう。



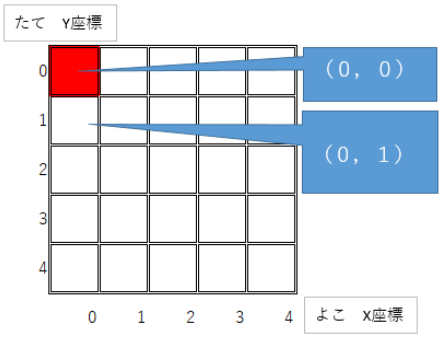
まず、よことたての数字を入れる箱をつくりましょう。この箱のことを「**変数（へんすう）**」と言います。図2を参考に変数の「よこ」と「たて」を追加しましょう。

図２

命令のブロックは、ある程度まとまったグループにまとめられています。今回は命令のグループ 変数 から、変数を追加しました。

変数を追加したら、図3のように、プログラムをつくってみましょう。<u>数字は半角で打たないと失敗</u>しますので、注意してください。

図３



プログラムを見る前に、LED の表示が上から下に流れる手順を確認しましょう。

(0,0)の LED を点ける
ちょっと待つ
(0,0)の LED を消す
たての数字に1を足す(ここでは0＋1)

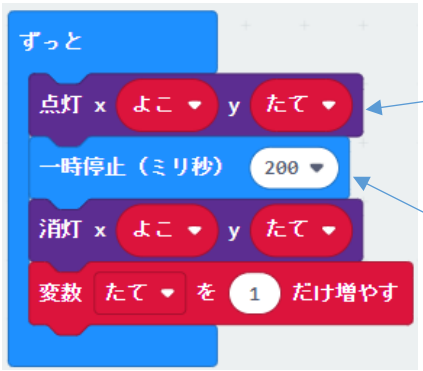
ここで１セット

(0,1)の LED を点ける
ちょっと待つ
(0,1)の LED を消す
たての数字に1を足す(ここでは1＋1)

ここで１セット

(0,2)の LED を点ける
・
・
・

というように、たての変数に、数字の1を足していけば、順番に表示することができます。



LED の点灯、消灯は命令のグループ LED の、
点灯 x 0 y 0 と 消灯 x 0 y 0 です。
一時停止ブロックは命令のグループ 基本 の下の方にあります。

図４

プログラムにすると図4のように表すことができます。ちなみに「ちょっと待つ」では、200 ミリ秒(0.2 秒)にしてみましたが、数字を変えて試してみると、動作をより理解できると思います。

2.どうしたら、下まで点灯した LED が上に戻ってくりかえすの？！

問題

では、一番下(0、4)まで下りた次はどのようにすれば良いでしょうか？
考えてみましょう。

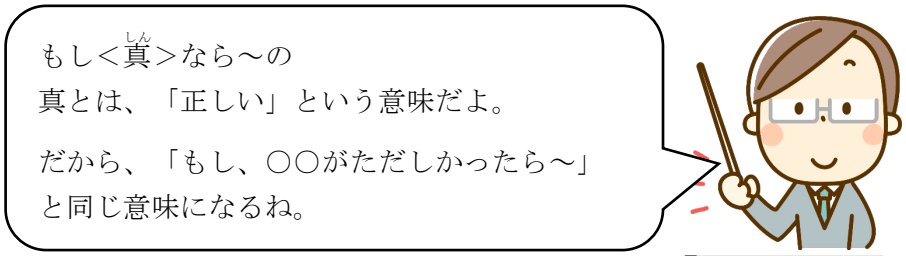
考えをここに入力しよう。
→

解答例

解答例
この青い図形をずらすと、解答が見られます。

命令のグループに 論理 (ろんり)グループがあります。その中に下のようなブロックがあると思います。これは**「もし○○ならば▲▲しなさい」**というブロックです。

図５



ここで、**「もし、たての数字が5ならば」**という命令をつくります。命令のグループ 論理 にある下の図 6 のブロックと、命令のグループ 変数 でつくった、たての変数を利用して、図7をつくります。できる人はできあがった図を見ないでつくり、実行してみましょう。

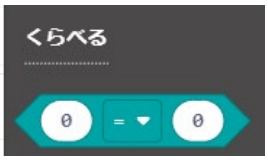


図６

間違っても大丈夫。間違った方が、「なぜ間違ったのだろう？」と考えるため、プログラミングの力が身に付きます。失敗を恐れず、どんどん実行しながら思い通りに動くようにしてみましょう。

