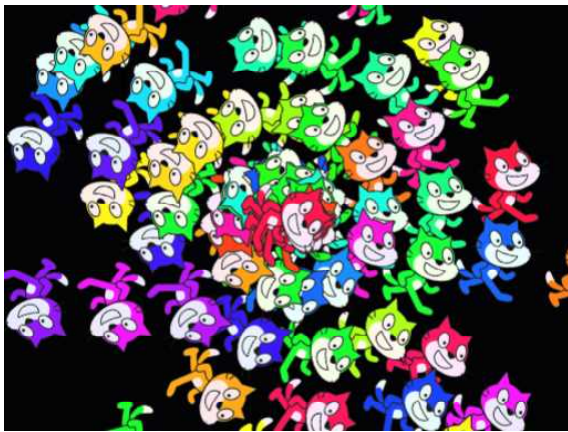


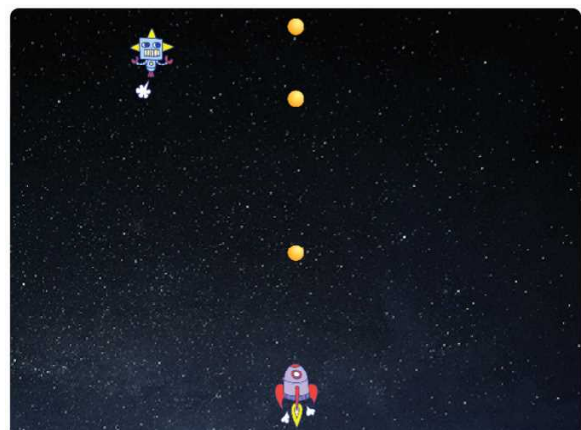
親子プログラミング入門^{に ゆ う も ん}

[2]



ネコ分身の術

シューティング・ゲーム

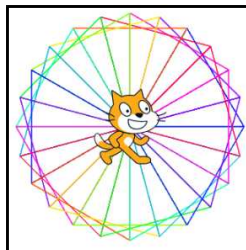


はじめに

■復習

「親子プログラミング入門」の実習内容は、

- ・Scratch の起動と基本的操作方法
- ・プログラム作成の基本
- ・ネコを歩かせる
- ・拡張機能「ペン」を使うための準備作業
- ・ネコを歩かせ、回転させ三角形を描く
- ・繰り返し処理で三角形を描画させる
- ・スクラッチの座標系は、横± 240 縦± 180
- ・色を変えながら軌跡を描き、三角形を 1 個描くごとに 15 度向きを変えてスピログラフを描画させる



でした。

■今回の実習内容とポイント

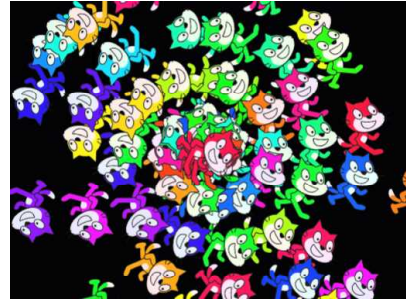
入門講座では「条件判断」「変数」と、「クローン」「メッセージ送信」という大きな要素が残っています、と言う所で終わりましたが、今回はこれらを勉強していきます。

具体的には、表紙に掲げた 2 つのプログラムを作りながら実習します。

- 1 ネコ分身の術
 - ・ネコを回転させて「クローン」を渦巻き状に拡散させる映像を作ります。
 - ・BGM を入れます。
- 2 シューティング・ゲーム
 - ・「条件判断」を使い左右に動く自機（スペースシップ）を作成
 - ・自動的に左右移動する敵キャラ（ロボット）を作成
 - ・「条件判断」と「クローン」で弾を発射し、上昇させます。
 - ・弾がロボットに当たったかを「条件判断」し、当たったらダメージ表示します。
 - ・発射音など効果を追加します。

お題1 「ネコ分身の術」

このお題は、前回の復習を兼ねつつ、新たな「クローン」技術を学びます。



■コーディング内容

- ・ ネコが画面中心で色を変えながら回転
- ・ 回転するたびに自身の「クローン」を生成
- ・ 「クローン」されたネコは一定回数歩き続ける
- ・ 一定回数歩くと「クローン消滅」

