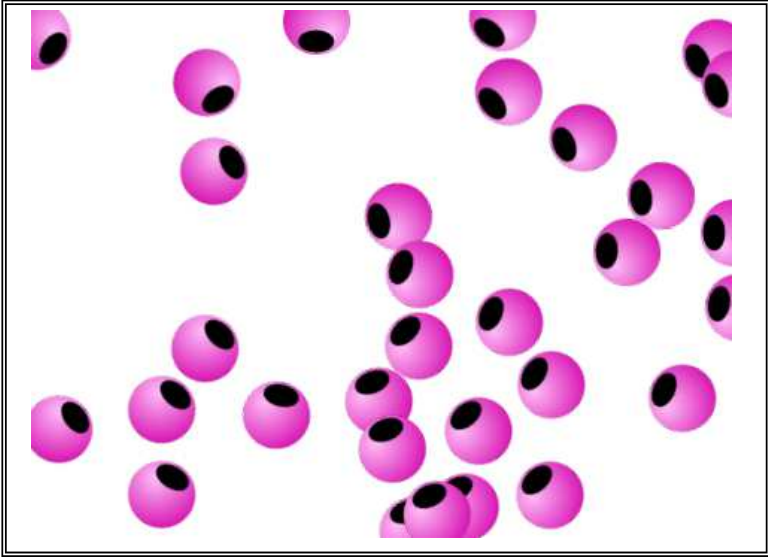


きも目玉



マウスのカーソルを動かすと目玉がギョロギョロうごくよ。
次の順番で作っていくよ。

1. 目玉のコスチュームをつくる。
2. 目玉のコードをつくる。
3. 改良・拡張してみよう。



Go.Ota CoderDojo 市川真間

本プログラムは、土肥先生(東京電機大学)のものを参考としています。

きも目玉[1]

1. 目玉のコスチュームをつくる。



Ballのスプライトに黒目を足して目玉をつくります

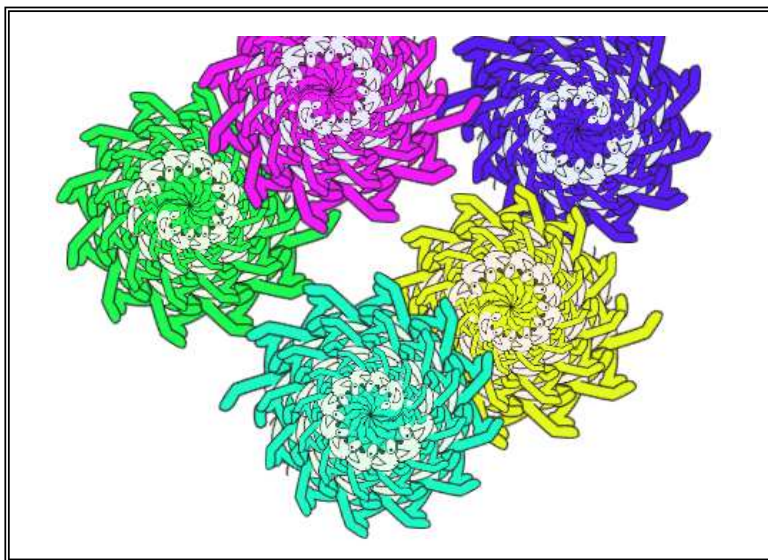
2. 目玉のコードをつくる。



3. 改良・拡張してみよう。

- 目玉からビームを出してみよう。
- 目玉の色や数を変えてみよう。
- いろいろなものをギョロギョロさせてみよう。

ぐるぐる



いろいろな色のねこが、ぐるぐる回ってきれいなもようを書くよ。
(マウスでカーソルを動かしてクリックします。)


次の順番で作っていくよ。

1. ネコのコードをつくる。
2. 改良・拡張してみよう。



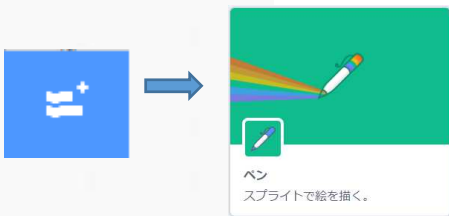
ぐるぐる[1]

1.ネコのコードをつくる。



The image shows a Scratch script for a cat drawing. The script starts with a 'when clicked' event block, followed by a 'set size to 30%' block, a 'clear' block, and a 'wait 1 second' block. A 'click and drag' loop contains a 'go to mouse pointer' block, an 'if mouse clicked' block, and a 'set size to 100%' block. Inside the 'if' block, there is a 'set color effect to random number from 0 to 200' block, a 'repeat 30 times' loop, and a 'stamp' block. The 'repeat' loop contains a 'repeat 30 times' loop and a 'set size to -2' block. The 'stamp' block is placed at the end of the 'if' block.

左下のブロックの追加をクリックして [ペン] のブロックを追加します。



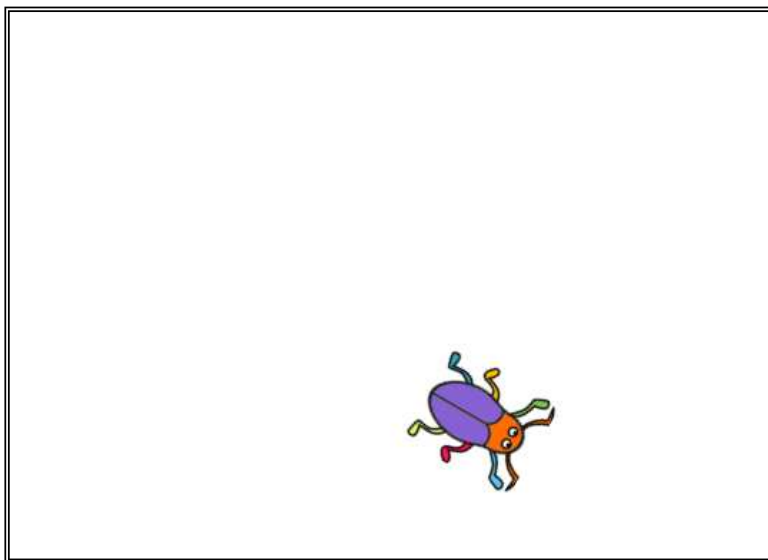
The diagram shows a blue 'stamp' block with a plus sign icon. An arrow points to a 'pen' block, which is a green block with a pencil icon. Below the pen block, there is a small preview of a green square with a pencil drawing a line, and the text 'ペン スプライトで絵を描く。' (Pen: Draw a picture with the pen sprite).

3. 改良・拡張してみよう。

○ ぐるぐるのしかたを変えて、きれいな図をえがこう。

○ いろいろなものを回してみよう。

ゴキブリたたき



みえかくれするゴキブリをクリックしてたいじしよう。

次の順番で作っていくよ。

1. ゴキブリのコードをつくる。
2. 改良・拡張してみよう。



ゴキブリたたき[1]

1. ゴキブリのコードをつくる。

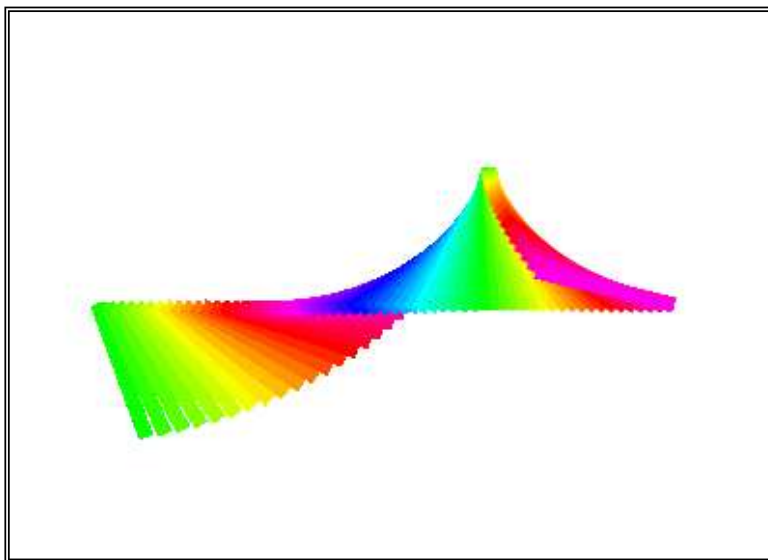


3. 改良・拡張してみよう。

○ ゴキブリの出方や動き方を変えてみよう。

○ 時間制限や得点をつけてみよう。

忍者の里 Scratch カード 04 難易度 ★☆☆☆☆(かんたん)
カラーウェーブ



七色の波がうごくよ。

次の順番で作っていくよ。

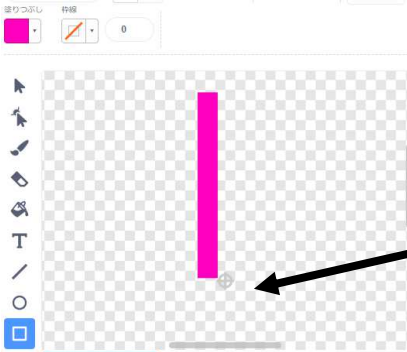
1. 新しく波のもとの四角のSpriteをつくる。
2. 波のコードをつくる。
3. 改良・拡張してみよう。



Go.Ota CoderDojo 市川真間

カラーウェーブ[1]

1. 新しくスプライトを作る。



色のついた長いぼうを作ります。

中心が端にくるようにつくります

2. 波のコードをつくる。



3. 改良・拡張してみよう。

○ もとのスプライトの色や形をかえてみよう。

○ 波の動きをかえてみよう。

おにごっこ



いろいろな物のかげに隠れて逃げ回るネコをクリックして捕まえてみよう。

次の順番で作っていくよ。

1. ネコのコードをつくる。
2. いろいろな物を配置する。
3. 改良・拡張してみよう。



Go.Ota CoderDojo 市川真間

本プログラムは、土肥先生(東京電機大学)のものをを参考にしています。

おにごっこ[1]

1. ネコのコードをつくる。



2. 2. いろいろな物を配置する。



ネコがかくれられるように、いろいろなスプライトを追加してみよう。

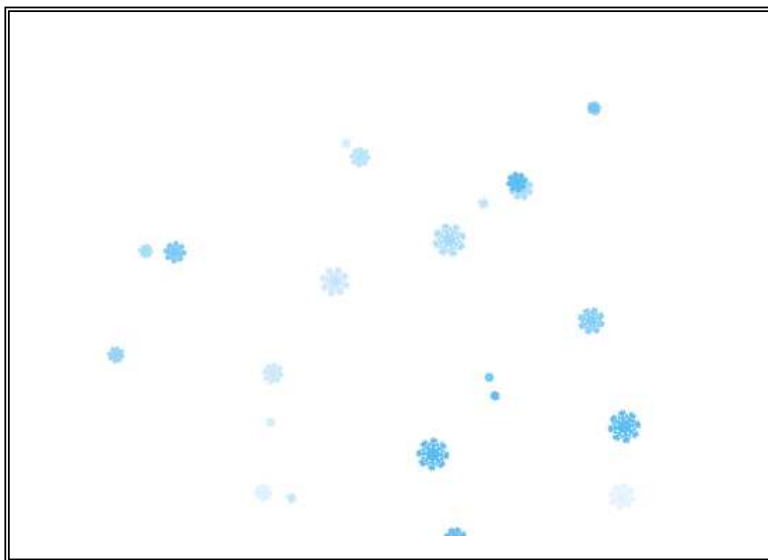
3. 改良・拡張してみよう。

○ ネコの動きをかえてみよう。

○ ネコが見つかった時のアクションを変えてみよう。

○ 他のスプライトの追加や配置を変えてみよう。

忍者の里 Scratch カード 06 難易度 ★★★☆☆(ふつう)
雪



きれいな雪がふる、けしきが見えます。

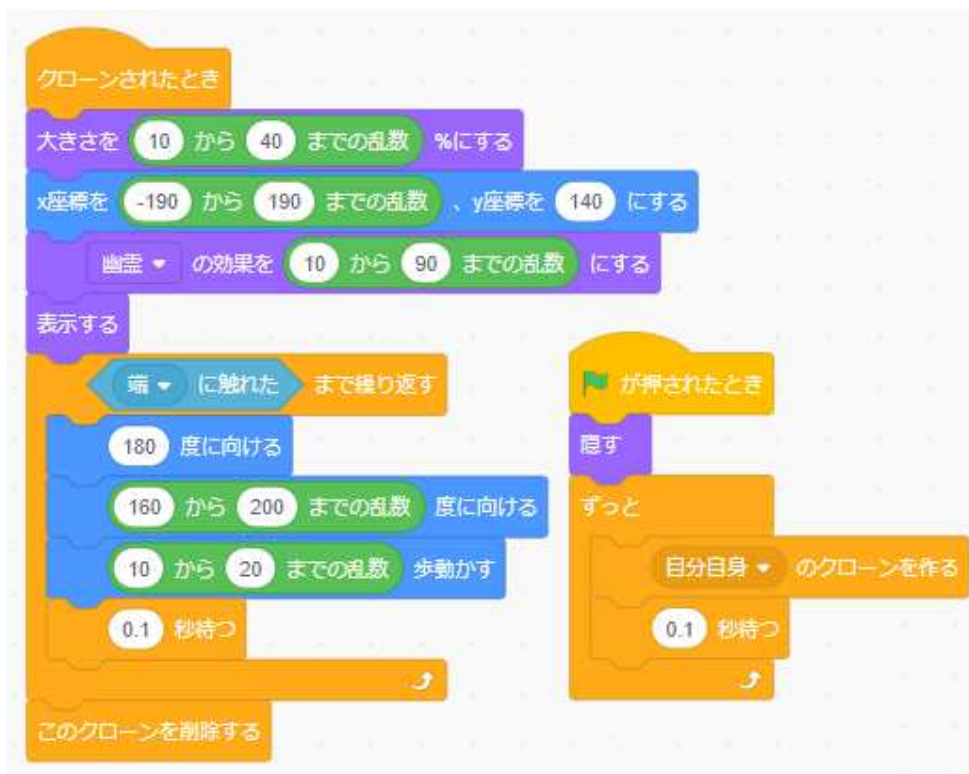
次の順番で作っていくよ。

1. 雪のコードをつくる。
2. 改良・拡張してみよう。



雪[1]