解答例

図７
解答例

この青い図形をずらすと、
解答が見られます。

3.ここまで出来たら、ボタンでとめたいよ！どうすればいいの？！

ここまで出来たら、ボタンを押したら、ぴたっと止まるプログラムをつくりたくなりますよね？

改良プログラム

次につくるプログラムは完成版では使いませんが、ボタンを押したら止まるプログラムをつくります。

「もし A ボタンが押されていたら、1秒間プログラムを止める」という命令をつくります。

命令のグループ 入力 にある ボタン A が押されている を新たに使います。

一時停止は命令のグループ 基本 にあります。

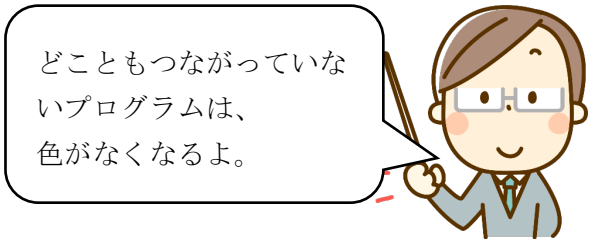


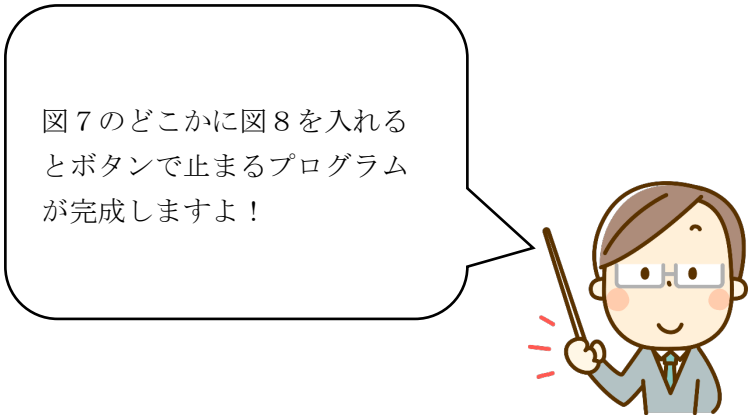
図８

図8は、**「もし A ボタンが押されていたら、1秒間プログラムを止める」**ためのプログラムです。

何ともつながっていない状態でプログラムをつくると、右のように色がなくなります。これは「このプログラムは実行しても動きようがないよ」ということを示しています。

図7でつくったプログラムのどこに入れたら、ちゃんと動くでしょうか？考えてから入れても良いですし、とりあえず実行してみても良いでしょう。大切なのは「あ、そうか○○だから動かないのか」など、振り返ることです。解答例は図9となります。

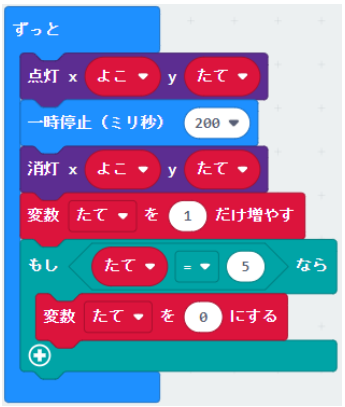
図９
この青い図形をずらすと、
解答が見られます。

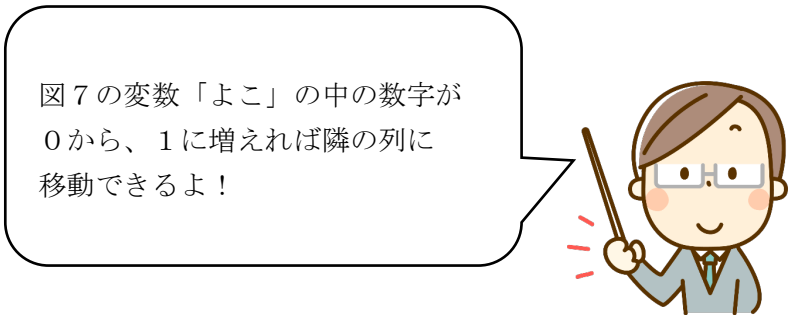


4.となりの列に移動したいよ！どうすればいいの？！

いよいよ、となりの列の LED が流れるように点灯するプログラムをつくります。ちょっと論理的な力が必要となりますので、頑張りましょう。

A ボタンを押したら、次の列に移動するため、変数よこが、0から1になります。図 7 でつくった下の図を見てみましょう。なんと素敵なことに、よこが変化してもこのプログラムは変更しなくても良いではないですか？！