■ネコを回転させる



「初期設定」をします。

「イベント」群から「旗が押されたとき」を置き、その下に「動き」群から「X座標を0, Y座標を0にする」

同じく「動き」群から「90度に向ける」

「見た目」群から「大きさを70%にする」を結合します。



「制御」群から「ずっと」を置き、「動き」群から「30度回す」を内包させ、先のブロックに結合します。

実行すると画面中央でネコが回転をします。

■猫の色を変える



「見た目」群から「色の効果を25ずつ変える」を「30度回す」の下に入れます。

■自分自身の「クローン（複製）」を作る

「クローン」とは「自分と同じ機能を持ったスプライトを作る」機能と考えてください。いわば「分身の術」です。



「制御」群から「(自分自身の) クローンを作る」を挿入します。

このままでは何も起きません。

■クローンされた複製ネコの動作を指定する



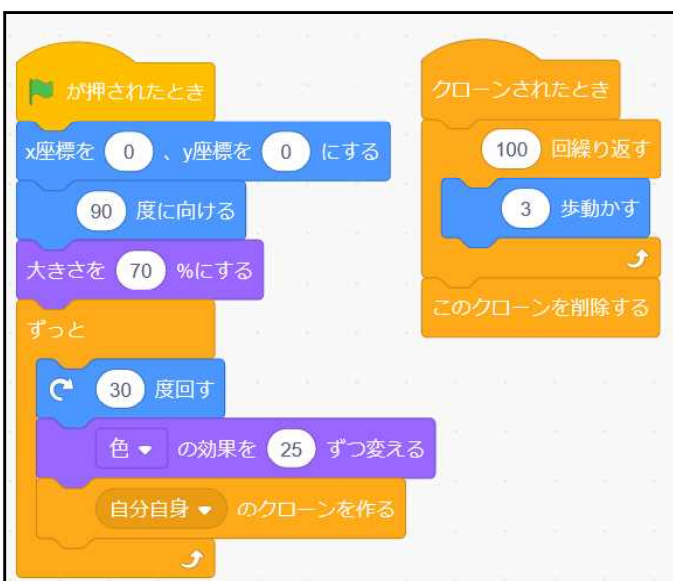
「制御」群から「クローンされたとき」を **画面の空きエリアへ単独で配置** します。