1. 雪のコードをつくる。



3. 改良・拡張してみよう。

○ 音や背景をつけてみよう。

○ いろいろな物をふらせてみよう。

昔話 1



Scratch でデジタル絵本を作ってみよう。

1,2,3 と数字を押すと話が進んでいくよ。

次の順番で作っていくよ。

1. 背景とスプライトを準備しよう。
2. 旗を押したときのコードを作ろう。
3. 1 を押したときのコードを作ろう。
4. 2 を押したときのコードを作ろう。
5. 3 を押したときのコードを作ろう。
6. 改良・拡張してみよう。

昔話 1[1]

1. 背景とスプライトを準備しよう。

○ 背景を追加しよう



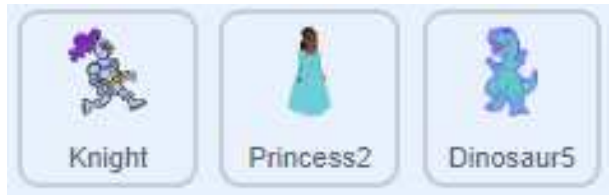
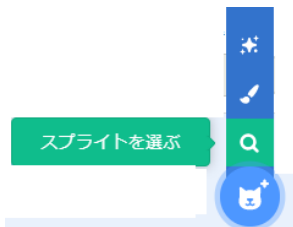
Woods



Castle 1

好きな背景を選んでみよう

○ スプライトを追加しよう。



旗をクリックすると音楽が流れるよ

昔話 1 [2]

2. 旗を押したときのコードを作ろう。

○背景のコード作る。(3.を押したときのコードも)

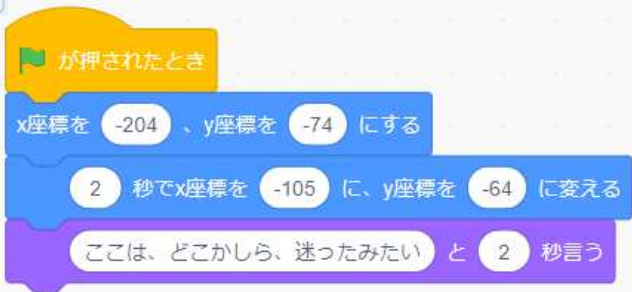


```
当緑旗が押されたとき
背景を Woods にする
```



```
3 キーが押されたとき
背景を Castle 1 にする
```

○ Princess2 のコードを作る。



```
当緑旗が押されたとき
x座標を -204 、y座標を -74 にする
2 秒でx座標を -105 に、y座標を -64 に変える
ここは、どこかしら、迷ったみたい と 2 秒言う
```

○ Dinosaur5 のコードを作る。



```
当緑旗が押されたとき
隠す
```


○ Knight のコードを作る。




```
当緑旗が押されたとき
隠す
```

昔話 1 [3]

3. 1 を押したときのコードを作ろう。

-  のコードを作る。




-  のコードを作る。




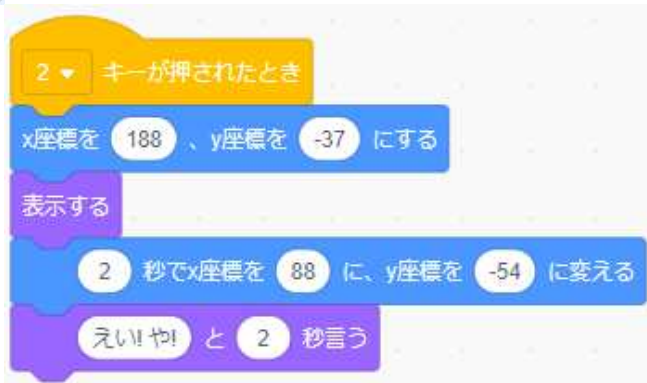
昔話 1 [4]

4. 2 を押したときのコードを作ろう。

-  のコードを作る。





-
-  のコードを作る。




昔話 1 [5]

5.3 押したときのコードを作ろう。

-  のコードを作る。



```
3 ▼ キーが押されたとき
x座標を -216 、y座標を -65 にする
3 秒でx座標を -61 に、y座標を -59 に変える
めでたし、めでたし... という
```

-  のコードを作る。



```
3 ▼ キーが押されたとき
x座標を 207 、y座標を -53 にする
3 秒でx座標を -6 に、y座標を -59 に変える
```

6. 改良・拡張してみよう。

- いろいろな動きをさせてみよう。
- いろいろなスプライトを登場させよう。
- いろいろな物語を作ってみよう。

ダンス



音楽に合わせてダンスをおどるよ。

次の順番で作っていくよ。

1. ダンサーのコードをつくる。
2. 背景を追加する。
3. 改良・拡張してみよう。



ダンス[1]

1. ダンサーのコードをつくる。

○Cassy Dance のスプライトを追加する



Cassy Dance
にはダンス用
の音楽が入っ
ています。

○ コードを作る



ヒント:

0.3 秒待つの値を変えたりしたらどうなるかみてみよう。

2. 背景をついかする。

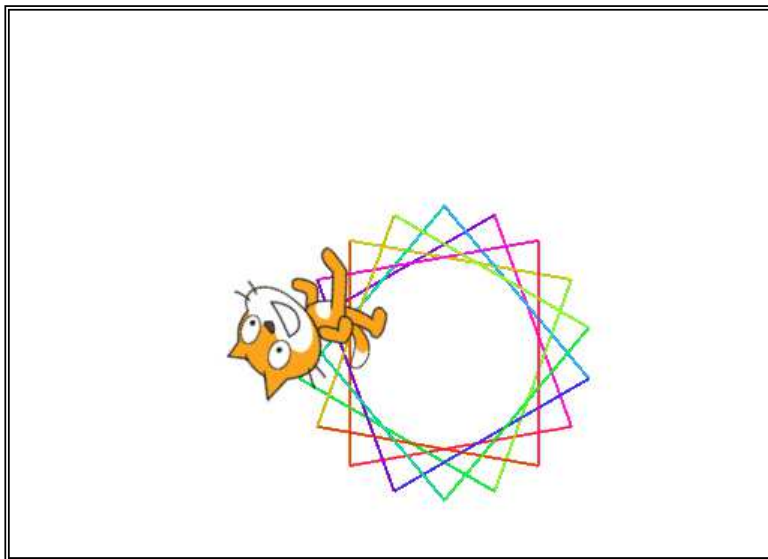
○ てきとうな背景をついかしてみよう。

3. 改良・拡張してみよう。

○ いろいろや音楽や背景にしてみよう。

○ ダンスする人を増やしてみよう。

回転模様(かいてんもよう)



きれいな回転のもようを描きます。

次の順番で作っていくよ。

1. 回転模様(かいてんもよう)のコードをつくる。
2. 改良・拡張してみよう。



回転模様(かいてんもよう) [1]

1. 目玉のコードをつくる。

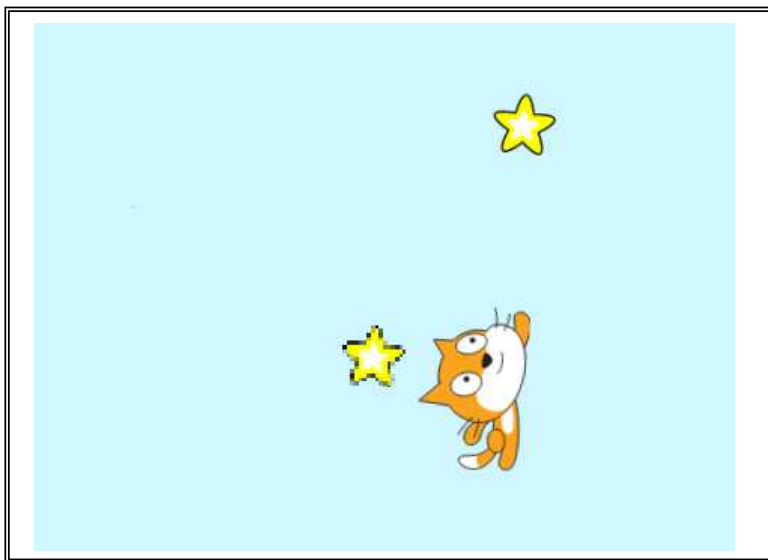


3. 改良・拡張してみよう。

○ [()回繰り返す][()度回す][()歩動かす]の値を変えてちがう図形を描いてみよう。

○ 描いたもようを残すようにしてみよう。

フライングネコアタック



マウスを操作して、星をつかまえよう。
次の順番で作っていくよ。

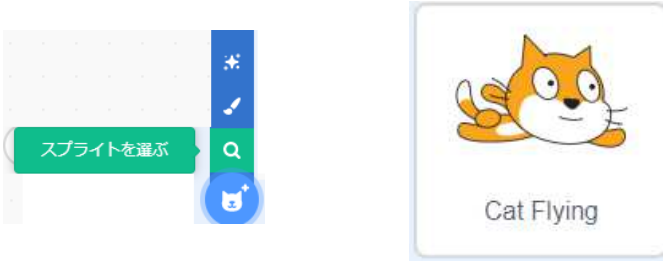
1. 空飛ぶネコのコードを作る。
2. 星のコードを作る。
3. 星をふやしてみよう。
4. 改良・拡張してみよう。



フライングネコアタック[1]

1. 空飛ぶネコのコードを作る。

○ 空飛ぶネコのスプライトを追加する。

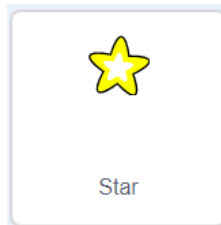


○ コードを作る。



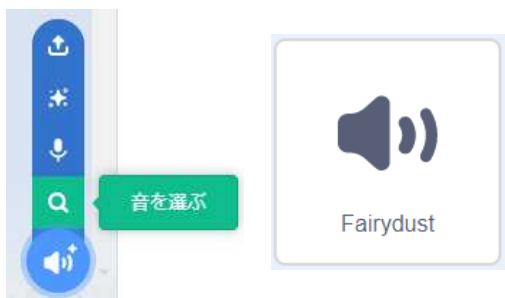
1. 空飛ぶネコのコードを作る。

○ 星のスプライトを追加する。



フライングネコアタック[2]

- 当たった時の音を追加するする。



ヒント:

プログラムでは Fairydust という音を使っていますが、好きな音を選びましょ

- 星のコードを追加する。



3. 星をふやしてみよう。

○ 星一個で正しく動くか確認しよう。

○ 星をふやす。



ヒント:

スプライトを右クリックしてから複製します。
同じ動きをするスプライトがコピーされます。

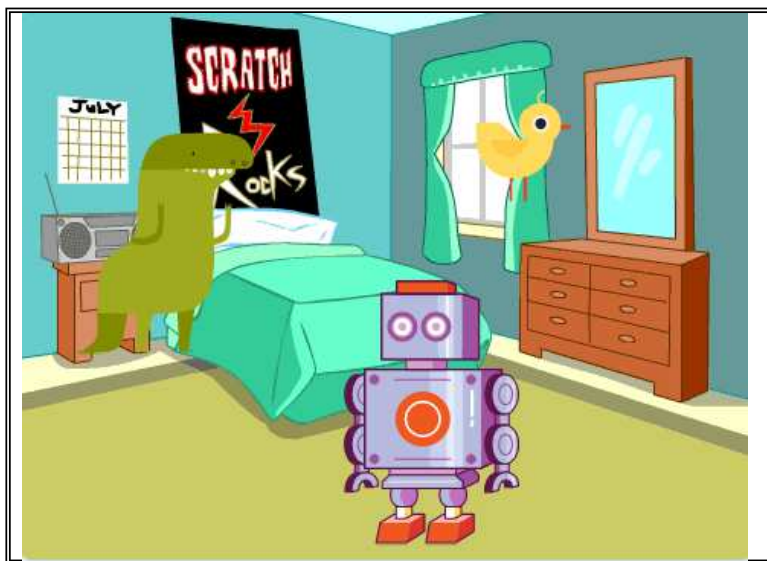
4. 改良・拡張してみよう。

○ いろいろなものを飛ばしてみよう。

○ ネコが星をとったときの、画像の効果をいろいろかえてみよう。

○ 背景をつけてみよう。

忍者の里 Scratch カード 11 難易度 ★★★☆☆(ふつう)
宝探し(たからさがし)