つまり、**「A ボタンが押されたら、押されたときの LED は点灯したまま、となりの列に移動する」**命令を追加すれば良いことになります。今度は、命令のグループ 入力 の、下の図10 を使います。

図 10

今回は、完成形を見てしまいましょう。

図 11

図11のとおり、今光っている LED を点けたまま、次の列へ移動（よこ＋1）するというだけのプログラムです。

LED の点灯処理を行っている理由は、ほんの少しの時間ですが、LED が消灯する時間が発生してしまうからです。このバグを無くすための処理として、LED を確実に点灯させています。これで感動の完成です！（※意味が分からない場合は、点灯のブロックを削除して実行してみましょう）

しかし、これだけですと、一度プレイしたらまた、電源を入れ直さなければなりません。そこで**「B ボタンを押すと、初めからやり直す」**というプログラムを追加します。プログラムは図12 です。

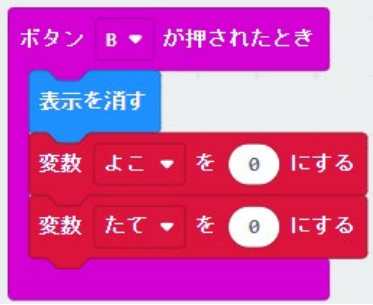


図 12　Ｂボタンを押したら、初めからやり直す

　全ての表示を消して、変数の「よこ」と「たて」を0にすれば、元通りになりますよね。これで、最初からやり直すことが出来ます。

　※このあと、いろいろな変数が出てきますが、同じように0を入れれば良いのです。これを**「初期化（しょきか）」**といいます。

　ここまでできたら、＜とりあえずの＞完成です。お疲れ様でした。

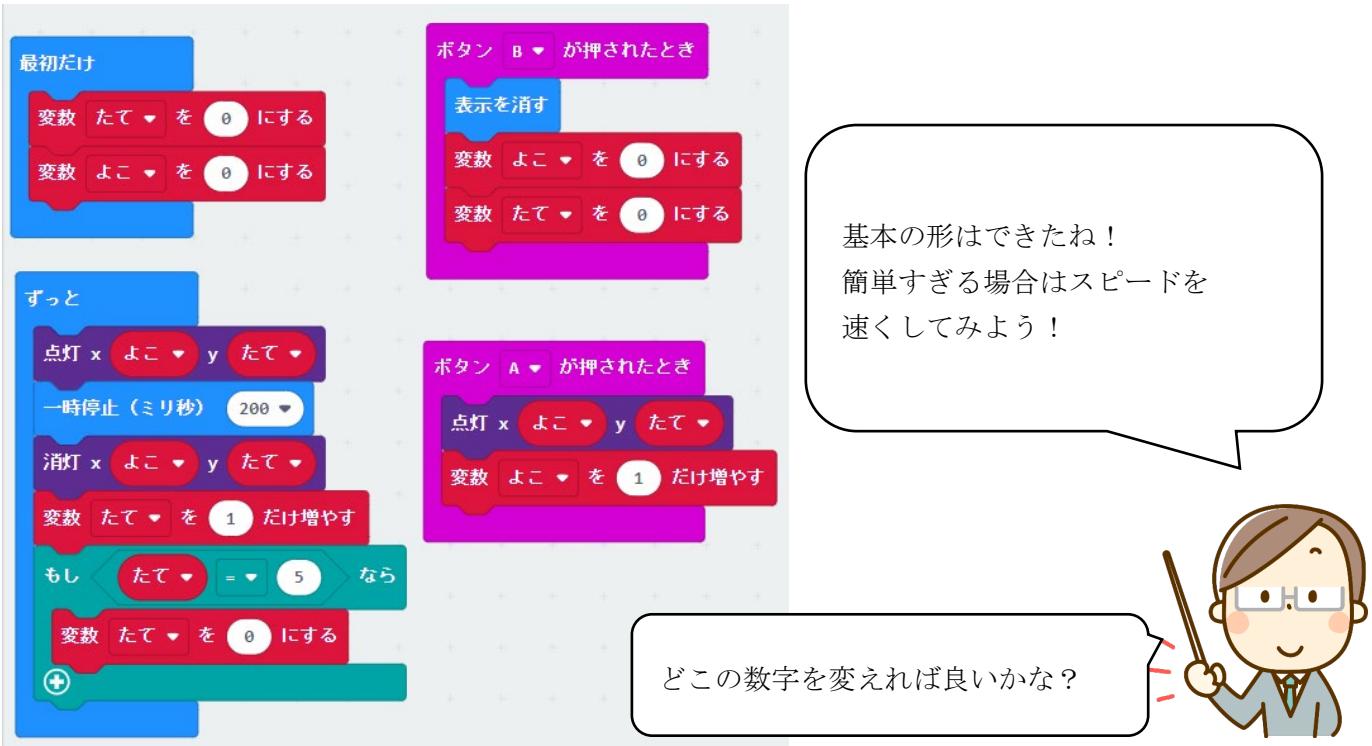


図 13　完成したプログラム１（スロットマシン＜当たり判定なし＞）

やってみよう