その下へ次のブロックを結合します。

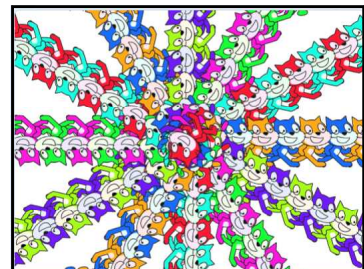
「制御」群から「**100 回繰り返す**」

「動き」群から「**3 歩動かす**」

「制御」群から「**このクローンを削除する**」



単純なネコの拡散プログラムが完成です。



結局、「クローンした側」の属性、色、形、向きなどはそのまま「クローンされた側へ」複製されます。

なので、「クローンされた側」で各ネコに歩かせると、放射状に画面の中を歩いて行くことになるのです。

いつまでも消えないと困るので 300 歩動いたら「消滅」してもらうために、「このクローンを削除する」が入れてあります。

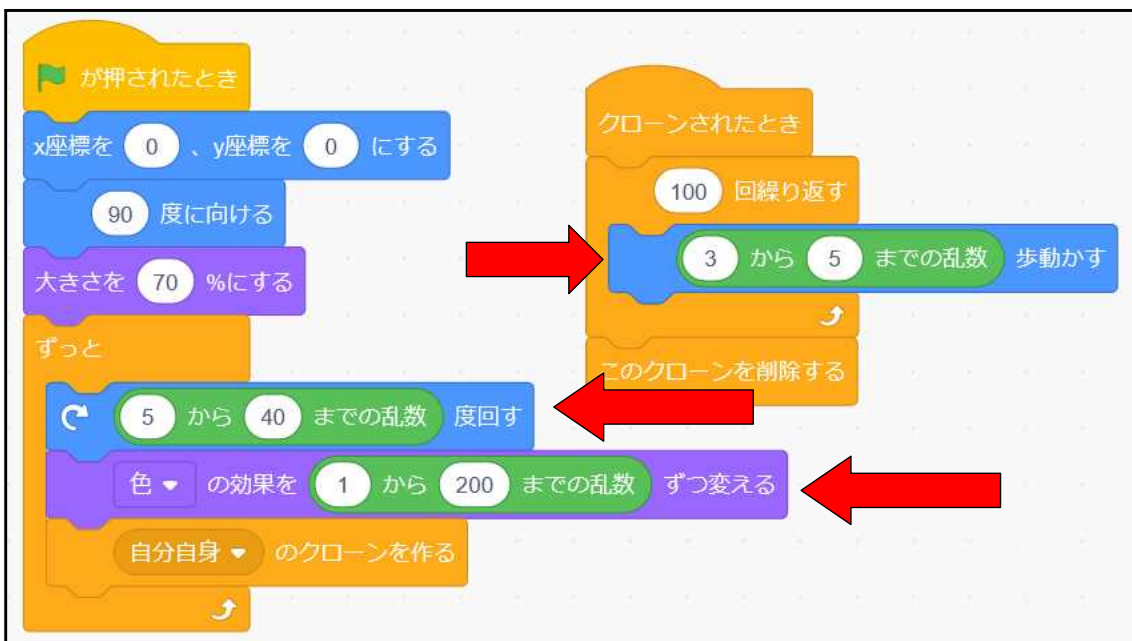
■クローンとは

Scratch が内部に持つ「オブジェクト」を丸ごと複製し、自分のプログラムとして利用したり、新たな機能を追加することが出来る機能です。

最近の言語の大多数はこの技術を利用することで、複雑な中身を知ることなく利用して、プログラミングの生産性を大幅に上げることができるようになりました。

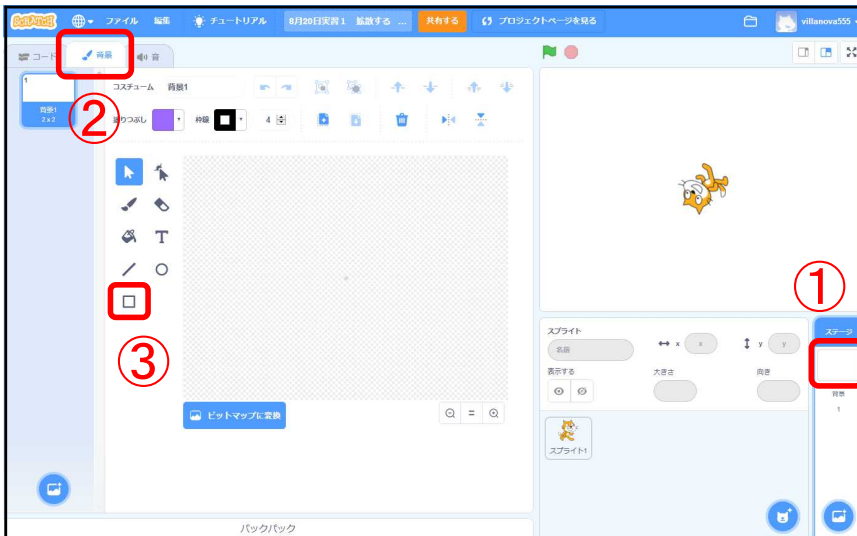
■拡散の仕方を変化させる

このままでは画一的でつまらないので、「乱数」で動作をランダムに変えてみます。「演算」群から「() から () までの乱数」を適宜ブロックに埋め込みます。



■背景を黒くする

色が見栄が良いように背景を黒くします。



①画面右下の「背景」をクリックします。

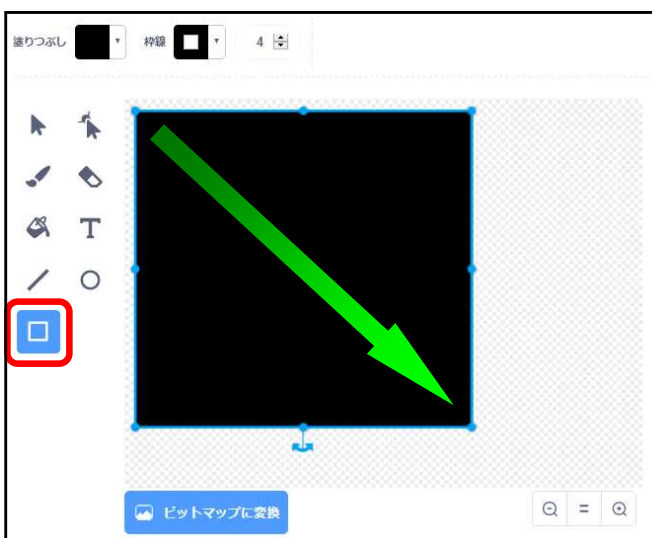
②「背景」タブをクリックします。

③描画ツールの「四角形」をクリックします。

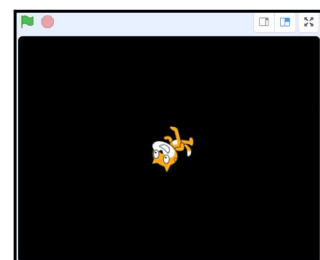