いろいろな場所でマウスをクリックして、隠れているものを探し出そう。

次の順番で作っていくよ。

1. 隠すもののコードを作る。
2. 隠すものをふやしてみよう。
3. 改良・拡張してみよう。

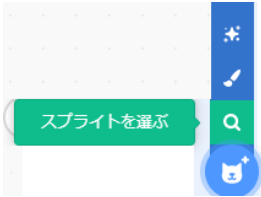


Go.Ota CoderDojo 市川真間

宝探し(たからさがし)[1]

1. かくすもののコードを作る。

○ てきとうなスプライトを追加する。



○ コードを作る。



ヒント:

えらんだスプライトがもともも持っている音に変更してね。

これで1個がかくれているプログラムが完成。動かしてみよう。

宝探し(たからさがし) [2]

2. 隠すものをふやしてみよう。

○ バックバックを使ってコードをコピーする。

バックバックを開きます

(クリックして開きます)

バックバック



コピーしたいプログラムをバックバックにドラッグして入れます。

貼り付ける場合は、バックバックからドラッグします。

4. 改良・拡張してみよう。

○ いろいろな物を隠してみよう。

○ 見つけた時のアクションをいろいろつけてみよう。

○ 見つけた数を表示したり、制限時間をつけてみよう。

ネコうらない



ネコが今日の運勢をうらなうよ。

次の順番で作っていくよ。

1. うらないの内容をつくる。
2. ネコのコードを作る。
3. ないようをいつも見せないようにする。
4. 改良・拡張してみよう。



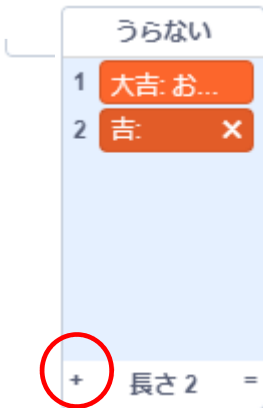
ネコうらない[1]

1. うらないの内容をつくる。
 - うらないを入れるリストを作る。



うらないを入れる[うらない]というリストを作ります。

-
- うらないの内容の追加する



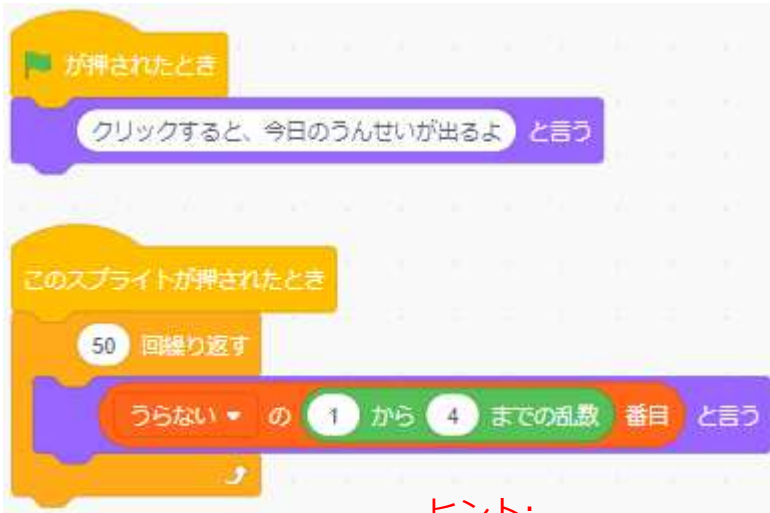
うらないリストに、うらないの内容を入力していきます。
+を押すと、新しい内容を入力することができます。

例えば:次の4個の内容を入力します。

- 大吉: おいしいものがたくさん食べられます
- 吉: おこづかいが少しあがります。
- 小吉: なくした物がみつかります。
- 凶: 給食にきれいなメニューが出ます。

ネコうらない[2]

5. 2. ネコのコードを作る。



ヒント:

[(1)から(4)までの乱数]の(4)には、リストに入力した、うらないの数を指定します。

3. ないようをいつも見せないようにする。。



ヒント:

うらないのリストのチェックを無くすと、ステージから、うらないのリストが表示されなくなるよ。

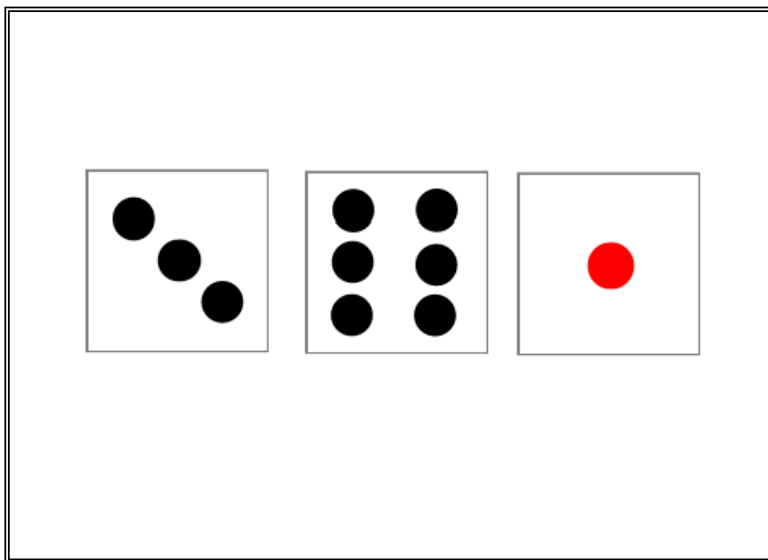
ネコうらない[3]

4. 改良・拡張してみよう。

○ いろいろなうらないの内容にしてみよう。

○ うらないの結果で音を出したり、いろいろな絵やキャラクターを表示してみよう。

スロットマシン



スペースキーを押すと、スロットマシンが動き出すよ。
次の順番で作っていくよ。

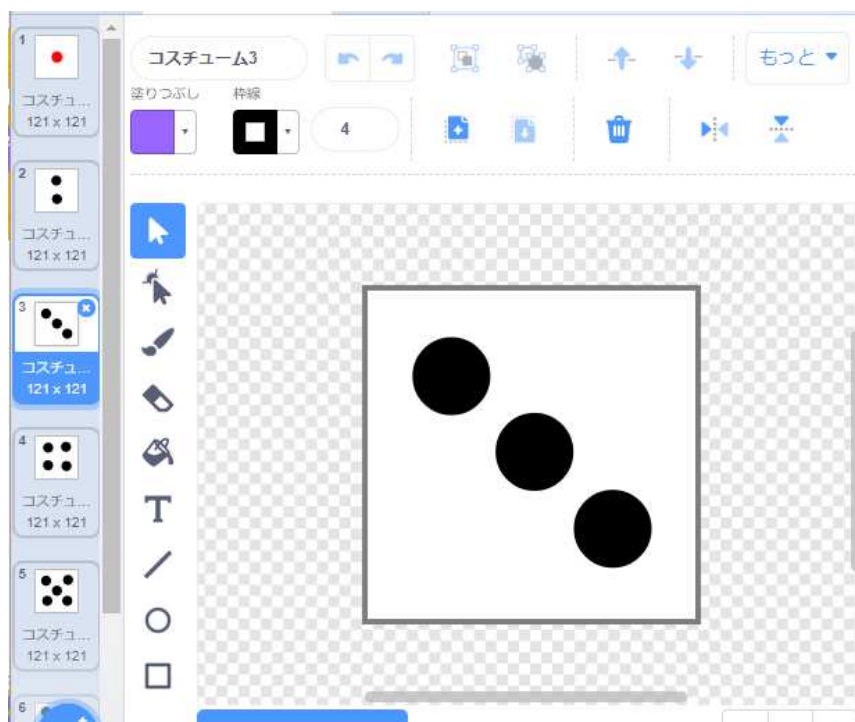
1. 1個のサイコロを作る。
2. サイコロをふやしてみよう。
3. 改良・拡張してみよう。



スロットマシン[1]

1. サイコロを作る。

○ サイコロのコスチュームを作る。



サイコロのコスチューム作ります。そして1から6までの6個のコスチュームを作ります。

ヒント:

四角のSpriteを選んだ後に、サイコロの芽を書き込んで作るともできます。

スロットマシン[2]

- サイコロのコードを追加する。



3. サイコロをふやしてみよう。

- サイコロ一個で正しく動くか確認しよう。

- サイコロをふやす。



ヒント:

スプライトを右クリックしてから複製します。
同じ動きをするスプライトがコピーされます。

4. 改良・拡張してみよう。

○ いろいろな絵柄のスロットマシンを作ってみよう。

○ 絵柄がそろった時に商品を出してみよう。

○ 背景をつけてみよう。

競走(きょうそう)



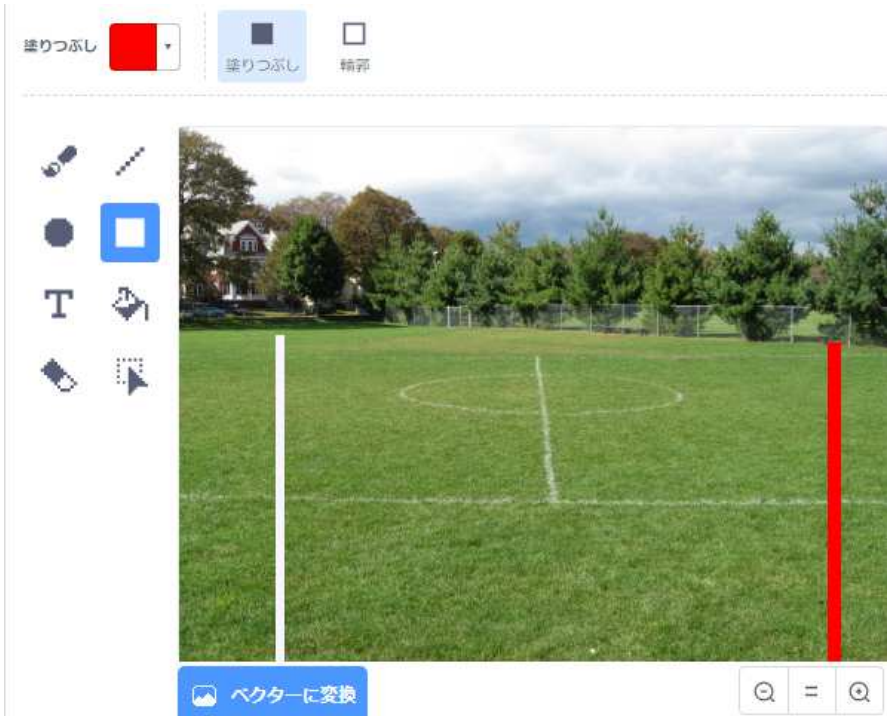
スペースキーをネコが走るよ。他の動物と競争だ。
次の順番で作っていくよ。

1. 競技場をつくる。
2. ネコのコードをつくる。
3. きょうりゅうのコードをつくる。
4. うさぎのコードをつくる。
5. 改良・拡張してみよう。



競走(きょうそう)[1]

1. 競技場をつくる。



背景にスタートラインとゴールラインを追加します。

ヒント:

ゴールは色で判定するので、背景で使っていない赤などの色にします。

競走(きょうそう)[2]

2.ネコのコードをつくる。

が押されたとき

x座標を -201 、y座標を -117 にする

ずっと

もし スペース キーが押された なら

スペース キーが押された ではない まで待つ

20 歩動かす

もし 色に触れた なら

ゴール!!! と 2 秒言う

すべてを止める

もし 色に触れた なら

色 0

鮮やかさ 100

明るさ 100

ヒント:

[XX 色に触れた]の色指定については、スポイトで吸い取ると正しく指定できます。

競走(きょうそう)[3]

3. きょうりゅうのコードを作る。



競走(きょうそう)[4]

3. うさぎのコードを作る。

```

当が押されたとき
  x座標を -206 、y座標を -20 にする
  ずっと
    もし 色に触れた なら
      ゴール!!!! と 2 秒言う
      すべてを止める
    次のコスチュームにする
  0 から 8 までの乱数 歩動かす

```

← きょうりょうと
ここがちがう。

← きょうりょうと
ここがちがう。

ヒント:

うさぎのコードはきょうりゅうと似ているので、バックパックをつかってコピーすると楽です。

4. 改良・拡張してみよう。

- ネコ、きょうりゅう、うさぎの走り方を変化させてみよう。
- ネコが勝ったらメダルをあげてみよう。
- タイムを計ってみよう。

ジャンプでよける



ころがってくるボールをジャンプしてよけます。

上向きの矢印でジャンプするよ。

次の順番で作っていくよ。

1. 地面のある背景を選ぼう。
2. ボールのコードを作ろう。
3. ジャンプ(ネコ)のコードを作ろう。
4. 改良・拡張してみよう。



ジャンプでよける[1]

1. 地面のある背景を選ぼう。



ネコが着地して止まるように地面のある背景を選びます。

2. ボールのコードを作ろう。



ジャンプでよける[2]

3. ジャンプ(ネコ)のコードを作ろう。



○

ヒント:

[XX 色に触れた]の色指定については、スポイトで吸い取ると正しく指定できます。

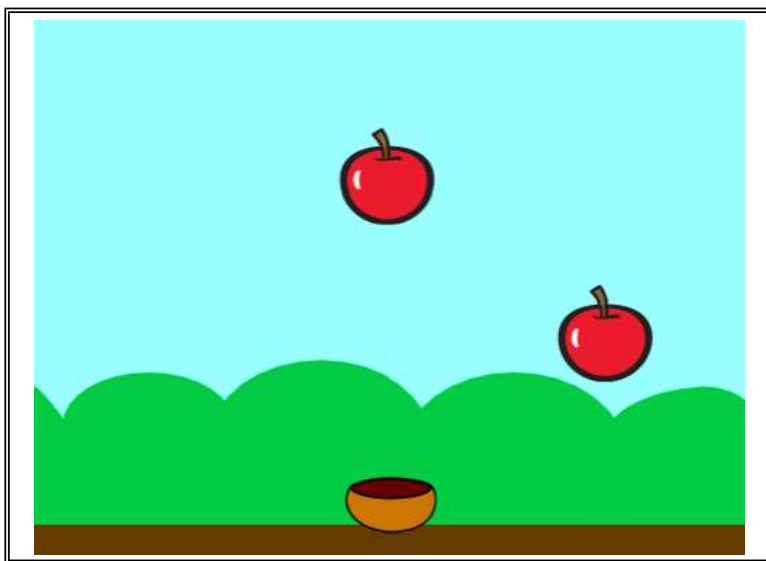


ジャンプでよける[3]

4. 改良・拡張してみよう。

- ジャンプの仕方を変えてみよう。
- ボールの出てくるタイミングや数を変えてみよう。
- たくさんよけることができたなら商品を出してみよう。

キャッチゲーム



落ちてくるものをキャッチしよう。
次の順番で作っていくよ。

1. うけとるもののコードを作る。
2. 落ちるもののコードを作る。
3. 落ちるものをふやしてみよう。
4. 改良・拡張してみよう。



キャッチゲーム[1]

1. うけとるもののコードを作る。



キャッチゲーム[2]

2. 落ちるもののコードを作る。

The image displays three Scratch code blocks for a falling object in a catch game. Each block starts with a 'when green flag clicked' event.

- Top Block:** A 'wait 10 seconds' block (1 from 10 random / 10) is followed by a 'go to some location' block, then 'set y coordinate to 140', and a 'forever' loop containing 'decrease y coordinate by 5'.
- Bottom-Left Block:** A 'forever' loop containing a 'when touching end' condition, followed by 'go to some location', and 'set y coordinate to 140'.
- Bottom-Right Block:** A 'forever' loop containing a 'when touching Bowl' condition, followed by 'play Chomp sound until finished', 'go to some location', and 'set y coordinate to 140'.

3. おとすものをふやしてみよう。

○ 一個で正しく動くか確認しよう。

○ ふやす。



ヒント:

スプライトを右クリックしてから複製します。

同じ動きをするスプライトがコピーされます。

4. 改良・拡張してみよう。

○ いろいろなものでキャッチしてみよう。

○ いろいろなものを落としてみよう。飛ばしてみよう。

ものを横にとばしたら何かできる？

ものを下から上にとばしたら何ができる？

○ 得点をつけてみよう、制限時間をつけてみよう。

(No.20 キャッチゲームゲーム(上級)を見てね)

忍者の里 Scratch カード 17 難易度 ★★★☆☆(ふつう)
ペアダンス



音楽に二人が合わせてダンスをおどるよ。

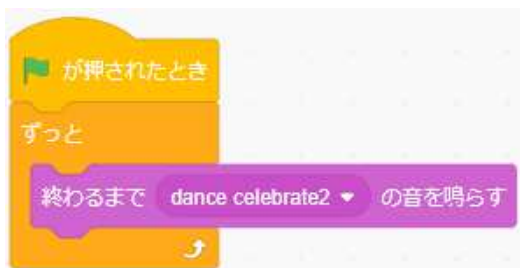
次の順番で作っていくよ。

1. ダンサー1のコードをつくる。
2. ダンサー2のコーサをつくる。
3. 改良・拡張してみよう。



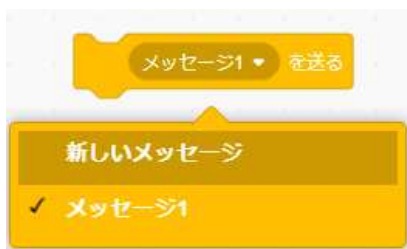
ペアダンス[1]

1. ダンサー1のコードをつくる。



ヒント:

「メッセージ2」は新しいメッセージでつくります。



ペアダンス[2]

2. ダンサー2のコードをつくる。

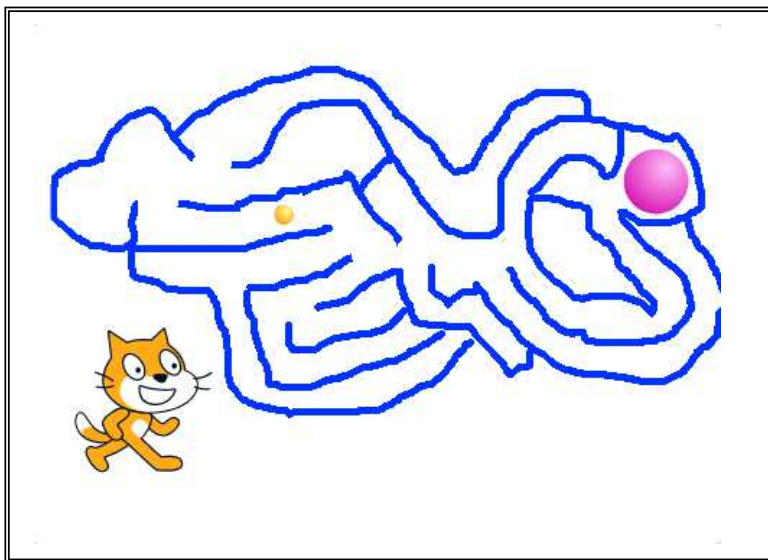


3. 改良・拡張してみよう。

○ いろいろや音楽や背景にしてみよう。

○ ダンスする人を増やしてみよう。

迷路(めいろ)



マウスで小さな玉を動かしながらゴールに向かうよ。
(小さな玉はマウスカーソルで触れると動き出します。)
次の順番で作っていくよ。

1. 背景で迷路をつくる。
2. ゴールの大きな玉をつくる。
3. 動かす小さな玉を作る。
4. 改良・拡張してみよう。



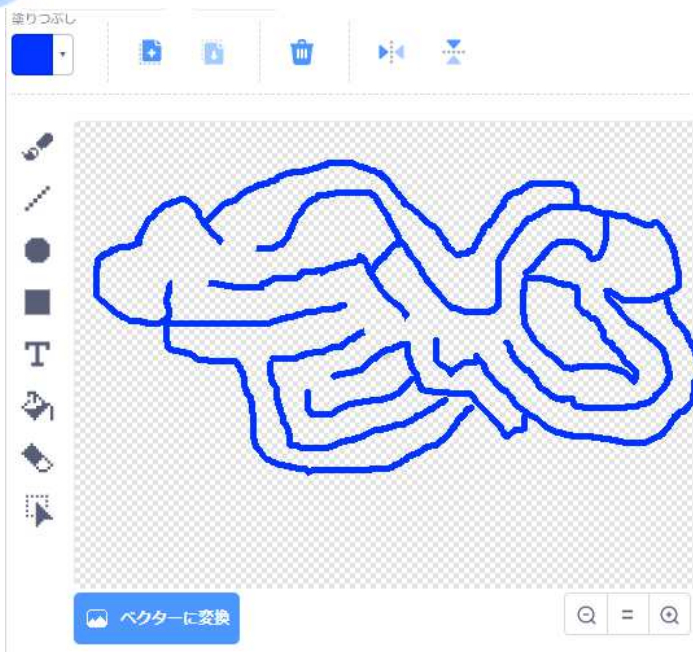
迷路(めいろ) [1]

1. 背景で迷路をつくる。。

○ うらないを入れるリストを作る。




背景として、迷路を作成してみよう。あまり道を細くすると難しくなるよ。



ヒント:

ビットマップで書くとペンで線がかけるよ。

 ビットマップに変換 をクリックするとビットマップで描けるよ。

迷路(めいろ) [2]

2. ゴールの大きな玉をつくる。



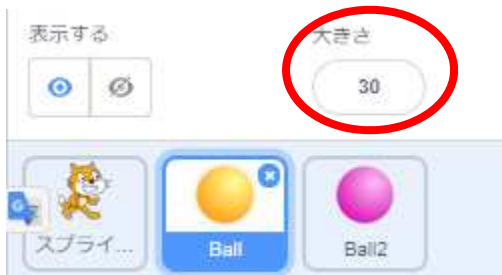
Ball のスプライトを選んで、迷路のゴールにおきます。

ヒント:

あらかじめ用意されている ball のスプライトには、いろいろな色のコスチュームが用意されています。好きな色をえらんで。

3. 動かす小さな玉を作る。

○ 小さな玉の大きさを調整する。



動かす Ball の大きさを、迷路を通れるように小さくして。

迷路(めいろ) [3]

○ 小さな玉のコードを作る。

スタート位置のざひょうを指定して。

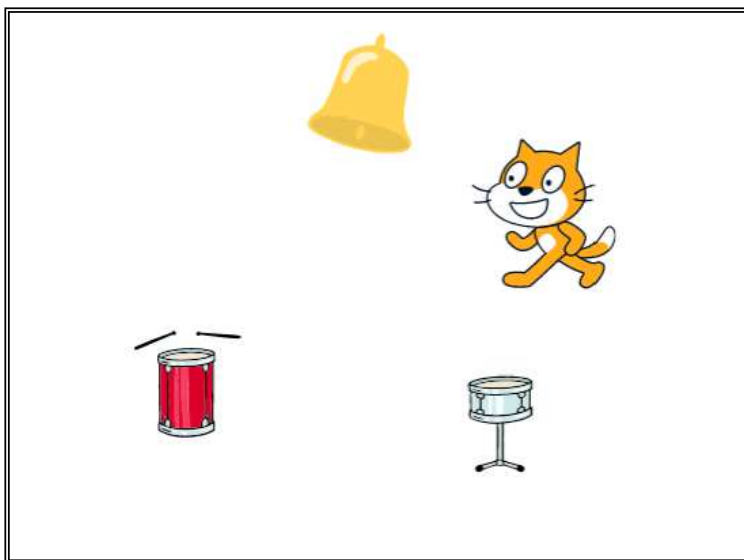
ゴールの sprites を指定して。

ヒント:
スポイトを使って迷路のかべの色をしているよ。

4. 改良・拡張してみよう。

- いろいろな迷路を作ってみよう。
- 時間制限をつけてみよう。
- 何回か失敗してもできるようにしてみよう。

ぐるぐる太鼓



ネコがぐるぐる回ってタイコをたたいていきます。
次の順番で作っていくよ。

1. ネコのコードを作ろう。
2. ドラムの配置しよう。
3. 改良・拡張してみよう。



ぐるぐる太鼓[1]

1. ネコのコードを作ろう。



ヒント:

拡張機能の[ペンを下す]を使うとネコが動いた場所がわかるよ



ぐるぐる太鼓[2]

2. ドラムを配置しよう。

○ 音がでそうなスプライトを追加する。



ヒント:

スプライトを選択する時、音楽を指定すると楽器などが表示されます。

○ コードを追加する。

すべてのスプライトに同じようなコードを作ろう。



ヒント:

別の音を追加してもいいよ。

<- スプライトごとに出せる音に、ここだけ変更します。

ぐるぐる太鼓[3]