　スロットがゆっくり回転したり、高速に回転したりするように数字を変えてみよう。ぎりぎりそろえられるスピードを見つけたら、友達と交互に使って、どちらが先にそろえられるか、対戦しよう。

発展　※ここからはプログラムの命令のブロックがどこにあるかの説明は少なくなります。

5.もっとゲーム性があると楽しいよ！どうすればいいの？！

ここまでできたら、**「LED の点灯が横一列にそろったら、当たり判定をする」**プログラムもつくりたいですね。
もし、そろった数字が、

０００００
１１１１１
２２２２２
３３３３３
４４４４４

だったとしたら、どのようにしたら当たり判定すればよいでしょうか？

「もし、00000なら当たり、もし11111なら当たり、もし22222なら当たり、もし33333なら当たり、もし44444なら当たり」というプログラムをつくっても良いかもしれません。しかし、もう少し論理的にできないでしょうか？

問題

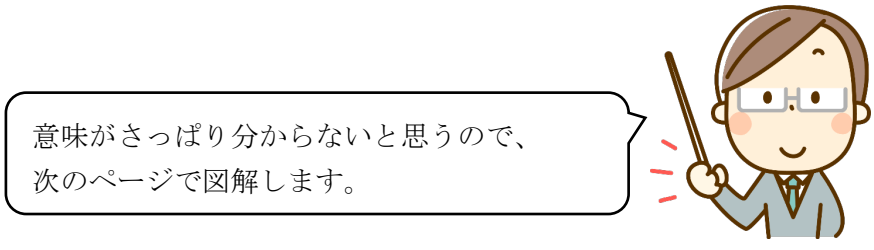
それぞれの数字を割ったときにあまりが0であり、また、他の数字の時にはあまりが必ず出る数字とは何だろうか？　ちなみに00000は何で割っても0あまり0です。

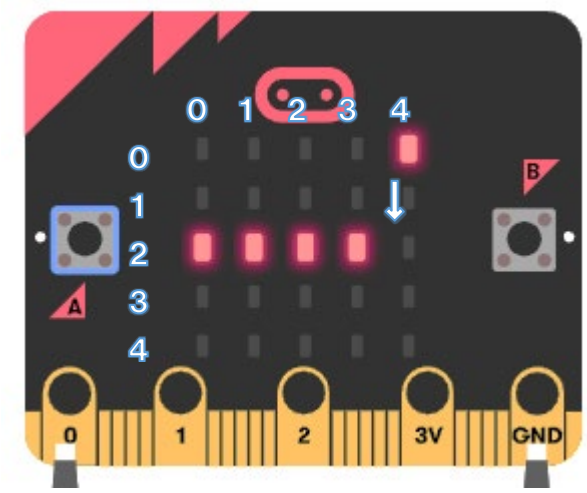
答え

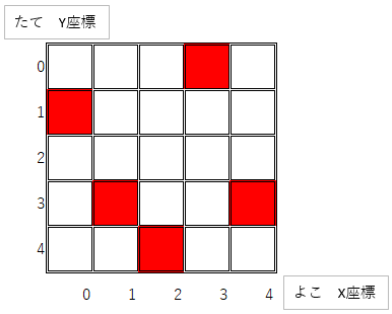
答えをここに入力

解答
この青い図形をずらすと、解答が見られます。

しかし、このスロットでは今のところ、たての位置と、よこの位置を記憶させる数字しか使っていません。
そこで、よこ0を1の位、よこ1を2の位、よこ2を3の位として足し算していけば、
33333のような数字がつくれます。