①「塗りつぶし」をクリックし、

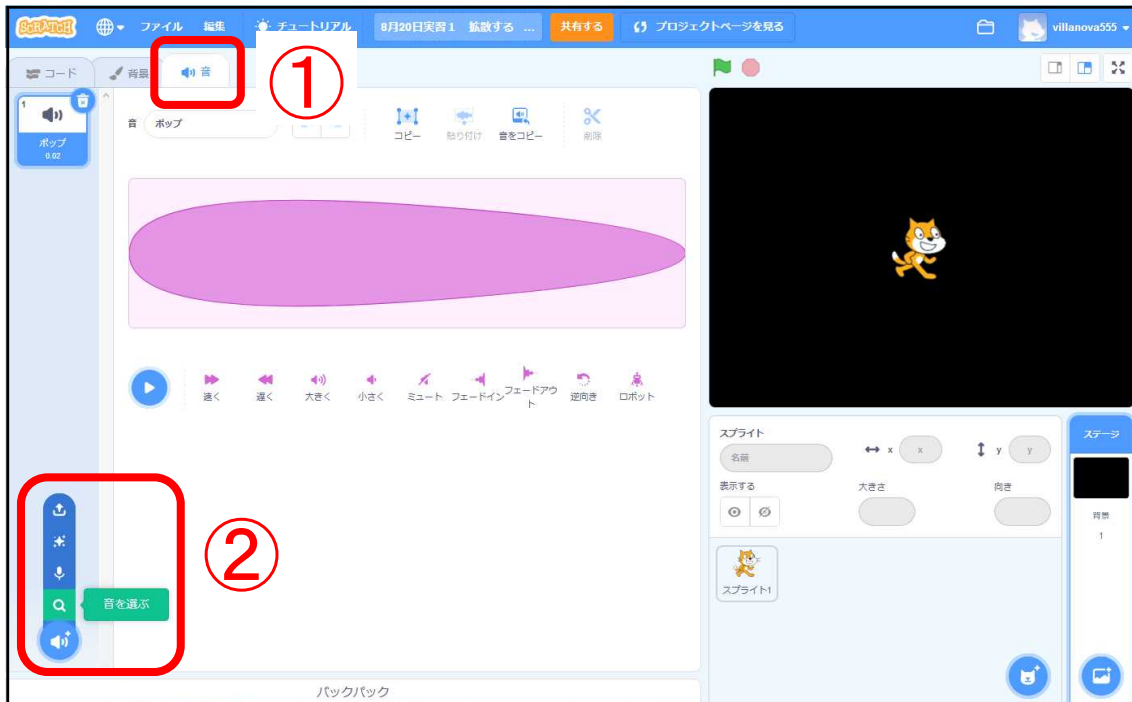
②「明るさ」スライダーをドラッグして「0」にします。



「四角形」が選択されていることを確認して、編集画面上で黒色の四角形を作成し、画面一杯に広げます。背景が黒くなりました。

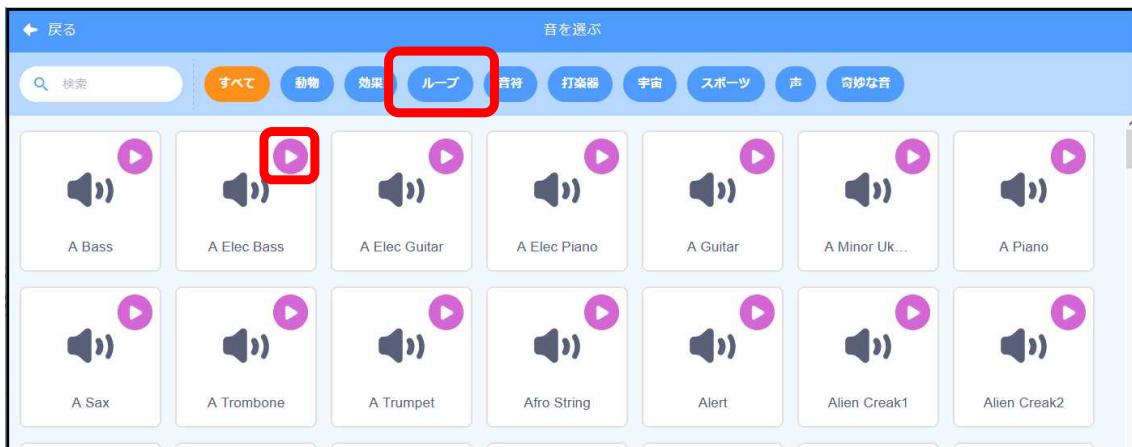


■ BGM を付けてみる



①背景が選択されている状態から、「音」タブを選択します。

②左下の「音を選ぶ」から虫眼鏡マークの「音を選ぶ」をクリックします。



紫色の△マークにマウスを乗せると音楽を再生させることができます。ジャンル「ループ」を選び、好きな楽曲を追加してください。

新たな音楽が「背景」に追加されました。

「コード」タブをクリックします。



背景画面に音楽をずっと流すようにプログラムを作成します。

制御から「旗が押されたら」その下に「ずっと」中に「音楽」から「終わるまで **XXXX** の音を鳴らす」を挿入します。

お疲れ様でした。お題1の終了です。

■ サンプル

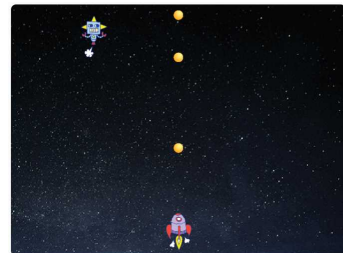
<https://scratch.mit.edu/projects/336424947/>

お題2 「シューティング・ゲーム」

このお題は、

- ・ゲーム作成の基本であるキー入力による自機の操作
- ・「クローン」を使った弾の発射
- ・当たり判定

を実習します。



■ コーディング内容

- ・新スプライトの導入（宇宙船、ロボット、弾丸）
- ・左右矢印キーによる宇宙船の左右移動
- ・画面上を左右移動するロボット
- ・スペースキーによる弾丸の発射（クローン生成）
- ・弾丸とロボットの当たり判定