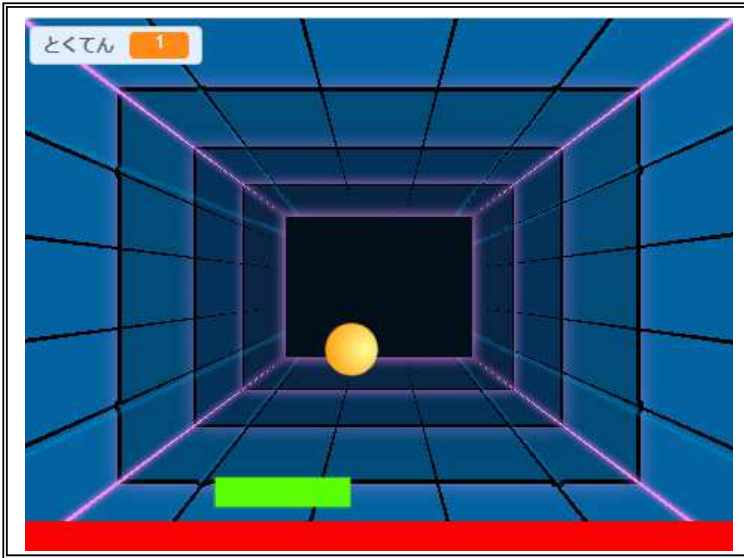
3. 改良・拡張してみよう。

○ いろいろな物や音を配置してみよう。

○ ネコの動きを変えてみよう。

○ ペンを使ってネコの動きを調べてみよう。

忍者の里 Scratch カード 20 難易度 ★★★★★(むずかしい)
ピンポン



マウスを操作して、ラケットでボールを跳ね返すよ。跳ね返した回数も得点です。赤いところに、触れると終了。

次の順番で作っていくよ。

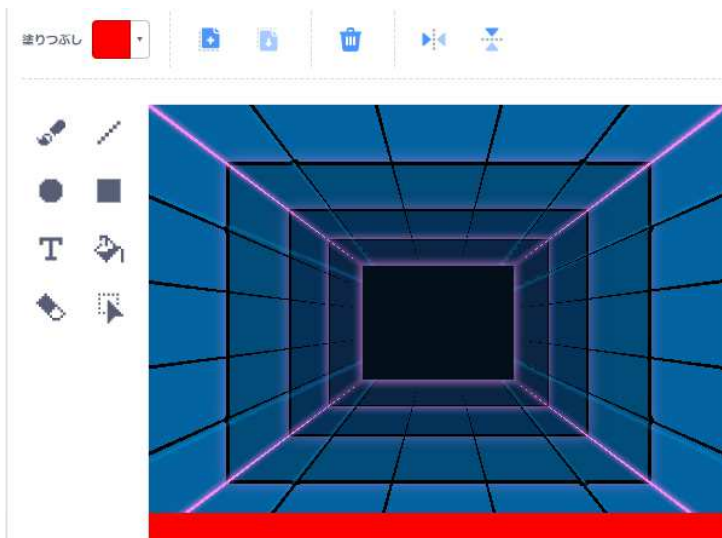
1. 背景を作ろう。
2. ラケットを作ろう。
3. ボールのコードを作ろう。
4. 得点をだそう。
5. 改良・拡張してみよう。



Go.Ota CoderDojo 市川真間

ピンポン[1]

1. 背景を作ろう。

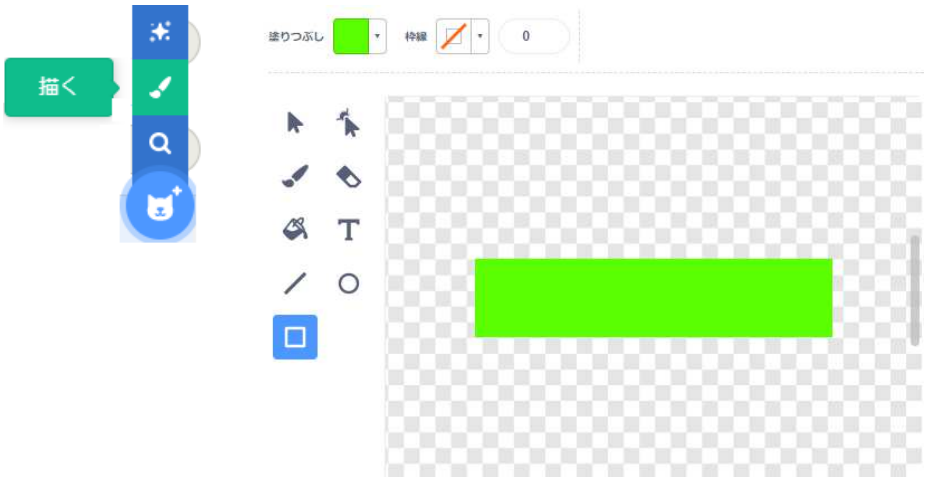


背景を選びます。次の、ポールがふれた時に、止まる合図になるように、下の方に赤い太めの線を追加します。

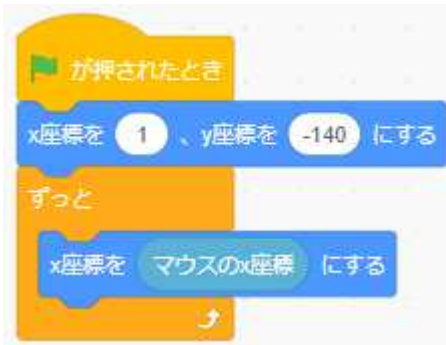
ピンポン[2]

2. ラケットを作ろう。

○ ラケットの_sprites を作る。



○ ラケットのコードを作る。



ピンポン[3]

3. ボールのコードを作ろう。



The image shows a Scratch script for a ball character. It starts with a 'when green flag clicked' event block. The first block is 'turn 180 degrees'. The second block is 'set x coordinate to -6, y coordinate to 138'. This is followed by a 'forever' loop. Inside the loop, there are three 'if' blocks: 1. 'if sprite 1 touched' - 'turn degrees from -60 to 60'. 2. 'if color touched' (red) - 'stop everything'. 3. 'if edge touched' - 'step 10'.



The image also shows a color picker dialog box. It has a color wheel at the top, a '鮮やかさ' (saturation) slider set to 100, and a '明るさ' (brightness) slider set to 100. A red circle highlights the 'color' icon at the bottom of the dialog.

ヒント:
[XX 色に触れた]の色指定については、スポイトで吸い取ると正しく指定できます。

ピンポン[4]

4. とくてん関係のコードを作ろう。

○ とくてんの変数を作る。



○ ラケットのコードを追加する。



ヒント:
これがある場合とない場合
にどうちがうか考えてみよう。

ピンポン[5]

5. 改良・拡張してみよう。

- 背景やラケットをいろいろな物に変えて、ボールを防ぐプログラムを考えてみよう。
- プログラムをもっと難しくしてみよう。
- たくさん打ち返したら、ゲームを終了させてみよう。
- 時間制限をつけてみよう。

クイズ



ネコがクイズを出すと。

順番にクイズを出すコードとランダムに出すコードの例があるよ。

次の順番で作っていくよ。

1. クイズの内容をつくる。
2. ネコのコードを作る(順番に出す)。
3. ネコのコードを作る(ランダムに出す)。
4. 改良・拡張してみよう。



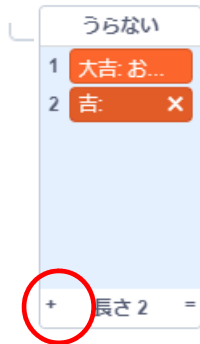
クイズ[1]

- クイズの内容をつくる。
 - クイズを入れるリストを作る。



うらないを入れる[問題]と[正答]の二つのリストを作ります。

- 問題と正答の内容の追加する



問題と正答のリストに、クイズの内容を入力していきます。
+を押すと、新しい内容を入力することができます。

例えば:次の4個の内容を入力します。

問題	正答
お米を一番食べる都道府県は?	熊本県
バナナを一番食べる都道府県は?	和歌山県
納豆を一番食べる都道府県は?	宮城県
餃子を一番食べる都道府県は?	京都府
お店でうどんを一番食べる都道府県は?	香川県

クイズ[2]

2. ネコのコードを作る(順番に出す)。

○ どのクイズを出すか決める変数をつくる。



[問題番号]の変数を作ります。

○ 問題と正答をいつも見せないようにする。。



ヒント:

リストのチェックを無くすと、ステージから、問題と正答のリストが表示されなくなるよ。

ヒント:

[(1)から(4)までの乱数]の(4)には、リストに入力した、うらないの数を指定します。

クイズ[3]

- 順番に問題を出すコード

The image shows a Scratch script for a quiz program. The script starts with a 'when green flag is clicked' event block. The first block is a 'say' block with the text 'これからクイズを出すよ。都道府県名で答えて' and a duration of 3 seconds. This is followed by an 'set question number to 0' block. A 'repeat 5 times' loop contains the following blocks: 'increase question number by 1', 'ask question' block, an 'if-then-else' conditional block, and a 'say' block. The 'if-then-else' block checks if the user's answer matches the correct answer. If correct, it says 'すごいね! 正解だと' for 3 seconds. If incorrect, it says '残念! 正解は' followed by the correct answer and 'と' for 3 seconds. The script ends with a 'stop all sounds' block.

```
Scratch Script:  
1. when green flag is clicked  
2. say [これからクイズを出すよ。都道府県名で答えて] for 3 seconds  
3. set question number to 0  
4. repeat 5 times  
5.   increase question number by 1  
6.   ask question [question number] and wait  
7.   if answer = correct answer [question number] then  
8.     say [すごいね! 正解だと] for 3 seconds  
9.   else  
10.    say [残念! 正解は correct answer と] for 3 seconds  
11. stop all sounds
```

クイズ[4]

- ランダムに問題を出すコード

The image shows a Scratch script for a quiz program. The script starts with a 'when green flag is clicked' event block. The first block is a 'say' block with the text 'これからクイズを出すよ。都道府県名で答えて' and a duration of 3 seconds. This is followed by a 'forever' loop block. Inside the loop, there are four blocks: 1. A 'set random number' block that sets a random number between 1 and 5 to the 'question number' variable. 2. A 'wait' block that waits for the 'question number' variable. 3. An 'if-then' block that checks if the 'answer' variable is equal to the 'correct answer' for the 'question number'. If true, it says 'すごいね! 正解だと' for 3 seconds. 4. An 'if-then' block that checks if the 'answer' variable is not equal to the 'correct answer'. If true, it says '残念! 正解は' followed by the 'correct answer' for 3 seconds. The script ends with a 'stop' block.

```
Scratch Script:  
- when green flag is clicked  
  - say [これからクイズを出すよ。都道府県名で答えて] for 3 seconds  
  - forever loop  
    - set random number [1] to [5] to question number  
    - wait for question number  
    - if answer = correct answer of question number  
      - say [すごいね! 正解だと] for 3 seconds  
    - if answer != correct answer of question number  
      - say [残念! 正解は] + correct answer of question number for 3 seconds  
  - stop
```

クイズ[5]

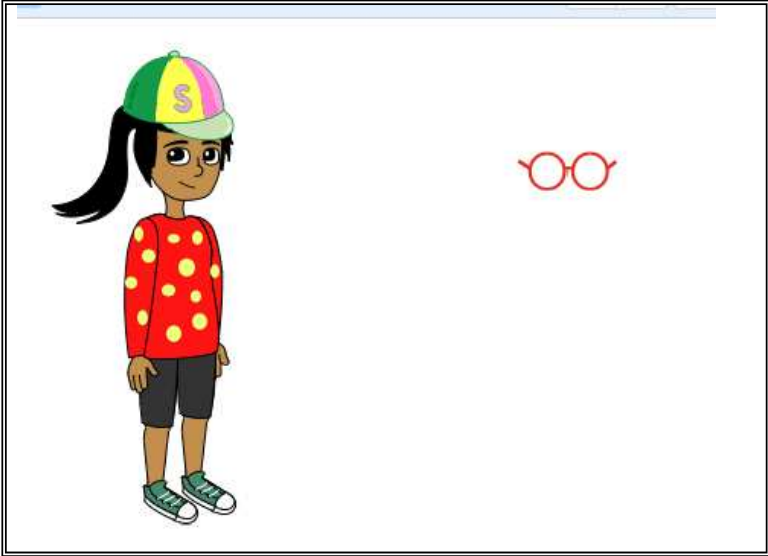
4. 改良・拡張してみよう。

○ いろいろなクイズの内容にしてみよう。

○ クイズの点数を出してみよう。

○ クイズの結果で音を出したり、いろいろな絵やキャラクターを表示してみよう。

着せ替え



服を変えたり、メガネやぼうしを着せてみよう。

大砲(たいほう)は←→のキーで回転し、スペースで玉を発射するよ。

次の順番で作っていくよ。

1. 人の服をデザインしたりコードを作る。
2. メガネのコードを作る。
3. ぼうしのコードを作ろう。
4. 改良・拡張してみよう。



着せ替え[1]

1. 人の服をデザインしたりコードを作る。

○ コスチュームを追加して、好きなデザインの服を作る。



コスチュームを指定して、右クリックすると、複製ができます。



自分の好きな服をデザインしてみよう。

○ 人のコードの追加



着せ替え[2]

2. メガネのコードを作ろう。



The image shows a Scratch script for changing a costume to glasses. It starts with a 'when clicked' event block. The first block is 'set costume to glasses-a'. This is followed by a 'forever' loop containing two 'if touched by Dani' blocks. The first 'if' block has 'x coordinate is -115 and y coordinate is 92'. The second 'if' block has 'x coordinate is 135 and y coordinate is 80'. To the right, there is a 'when this sprite is clicked' event block with a 'set next costume' block. Below this is a red glasses icon.

が押されたとき

コスチュームを glasses-a にする

ずっと

もし Dani に触れた なら

x座標を -115 、y座標を 92 にする

でなければ

x座標を 135 、y座標を 80 にする

このスプライトが押されたとき

次のコスチュームにする



ヒント:

赤いメガネをクリックする時は、赤い線の部分を正確にクリックしないとダメだよ。

3. ぼうしのコードを作ろう。



The image shows a Scratch script for changing a costume to a hat. It starts with a 'when clicked' event block. The first block is 'set costume to hat-a'. This is followed by a 'forever' loop containing two 'if touched by Dani' blocks. The first 'if' block has 'x coordinate is -120 and y coordinate is 130'. The second 'if' block has 'x coordinate is 135 and y coordinate is -40'. To the right, there is a 'when this sprite is clicked' event block with a 'set next costume' block.