↑このように止まった時を出目が
「３０４３１」にすると考えます。

図 14　マイクロビット光っている LED を数字で表すには

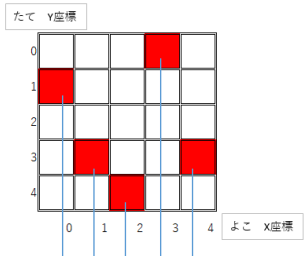
例

図14のように、**「３０４３１」**がスロットの出目だったとします。

よこの位置（０）、たての位置（１）＝１×【１】＝　　**１**
よこの位置（１）、たての位置（３）＝３×【１０】＝　　**３０**
よこの位置（２）、たての位置（４）＝４×【１００】＝　　**４００**
よこの位置（３）、たての位置（０）＝０×【１０００】＝　　**０**０００
よこの位置（４）、たての位置（３）＝３×【１００００】＝ **３００００**

合計　　３０４３１

です。【】の部分を見てもらうと、かけ算でけた上げした位の数を作り出しています。



３×１００００＝**３００００**
０　×　１０００＝　**０**０００
４　×　１００＝　**４００**
３　×　１０＝　**３０**
１　×　１＝　**１**
全部足すと・・・　**３０４３１**

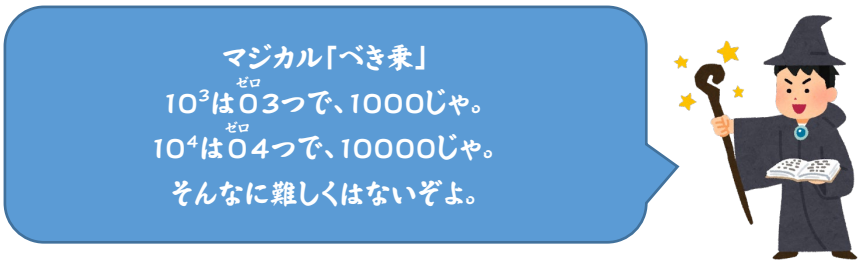
よこの X 座標と、数字が反対なので、わかりにくさが
倍増してますが、発展レベルなのでがんばって！

ここからは中学校で習う、**数学マジカル**をお伝えします。

数学マジカルとは、
今は知らなくても良い。
呪文のようなものだと思えばよい。
ただ、ちょっとは説明するかのう。
ふぉ、ふぉ

10^0＝　　＝１（これはルール）
10^1＝１０が一個　　＝１０
10^2＝１０×１０　　＝１００
10^3＝１０×１０×１０　　＝１０００
10^4＝１０×１０×１０×１０＝１００００

数学では、例えば**10^3を「10の3乗（さんじょう）」**と呼びます。これは、「10を3回かける」つまり「10×10×10」という意味です。こういった「同じ数を右上に小さく書いた回数だけかける」ことを**「べき乗（じょう）」**または**「累乗（るいじょう）」**といいます。数学マジカルですので、中学校へ行くまでは、呪文のようなものだと思ってください。



合計のための変数をつくる

それぞれのたて列で、選ばれた数字を10のべき乗で求めて、それらを足す必要があります。そのために命令のグループ 

で「変数を追加する」をクリックし、 

をつくります。

図15 がべき乗を使った計算式となります。いきなりパニックですね。何が書いてあるのか、ぱっと見は分かりませんが、ていねいに見ていけばわかると思います。

図 15　スロットの出目を数字に５ケタの数字に変換する計算式

まず、中心となる計算式を見てみましょう。
例えば　<u>たてが2、よこが0</u>で、数字としては　<u>2となる</u>ときと、
<u>たてが1、よこが3</u>で、数字としては<u>1000になる</u>ときを考えてみましょう。

図15 のうち、下のプログラムだけ抜き取って、考えると分かりやすいです。

2×10^0
$= 2 \times 1 = \underline{2}$
1×10^3
$= 1 \times 1000 = \underline{1000}$