■ 新規画面の作成

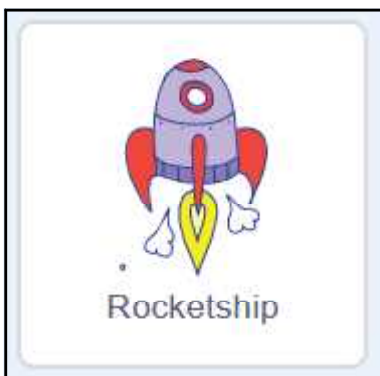


「ファイルメニュー」から「新規」で新たなプロジェクトを作成
スプライト1（ネコ）の右上にあるゴミ箱ボタンで削除します。

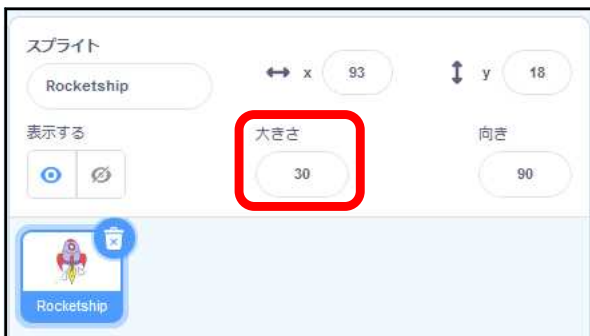
■ 宇宙船の配置



「スプライトを選ぶ」をクリックします。



一覧から「RocketShip」をダブルクリックで選択します。



ロケットがスプライトエリアに表示されるので、「大きさ」を「30」に書き換えます。



「制御」群から「旗が押されたとき」を置いて、その下に「動き」群から「X 座標を (0)、Y 座標を (-145) にする」を結合し、数値をそれぞれ「0,-145」と書き換えます。

■ 宇宙船の左右移動

ユーザーがキーボードの左右矢印キーを押すと、宇宙船が左右に移動するプログラムを書いています。



「制御」群から「ずっと」を結合します。この中へ、キー入力があった場合のロケットの反応を書いています。

まずは、右向き矢印キーが押された場合の処理を書きます。

ロケットが右へ動くのは、ロケットの X 座標を加算することで実現できます。

同様に左に動くには、X 座標を - 10 します。

(画面ハードコピーは右移動部分のみを表示させています。)



「制御」群から「もし > なら」をセットし、その「< >」の中に、「調べる」群から「(スペース) キーが押された」を埋め込みます。



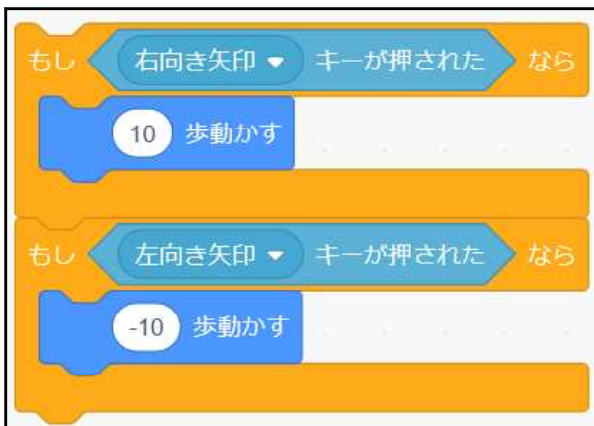
「▽」をクリックし、「右向き矢印」を選択し直します。



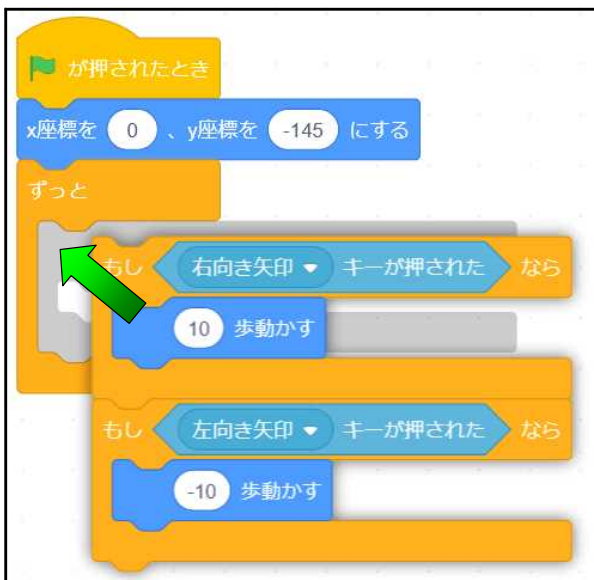
「動き」群から「10 歩動かす」を「もしブロック」の中に入れます。



「もし」を右クリックし、表示されたメニューから「複製」を選ぶと、同じブロックが作成されます。



複製されたブロックをこの下に結合します。
複製されたブロックは同様の手順を適用し、「左向き矢印」が押されたら「-10 歩動かす」に修正します。

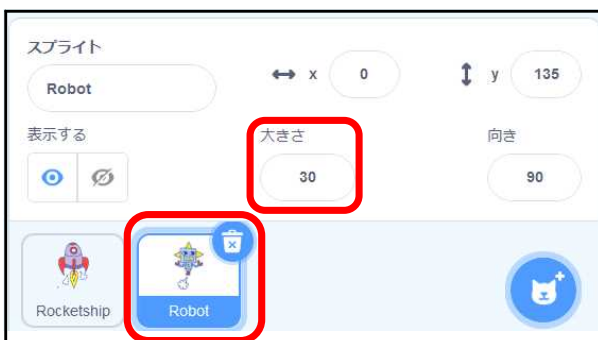


できあがった「もし」文のブロックを、「ずっと」の中へ挿入します。



「ロケット左右移動」の完成です。

■ ロボットの配置と移動



宇宙船と同じ手順で、ロボットのスプライトを追加します。(ロケットの隣にあります)