が押されたとき

コスチュームを hat-a にする

ずっと

もし Dani に触れた なら

x座標を -120 、y座標を 130 にする

でなければ

x座標を 135 、y座標を -40 にする

このスプライトが押されたとき

次のコスチュームにする

ヒント:

X座標とY座標の値は、メガネといっしょだよ。ネットで使っている時は、バックパックを使ってコピーできるよ。

着せ替え[3]

4. 改良・拡張してみよう。

○ いろいろな服をデザインしてみよう。

○ いろいろなメガネ以外にいろいろなものを身につけられるようにしてみよう。

○ 背景をつけたり、背景も変わるようにしてみよう。

忍者の里 Scratch カード 23 難易度 ★★★★★(むずかしい)
キャッチゲーム(上級)



No15 のキャッチゲームをパワーアップして、得点や時間制限をつけるよ。ゲームクリア/ゲームオーバー画面もあります。

No15 のキャッチゲームに加えて次の順番で作っていくよ。

1. 得点と制限時間の制限をつくる。
2. ゲームクリア/ゲームオーバーの背景をつくる。
3. 得点関係のコードを作る。
4. 制限時間関係のコードを作る。
5. 改良・拡張してみよう。



Go.Ota CoderDojo 市川真間

キャッチゲーム(上級)[1]

1. 得点と制限時間の制限をつくる。



「タイム」と「とくてん」の変数をつくります。

2. ゲームクリア/ゲームオーバーの背景をつくる。。



はいけいに、ゲームオーバーとゲームクリアを追加します。文字はTで入力できます。

ヒント:

はじめからある背景を複製して追加すると楽です。

キャッチゲーム(上級)[2]

2. 得点関係のコードを作る。

○ おちるものにブロックを追加する。



<- この得点の1ブロックを追加します。
おちるものすべてに追加します。

キャッチゲーム(上級)[3]

- うけとるものにスクリプトを追加する。



このスクリプト全体をついかします。

とくてんが20点になるとゲームクリアーの背景に変わります

この段階で実行してみて、ただしく背景が変わるか確認してみましょう。

キャッチゲーム(上級)[4]

3. 制限時間関係のコードを作る。



このスクリプト全体を、うけとるものにつかします。

タイムが 0 秒になるとゲームオーバー背景に変わります

4. 改良・拡張してみよう。

- いろいろ設定をかえてクリア条件や制限時間を変更してみよう。
- いろいろなものを落としてみよう。飛ばしてみよう。
ものを横にとばしたら何かできる？
ものを下から上にとばしたら何かできる？
- 拾ったら原点になるものを追加してみよう。
- 拾ったら死ぬものを追加してみよう。

忍者の里 Scratch カード 24 難易度 ★★★★★(難しい)
昔話自動



Scratch で自動的に話が進むデジタル絵本を作ってみよう。
(No.7 の昔話の自動化です)

メッセージと画面の変化をつかうよ。

次の順番で作っていくよ。

1. 背景とスプライトを準備しよう。
2. 旗を押したときのコードを作ろう。
3. メッセージ 1 の時のコードを作ろう。
4. メッセージ 2 の時のコードを作ろう。
5. メッセージ 3 の時のコードを作ろう。
6. 改良・拡張してみよう。



昔話 1[1]

1. 背景とスプライトを準備しよう。

○背景を追加しよう



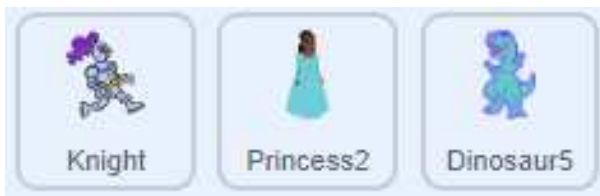
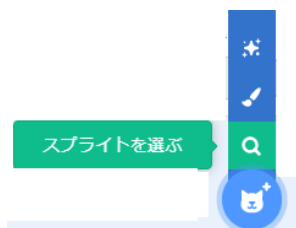
Woods



Castle 1

好きな背景を選んでみよう

○ スプライトを追加しよう。




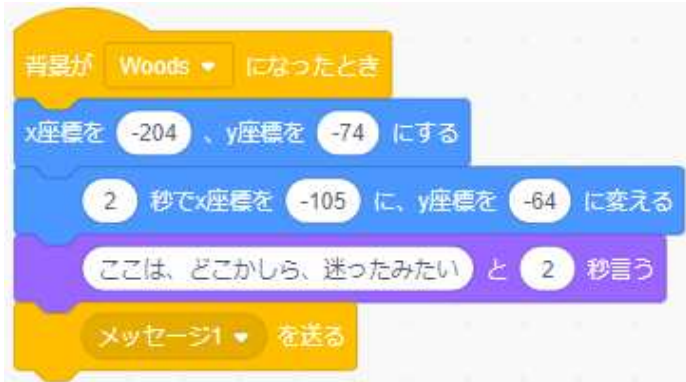
昔話 1 [2]


2. 旗を押したときのコードを作ろう。

○背景のコード作る。(メッセージ3の時のコードも)




○  のコードを作る。



○  のコードを作る。




○  のコードを作る。




昔話 1 [3]

3. メッセージ 1 の時のコードを作ろう。

-  のコードを作る。



```
メッセージ1 を受け取ったとき
  3 秒待つ
  誰か助けて と 2 秒言う
  メッセージ2 を送る
```


-  のコードを作る。




```
メッセージ1 を受け取ったとき
  90 度に向ける
  x座標を 205 、y座標を 75 にする
  表示する
  2 秒でx座標を 11 に、y座標を -37 に変える
  食ってやるぞ と 2 秒言う
```

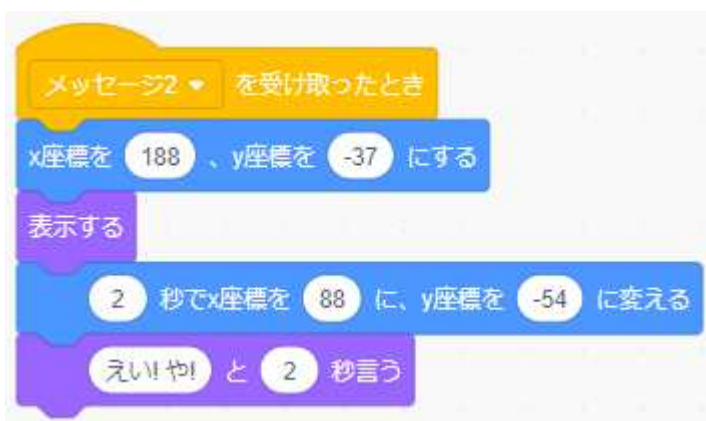
昔話 1 [4]

4. メッセージ 2 の時のコードを作ろう。

-  のコードを作る。



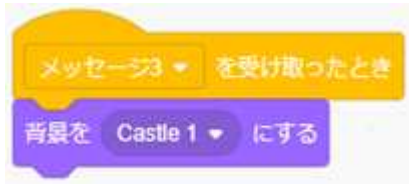
-  のコードを作る。




昔話 1 [5]


5. メッセージ3の時のコードを作ろう。

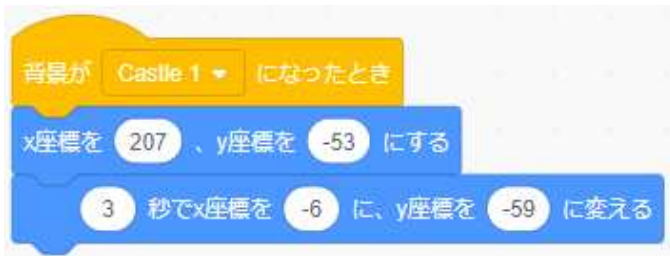
○ 背景のコードを作る。



○  のコードを作る。



○  のコードを作る。



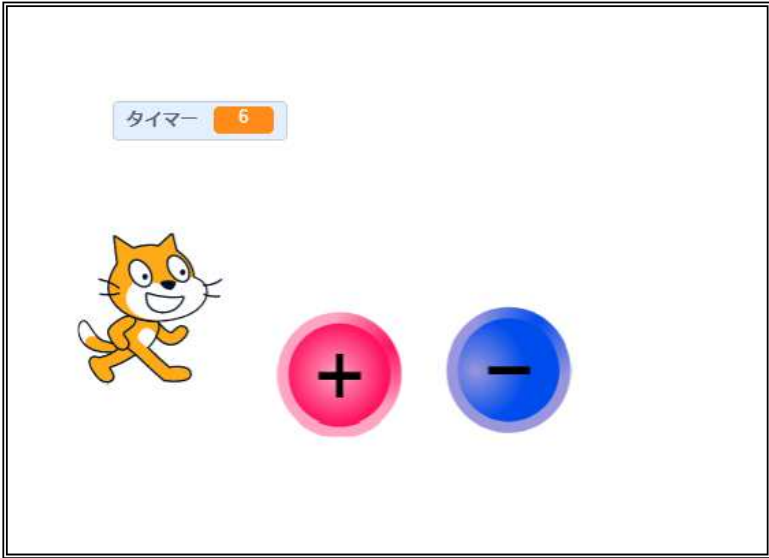
6. 改良・拡張してみよう。

○ いろいろな動きをさせてみよう。

○ いろいろなスプライトを登場させよう。

○ いろいろな物語を作ってみよう。

キッチンタイマー



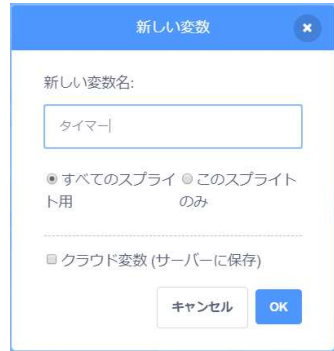
+ -で時間を設定できるキッチンタイマーだよ。

1. タイマーの変数を作る。
2. + -のボタンとコードを作ろう。
3. タイマー(ネコ)のコードを作ろう。
4. 改良・拡張してみよう。



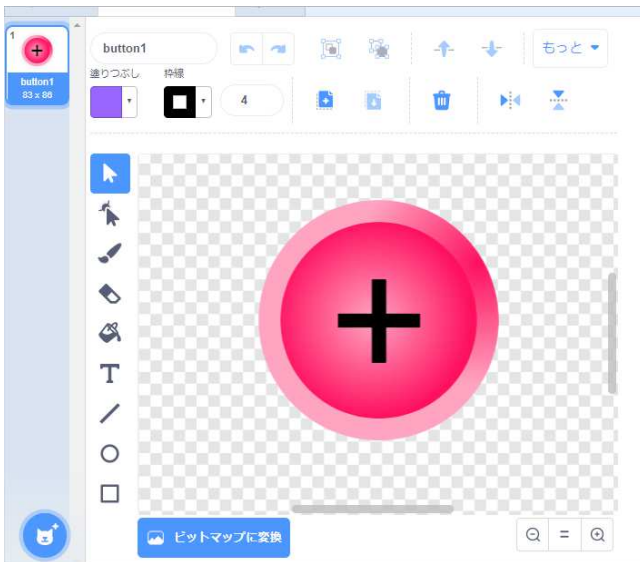
キッチンタイマー[1]

1. タイマーの変数を作る。



2. + -のボタンとコードを作ろう。

○+-のスプライトを作る。



Buttonのスプライトをもとに+と-のボタンを作ります。

キッチンタイマー[2]

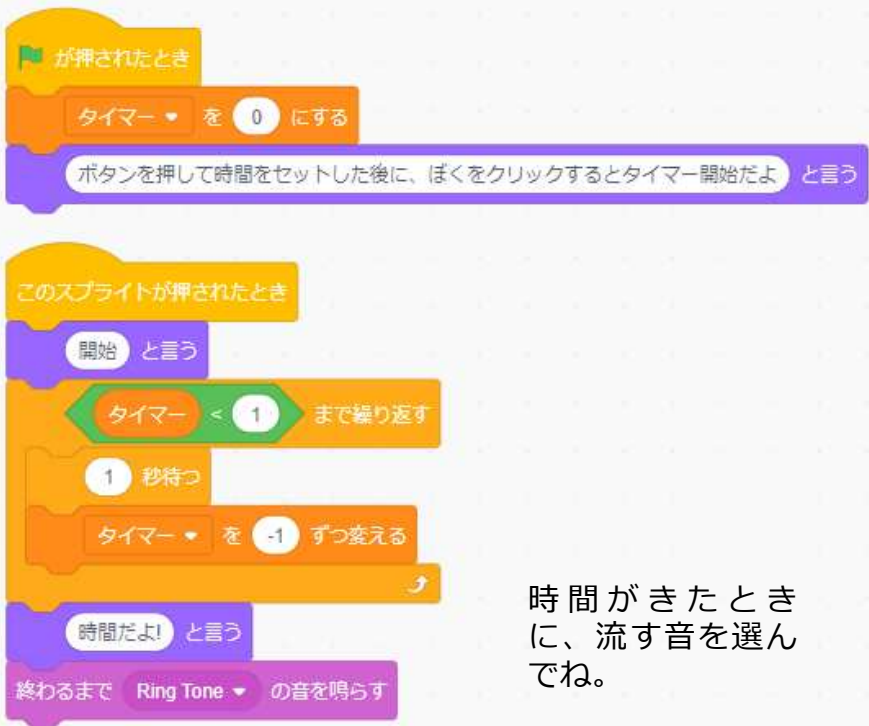
○+-のボタンのコードを作る。



+のボタンのコード

-のボタンのコード

3. タイマー(ネコ)のコードを作ろう。



時間がきたときに、流す音を選んでね。

キッチンタイマー[3]