となります。これをよこ(0)～よこ(4)まで全て足し算するので、それをプログラムにすると次のようになります。

これは、**「合計の数は、計算したものを合計の数に足していってね」**という命令です。

ここで、先ほどの当たり判定の考え方である、
「もし、合計の数を11111で割って、余りがなかったら当たりです」という命令を追加します。

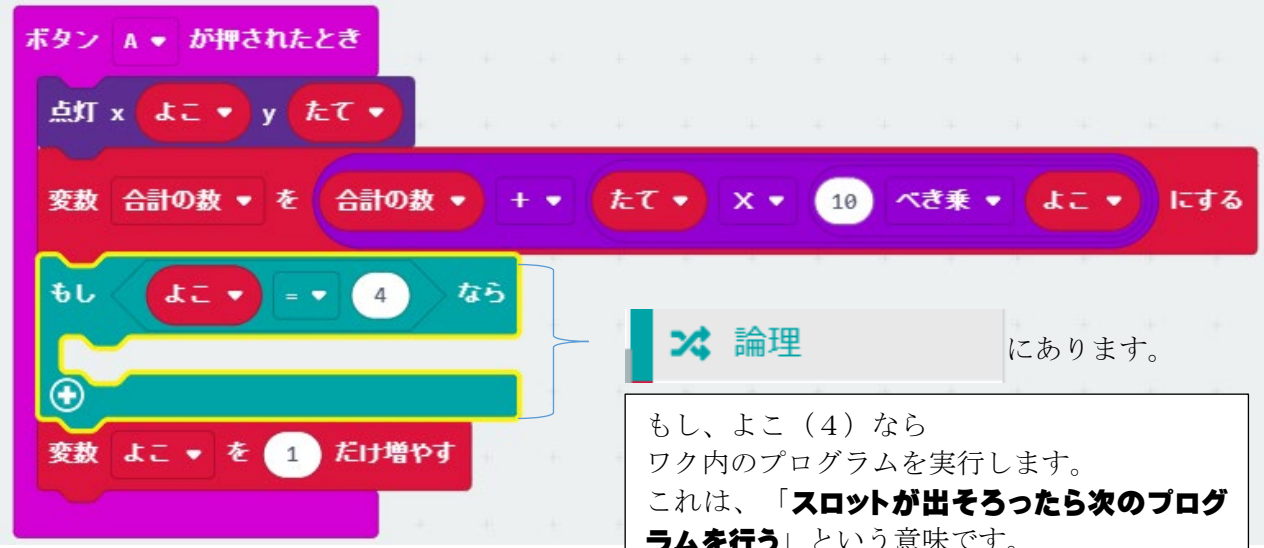


図 16　当たり判定のプログラムを入れる場所

図16 を見てください。**「もしよこ（4）なら～する」**のブロックをつくります。つまり**「全部回した段階で、当たり判定をする」**ことにしています。
続いて、**「もし、11111で割って、余り0なら」**というプログラム部分です。図 17 を見てみましょう。