同様に大きさを「30」に設定します。



「旗が押されたとき」の下に X,Y 座標を設定するブロックを結合し、X,Y の値を「0, 135」に変更します。

「動き」群から「回転方法を[左右のみ]にする」を結合します。(おなじないと思ってください。



ロボットは、画面をずっと左右移動するようにプログラムします。

「制御」群から「ずっと」を結合し、中に、「動き」群から「10 歩動かす」と、「もし端に着いたら、跳ね返る」を入れます。



少しロボットの動きが速いので、間に「制御」群から「1 秒待つ」を入れて、値を「0.1 秒待つ」に変えてください。

これでロボットの動きは完成です。

■ 弾丸発射の機能



スペースキーを押すとロケットから弾丸が発射され上へ向かって飛んで行く仕組みを作ります。

- ・弾丸は初めは隠しておき
- ・スペースキーが押されたら、ロケットの位置へ姿を現し
- ・一定時間ごとに Y 座標を加算することで、飛行しているように見せます。

弾丸は同時複数存在（連射）できるように「クローン」機能を使って生成します。

■ 弾丸の配置



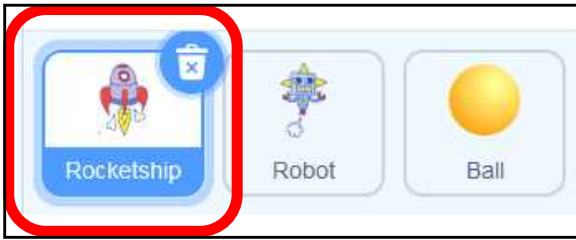
他のスプライトと同様に「Ball」スプライトを追加し、大きさを「30」にして、画面の適当な位置に配置します。

■ 弾丸の初期設定



「イベント」群の「旗が押された時」の下に「見た目」群から「隠す」を結合します。

■宇宙船のSpriteに弾丸発射のきっかけ（キック）を記述する。



宇宙船「RocketShip」のSpriteをクリックします。



「もし左向き矢印キーが押された」を右クリックして「複製」を作ります。



条件を「スペースキーが押された」に変更し、中の「-10歩動かす」を削除します。



「制御」群から「(自分自身)のクローンを作る」を挿入します。



「▽」をクリックし「Ball」を選択します。これで、弾丸のクローンが生成されます。



このままでは連射され過ぎるので、発射間の時間をとるため、「制御」群から「1秒待つ」を挿入し、時間を「0.2秒」に変更します。

※参考 連射制御のためこの時間をあまり長くすると左右移動がカクつきます。連射禁止で時間を長くする場合は、スペースキーが押された処理を独立させます。



「左向きキーが押された」処理ブロックの下側へ挿入します。

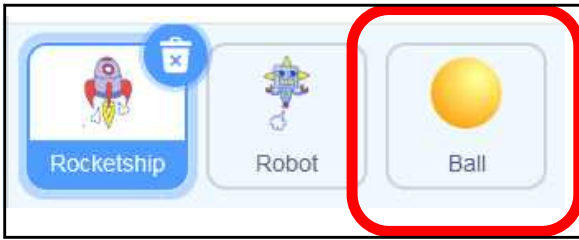


宇宙船のSpriteに係るプログラムはこれで変更完了です。

赤枠部分が追加になった部分です。

作成されたクローンの上昇については、Sprite「Ball」の「スクリプト」エリアへプログラムを組んでいきます。

■ クローンされた弾丸の上昇処理



スプライト「Ball」をクリックして、弾丸の上昇処理をプログラムしていきます。



「イベント」群から「クローンされたとき」を画面上の独立した場所へ配置します。



「クローンされたとき」に「動き」群から「(どこかの場所)へ行く」を結合し、

「▽」をクリックして「Rocketship」を選択します。

これでクローンされた時に出現する場所が指定されました。

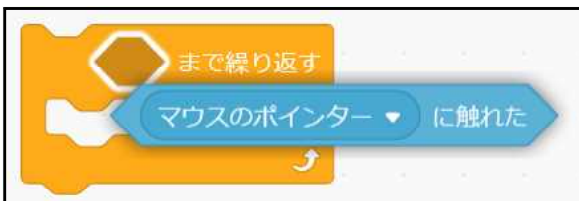
(正確にはロケットの中心部にあたります)