4. 改良・拡張してみよう。

- キッチンタイマーを便利にしてみよう。
- 時間がきたときの、通知のしかたを工夫してみよう。
- タイマーの数字を大きく表示する方法を考えてみよう。

バーチャルペット



ペットにえさをあげよう。

あげないと、おなかが空いて小さくなってしまふよ。

1. サルのコードを作る
2. バナナのコードをつくる。
3. 改良・拡張してみよう。



バーチャルペット[1]

1. サルのコードを作る。

ヒント
えさ、のメッセージ
を作ります

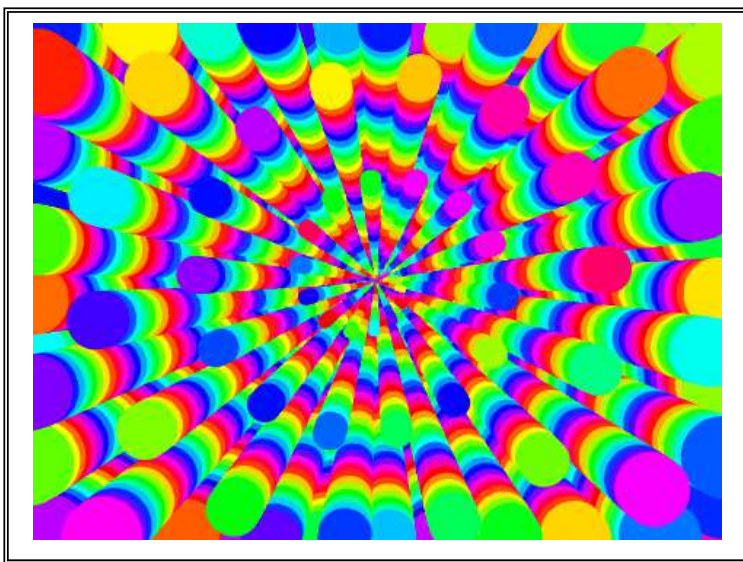
2. バナナのコードをつくる。

3. 改良・拡張してみよう。

○ いろいろな動物を育てよう

○ いろいろなえさや、世話をしてみよう。

[100%PEN]模様



[100%PEN]というスプライトを使わずに、きれいなもようを作るよ。
次の順番で作っていくよ。

1. Pen ブロックの準備をする。
2. スプライトとコードを作る。
3. 改良・拡張してみよう。



[100%PEN]模様[1]

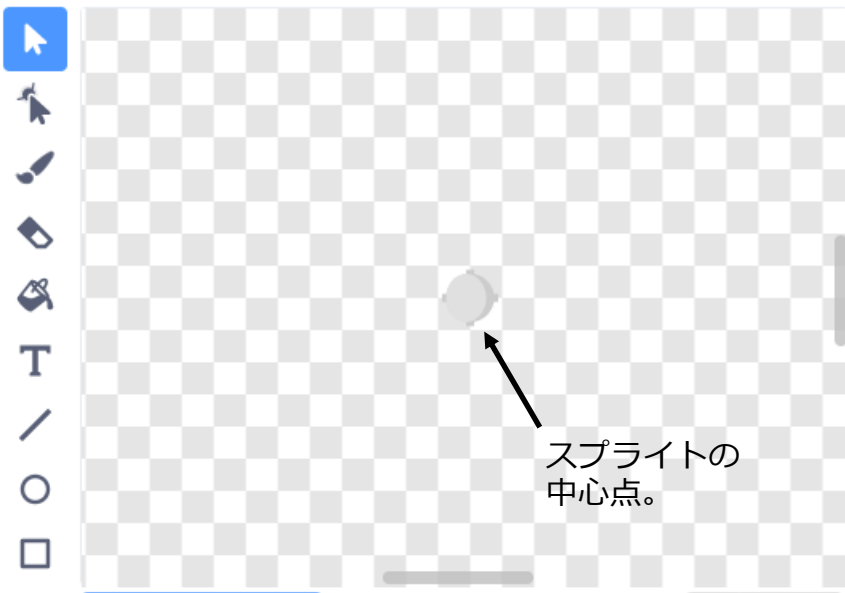
1. Pen ブロックの準備をする。



拡張機能の追加でペンのブロックを追加します。

2. スプライトとコードを作る。

○スプライトを作る。



おへそみたいな示しのスプライトの中心点に、小さな丸い図形をつくれれば十分です。

[100%PEN]模様[2]

○コードを作ろう。

```
当緑旗が押されたとき
  x座標を 0、y座標を 0 にする
  隠す
  全部消す
  ペンを下ろす
  ずっと
    25 度回す
    自分自身 のクローンを作る
```

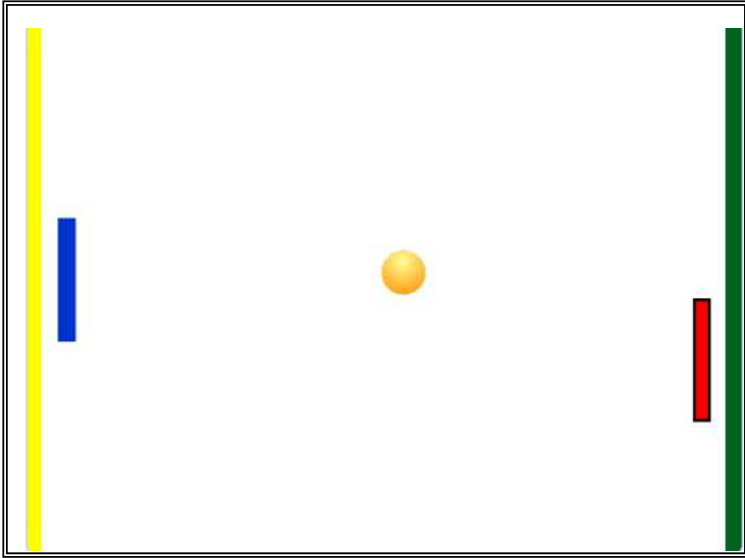
```
クローンされたとき
  ペンの太さを 1 にする
  ペンの色 を 0 から 200 までの乱数 にする
  ずっと
    5 歩動かす
    ペンの色 を 10 ずつ変える
    ペンの太さを 1 ずつ変える
    もし 端 に触れた なら
      このクローンを削除する
```

[100%PEN]模様[3]

3. 改良・拡張してみよう。

- いろいろな模様をつくってみよう。
- 複数の模様(スプライト)を組み合わせてみよう。
- マウスを使ってインターアクティブにしてみよう。

エアホッケー



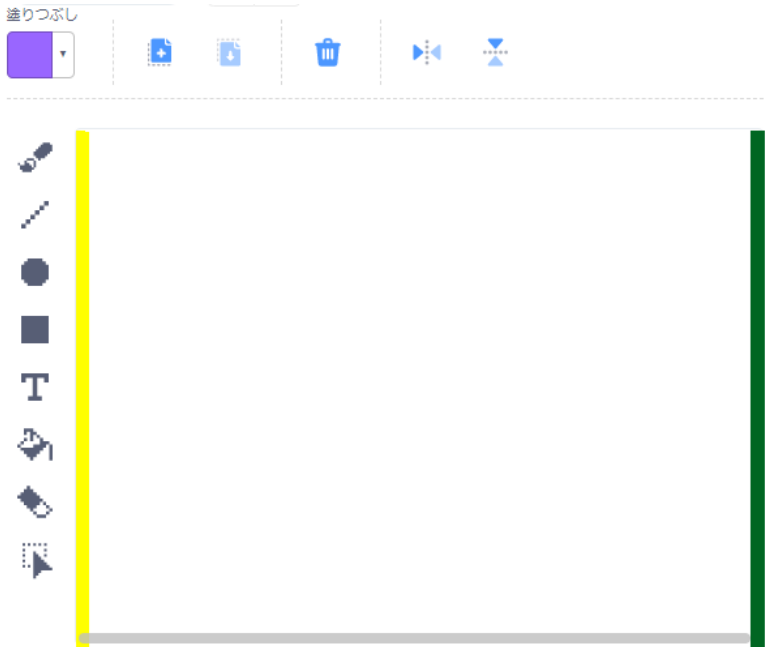
パドルを二人で操作してボール打ち合おう。
次の順番で作っていくよ。

1. 背景を作ろう。
2. ラケットを作ろう。
3. ボールのコードを作ろう。
4. 改良・拡張してみよう。



エアホッケー[1]

1. 背景を作ろう。

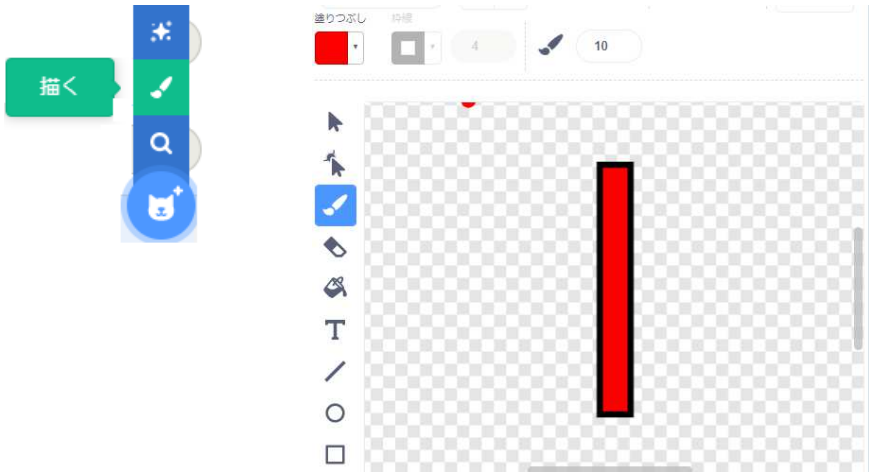


背景にボールがふれた時に、止まる合図になるように、左右に色の異なる線(細い長方形)を描きます。

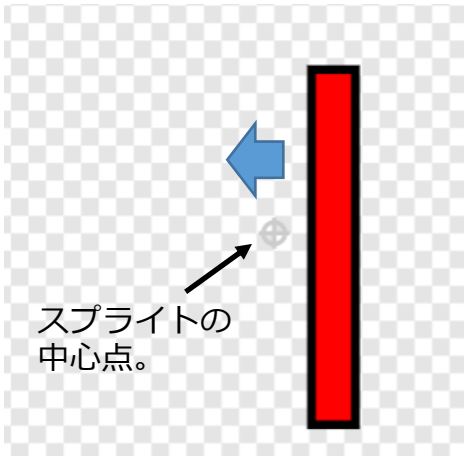
エアホッケー[2]

2. ラケットを作ろう。

○ ラケットのSpriteを作る。



左右の二個のSpriteを作ります。



ヒント:

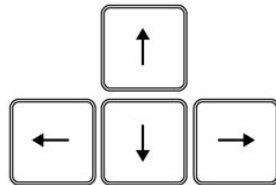
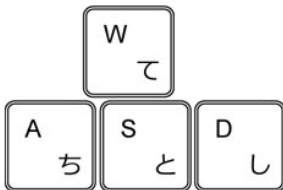
コスチュームを描く画面には、Spriteの中心点を示す印が薄く示されています。Spriteを描く時は、この中心点に真ん中にくるようにします。

エアホッケー[3]

- ラケットのコードを作る。
- ・ 右側のラケットのコード



- ・ 左側のラケットのコード



ヒント:

対戦型のゲームを作る場合は、一人は方向キーを使います。もう一人はAWDSのキーを方向キーの代わりに使用します。

エアホッケー[3]

3. ボールのコードを作ろう。

The image shows a Scratch script for a ball object. It is divided into two main sections: a 'when clicked' section and a 'when received from ovari' section.