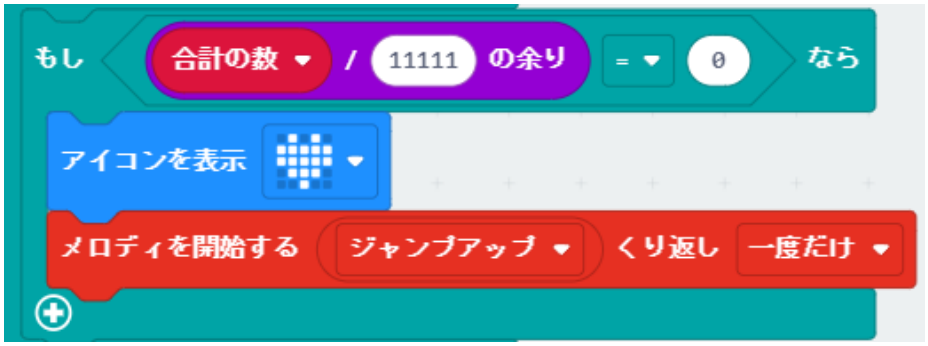


図 17　当たり判定して、当たった時のプログラム

「もし、あたりなら、♥を表示し、音を鳴らす」という設定を加えました。これで完成です。
ただし、少しバグがあります。

・ハートが表示された後も、スロットが回転している。
・1回目のスロット以降、当たらない。

プログラムを直したものが図18 になります。

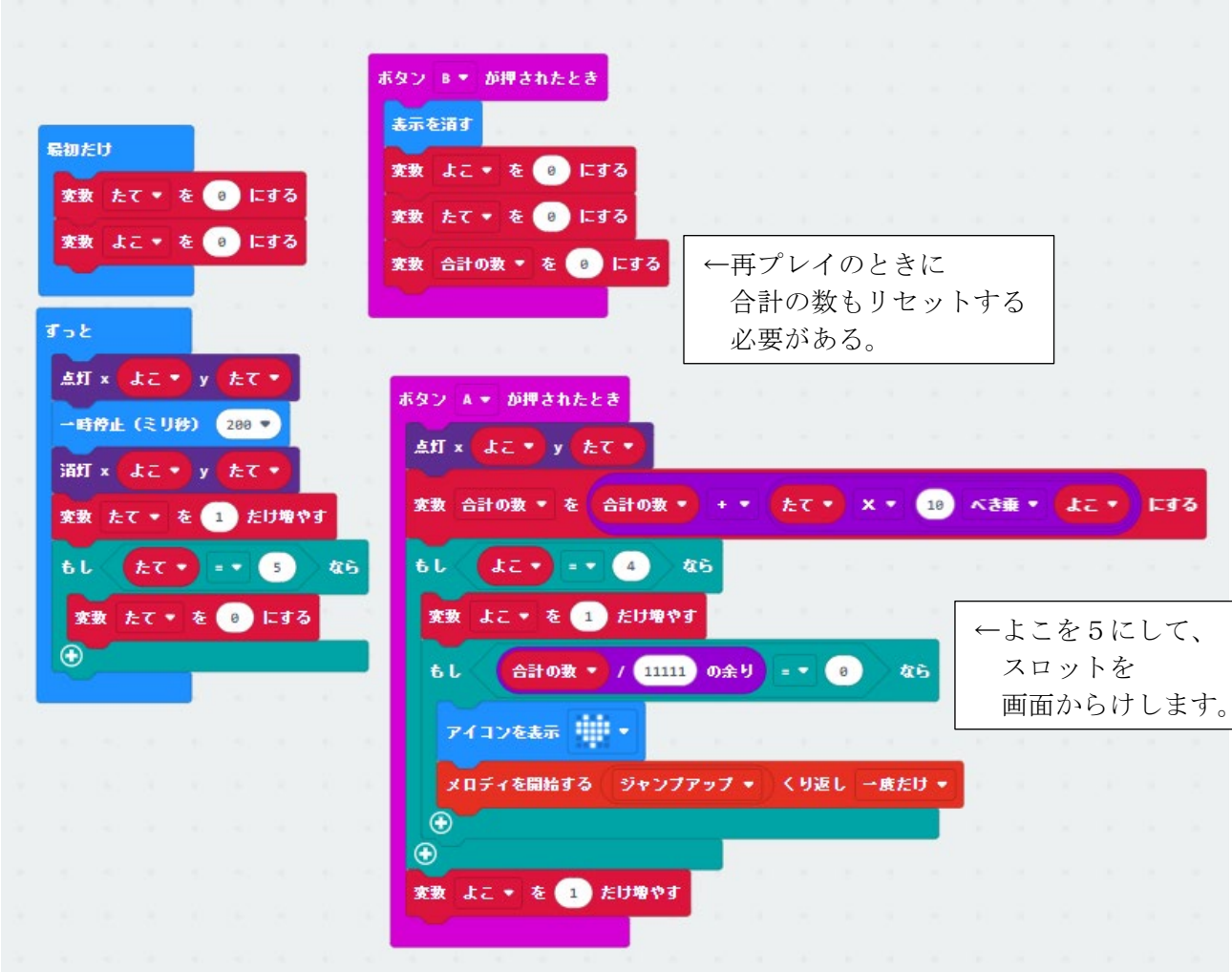


図 18　スロットマシンゲーム＜当たり判定ありバージョン＞

簡単に改良を加えるとすると、「スイッチを押したときに音がなる」とか、「ハズレだったときに、音がなる」など効果音を加えることがあげられます。ワクワクする仕組みを追加してみましょう。