「見た目」群から「表示する」を結合します。

クローン元は「非表示」でしたが、これで見えるようになります。

続いて、弾丸を画面の上辺に達するまで上昇する処理を書いていきます。



「制御」群から「< >まで繰り返す」を置きます。

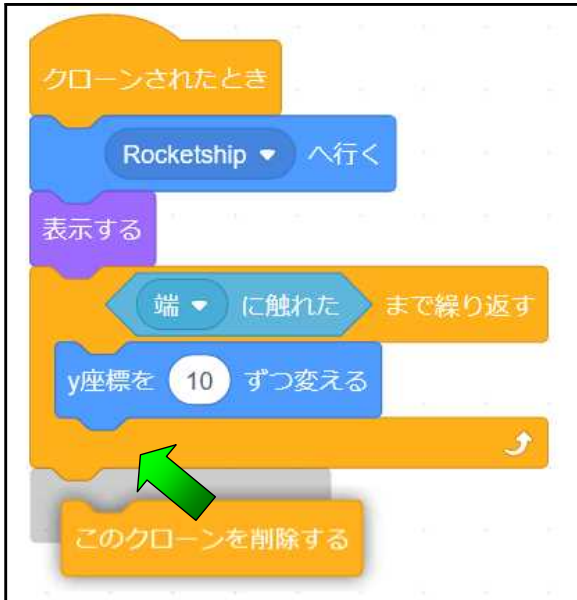
その条件に「調べる」群から「(マウスポインター)に触れた」を結合します。



「▽」をクリックして「端」を選択します。



「動き」群から「Y座標を 10 ずつ変える」を挿入します。
10 ずつ増えるということは、画面の上へ上がっていくことを意味します。

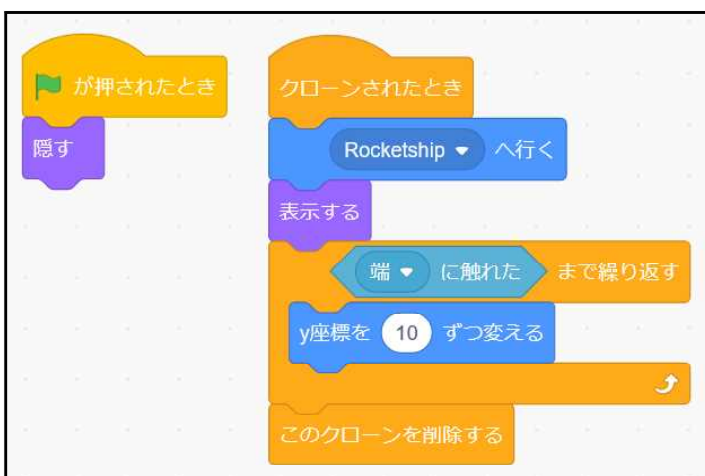


作成した「端に触れたまで繰り返す」を「表示する」の下に結合し、

最下段に、「制御」群から「このクローンを削除する」を結合します。

- 1 端に触れるまで上昇し、
- 2 触れたらこのブロックが終了
- 3 最後に自分自身を消去する。

という流れになります。



完成した「弾丸上昇ルーチン」です。

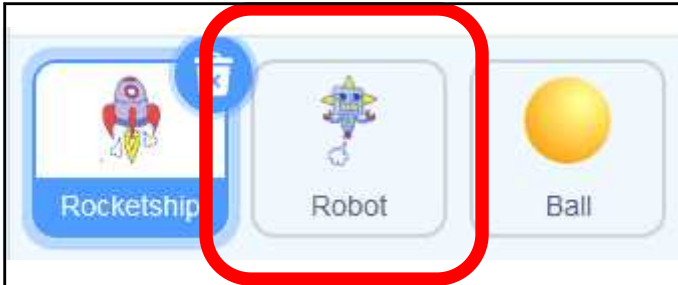
※お断り

実習内容を簡単にするため、ゲームの完成度とか見た目を度外視して作成しています。

■弾丸とロボットの当たり判定

弾がロボットに当たったら、ロボットの色を変える処理を書きます。

弾のスクリプト内でも当たり判定は可能ですが、ロボットを変化させるので、ロボットのスクリプト内にプログラムを書きます。



ロボットのSpriteをクリックします。



今までのプログラムの横に、「制御」群から「もしく > なら」を仮置きします。

(後で「ずっと」の最後に挿入します。)



<>の中に、「調べる」群から「(マウスポインター)に触れた」を挿入します。



「▽」をクリックして「Ball」を選択します。

「ロボットが弾丸に触れたら」という条件を書いたこととなります。



「もし Ball に触れたなら」の中に、「見た目」群から「色の効果を (25) ずつ変える」を挿入します。



作成したブロック群を、「もし端に着いたら、跳ね返る」の下に挿入します。



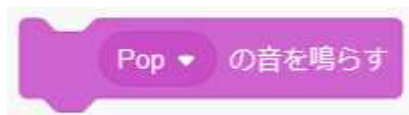
これで、最低限のコーディングが完成しました。

ゲームとしては物足りませんので、

- ・弾の発射音
- ・ロボット被弾時の効果音
- ・背景画像の挿入
- ・BGM

等を入れるとそれらしくなります。

音の挿入は「音」群を開き「XX の音を鳴らす」を使います。



時間があればチャレンジしてみてください。

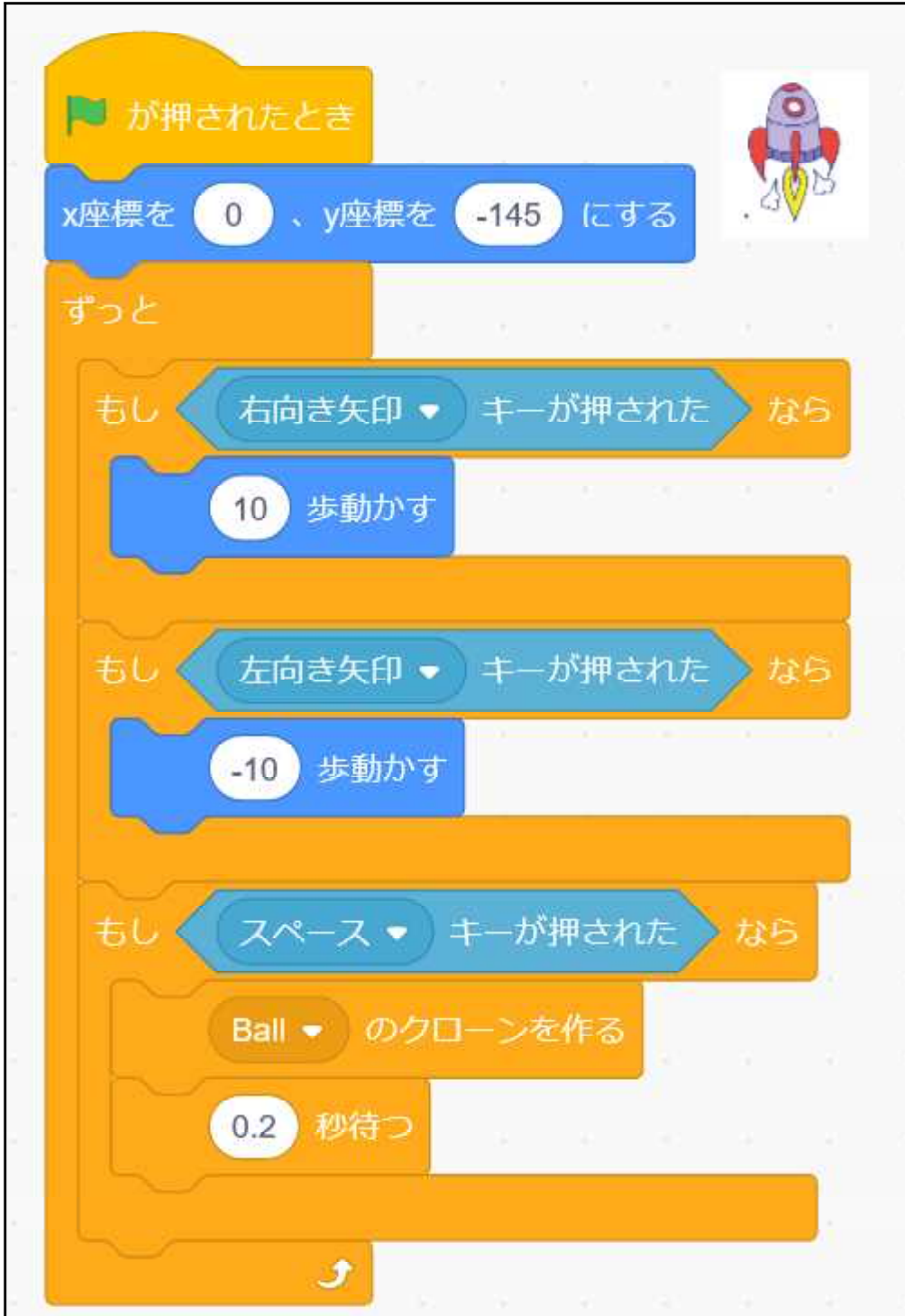
実習はここまでです。

お疲れ様でした。

■ サンプル

<https://scratch.mit.edu/projects/336426235/>

「宇宙船」「ロボット」「ボール」それぞれにプログラムを書いていきます。
書き込むオブジェクトを間違えないでください。



The image shows a Scratch script for a rocket ship. The script starts with a 'when green flag is clicked' event block. This is followed by a 'set x coordinate to 0, y coordinate to -145' block. A 'forever' loop block contains three 'if key pressed' blocks. The first 'if key pressed' block checks for the right arrow key; if pressed, it moves the rocket 10 steps to the right. The second 'if key pressed' block checks for the left arrow key; if pressed, it moves the rocket 10 steps to the left. The third 'if key pressed' block checks for the space key; if pressed, it clones the 'Ball' object and then waits for 0.2 seconds. The script ends with a 'repeat forever' loop arrow.

宇宙船


```

    旗が押されたとき
    x座標を 0、y座標を 135 にする
    回転方法を 左右のみ にする
    ずっと
    10 歩動かす
    0.1 秒待つ
    もし端に着いたら、跳ね返る
    もし Ball に触れた なら
    色 の効果を 25 ずつ変える
  
```

ロボット

```

    旗が押されたとき
    隠す
    クローンされたとき
    Rocketship へ行く
    表示する
    端 に触れた まで繰り返す
    y座標を 10 ずつ変える
    このクローンを削除する
  
```

ボール