おまけ

マイクロビットは、Bluetooth（ブルートゥース）を使った無線機能も付いています。プログラムをほんの少し改造して、友達とどちらが速くそろえられるか、勝負してみましょう。

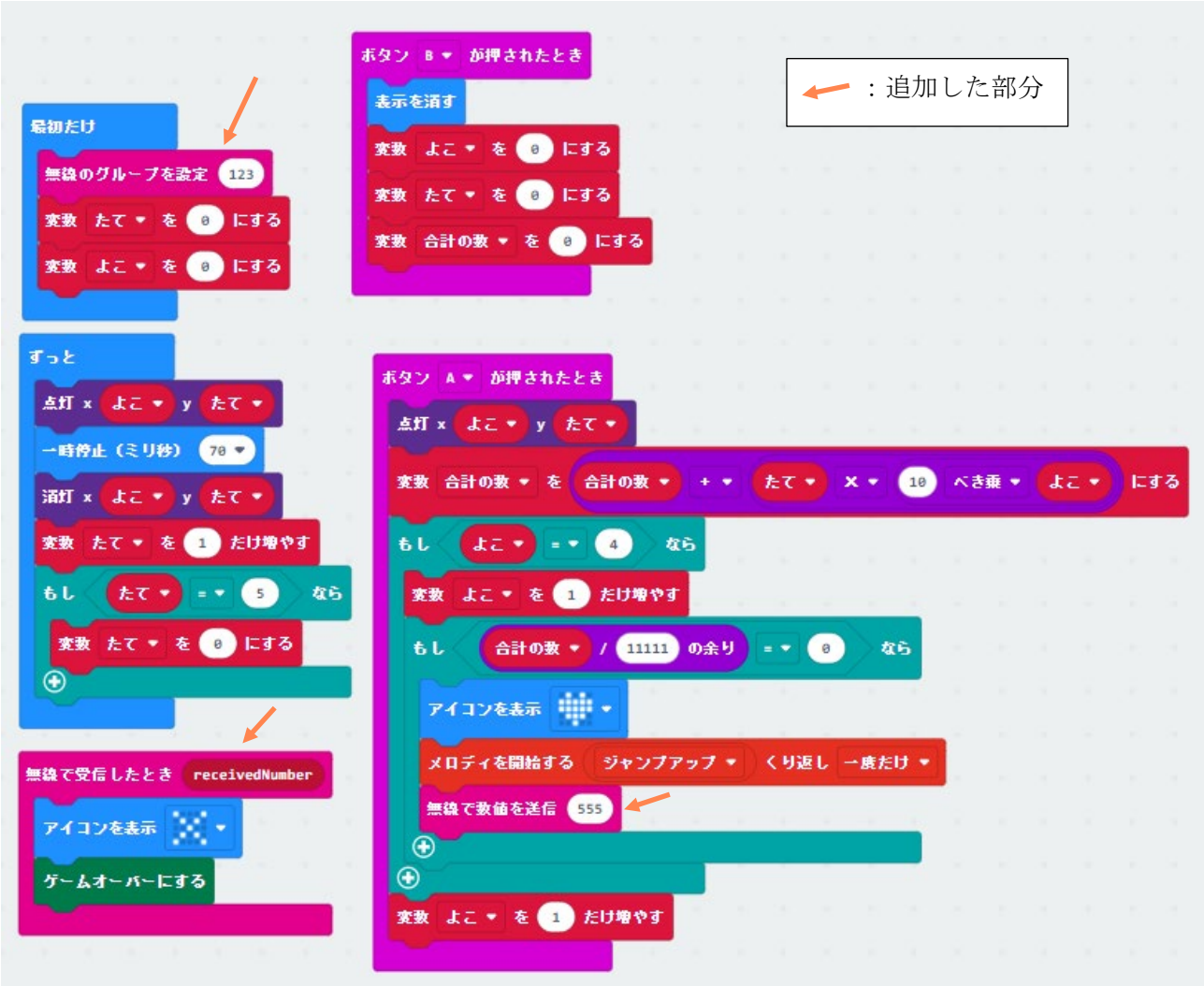


図 19　２台で勝負できるスロットマシンゲーム（おそらく 10 台でも可）

プログラムを見やすくするため、バグの処理が足りない部分があります。「もっとこうしたい」という変更点を見付けたら、ぜひ修正してみてくださいね！