When clicked:

- Display
- Set x coordinate to 12, y coordinate to 10
- Turn 45 degrees

When received from ovari:

- Set x coordinate to 12, y coordinate to 10
- Turn 45 degrees
- Display

Forever loop:

- Move 10 steps
- If Sprite 1 is touched:
 - Turn 0 degrees
- If Sprite 2 is touched:
 - Turn 0 degrees
- If Green flag clicked:
 - Send ovari
 - Wait 2 seconds
- If Yellow flag clicked:
 - Send ovari
 - Wait 2 seconds
- If at edge, bounce back

エアホッケー[5]

5. 改良・拡張してみよう。

○玉の速さや角度を変えてみよう。

○ラケットに当たった時に、球の跳ね返る角度を変化させてみよう。

○二人の得点を表示してみよう。

○コンピュータとの対戦型にしてみよう。

○コンピュータとの対戦で、レベル設定をしてみよう。

○コンピュータ同士の超高速体制を実現してみよう。

かいじゅうをたおせ



魔法使いが、クリスタルやロボットを使って、怪獣をやっつけるよ。

1. 魔法使いのコードを作る。
2. クリスタルとロボットのコードを作る。
3. かいじゅうのコードを作る。
4. 改良・拡張してみよう。



かいじゅうをたおせ[1]

1. 魔法使いのコードを作る。

○ 変数の準備する。



魔法使いの向きを記録する「角度」とかいじゅうの動きを記録する「動き」の変数をつくります。

○ 魔法使いのコード



かいじゅうをたおせ[2]

2. クリスタルとロボットのコードを作る。

○ クリスタルのコード。



○ ロボットのコード。



かいじゅうをたおせ[3]

3. かいじゅうのコードを作る。



4. 改良・拡張してみよう。

○かいじゅうからも攻撃してみよう。

○クリスタルやロボットをもっとスムーズに発射できるようにしてみよう。

○もっと、いろいろなものでかいじゅうを倒そう。

○得点や制限時間をつけてみよう。

スペースシューティング



ロボットをビームでやっつけよう。

1. 宇宙船のコードを作ろう。
2. ビームのSpriteとコードを作ろう。
3. ロボットのコードを作ろう。
4. 改良・拡張してみよう。



Go.Ota CoderDojo 市川真間

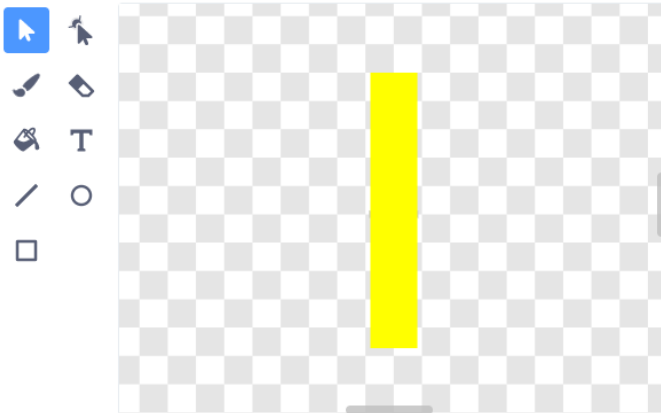
スペースシューティング[1]

1. 宇宙船のコードを作ろう。



2. ビームのSpriteとコードを作ろう。。

○ ビームのSpriteを作る。



ビームのSpriteをつくります。

スペースシューティング[2]

○ビームのコードを作る。



ヒント:
調べるの中のブロックを
使います。

スペースシューティング[3]

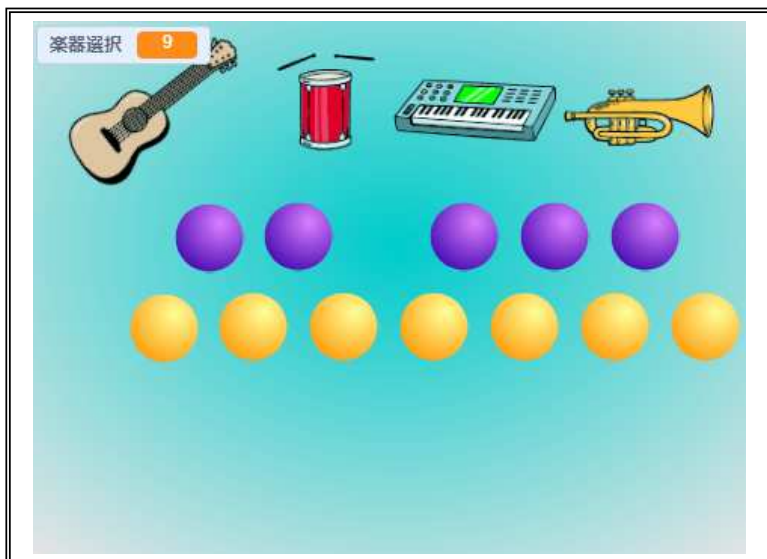
3. ロボットのコードを作ろう。



4. 改良・拡張してみよう。

- 相手からもビームを出してみよう。
- ゲームのクリア条件を考えよう。
- ビームに当たりやすくしてみよう。
- 音や音楽をもっといれてみよう。

エレクトーン



いろいろな楽器の音色がでるエレクトーンを作ります。
次の順番で作っていくよ。

1. 変数などの前準備。
2. 鍵盤のコードを作る。
3. 楽器選択のコードを作る。
4. 改良・拡張してみよう。



エレクトーン[1]

1. 変数などの前準備。

- 楽器の種類を記録する変数を作る。



「楽器選択」の変数を作ります。



- 音楽のブロックを準備する。



拡張機能追加
を選択

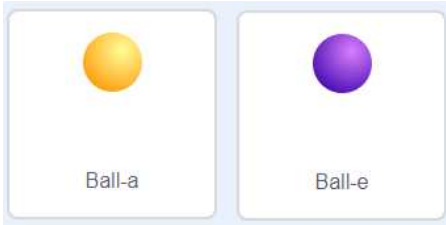


「音楽」の
ブロックを
追加。

エレクトーン[2]

2. 鍵盤のコードを作る。

- 黒鍵と白鍵のSpriteを選択する。

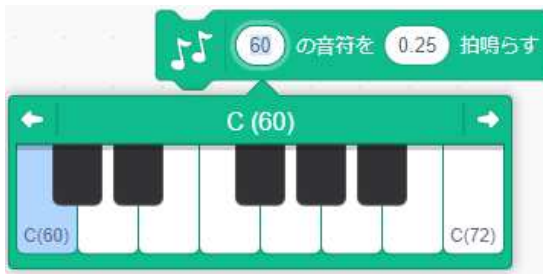


ピアノのような鍵盤を作るのはむずかしいので Ball の Sprite を使い、黄色を白鍵、むらさきを黒鍵として使用します。

- 鍵盤のコードをつくる。



鍵盤は、複製して作ると楽です。
複製したら、表示する位置と、音程を変更します。



音程を設定します。

エレクトーン[3]

3. 楽器選択のコードを作る。

○ 楽器のスク립トを追加する。



好きな楽器のspraitを追加しよう。

○ 楽器のコードを追加する



楽器の番号は
右の図を参考にしてください。



エレクトーン[4]

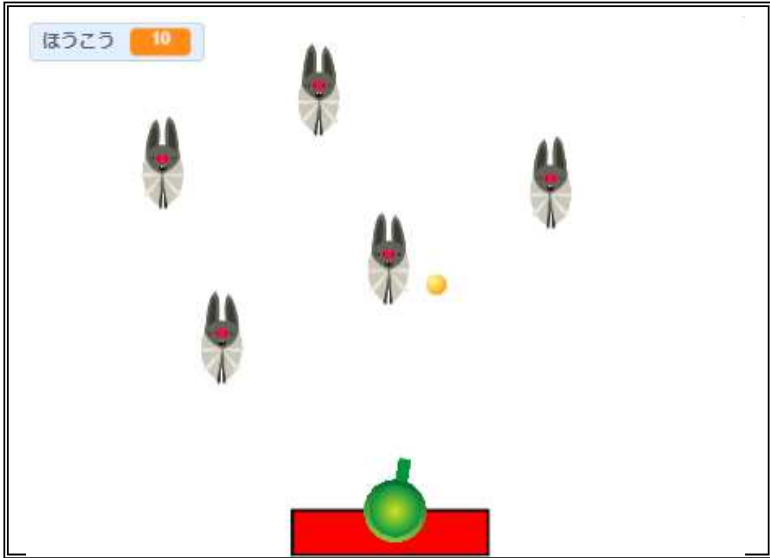
4. 改良・拡張してみよう。

○ いろいろな楽器を追加してみよう。

○ いろいろなものをクリックすることで音ができるようにしてみよう

○ 自動的にメロディやリズムが再生されるようにしてみよう。

コウモリ軍団をたおせ



おそいくるコウモリ軍団を大砲(たいほう)でやっつけるよ。
大砲(たいほう)は←→のキーで回転し、スペースで玉を発射するよ。
次の順番で作っていくよ。

1. スプライトを用意しよう。
2. 砲台のコードを作ろう。
3. コウモリのコードを作ろう。
4. 玉のコードを作ろう。
5. 改良・拡張してみよう。



Go.Ota CoderDojo 市川真間

本資料は、CoderDojo 水戸の教材を参考にしています。

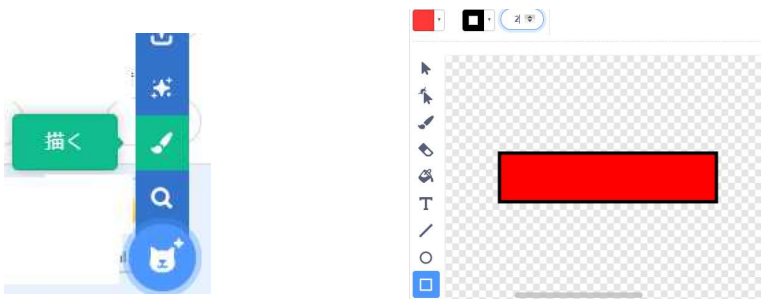
コウモリ軍団をたおせ[1]

1.スプライトを用意しよう。

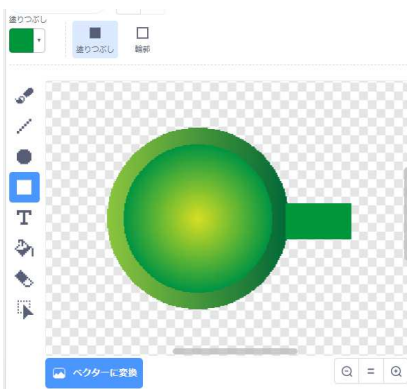
○ コウモリのスプライトの追加



○ 基地のスプライトを描いて追加。



○ たいほうのスプライトの追加



丸いスプライトに出っ張りをつけて、たいほうを作ろう。

大きさも調整してね。

コウモリ軍団をたおせ[2]

○玉のSpriteを追加



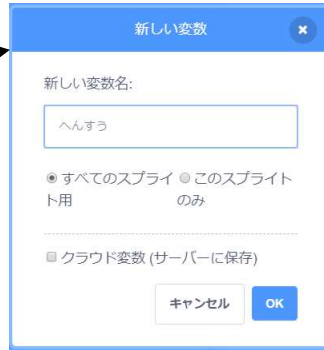
大きさ

30

サイズを小さく
しよう。

2. 大砲(たいほう)のコードを作ろう。

○ [ほうこう]の変数をつくる。



玉を打ち出す方向を記録する
変数を作ります。

○大砲(たいほう)のコード。



コウモリ軍団をたおせ[3]

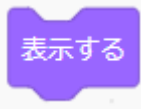
3. コウモリのコードを作ろう。

```
Scratch code for 'when green flag clicked':  
1. when green flag clicked  
2. hide  
3. loop 'forever':  
   - create 1 clone of self  
   - wait 1 second  
   - loop ends
```

```
Scratch code for 'when clone created':  
1. go to location: 'どこかの場所' -> 'へ行く'  
2. set y coordinate to: '130'  
3. pick random direction: '10' from '170' -> '度に向ける'  
4. set rotation method to: '左右のみ' -> 'にする'  
5. show  
6. loop 'forever':  
   - move 10 steps  
   - next costume  
   - if touching 'もし' 'Ball' -> 'に触れた' -> 'なら':  
     - delete this clone  
   - if touching 'もし' 'スプライト1' -> 'に触れた' -> 'なら':  
     - delete this clone  
   - loop ends
```

ヒント:

コウモリや玉は、隠した状態なので、再度表示されたい場合は



を使います。

コウモリ軍団をたおせ[4]

4. 玉のコードを作ろう。

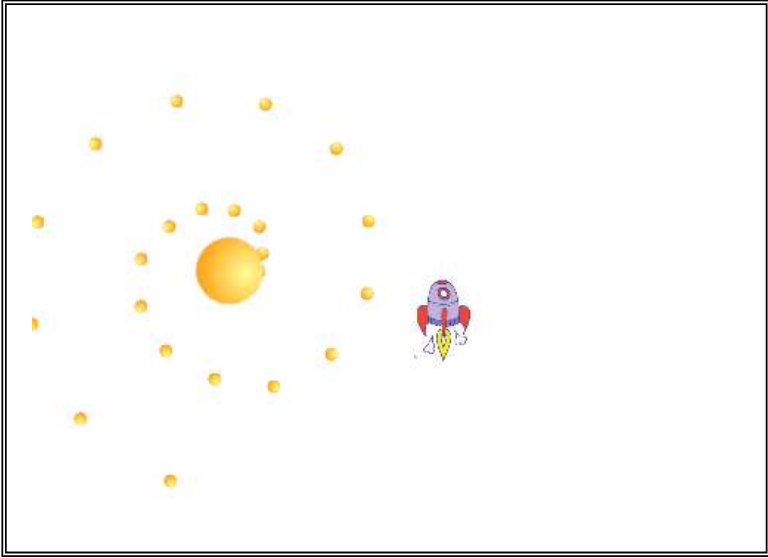


コウモリ軍団をたおせ[5]

5. 改良・拡張してみよう。

- コウモリの出現のタイミングを考えてみよう。
- コウモリを撃ち落とした数を記録してみよう。
- コウモリが基地に何回か突入したら、終了してみよう。
- ゲームとして時間制限をつけてみよう。

弾幕(だんまく)をよける



弾幕をよけるプログラムを作ってみよう。

次の順番で作っていくよ。

1. 弾幕を出すようにしよう。
2. 弾幕をよける宇宙船をつくる。
3. 改良・拡張してみよう。

弾幕をよける[1]

1. 弾幕を出すようにしよう。

○ 弾幕を出すスプライトを追加しよう



○弾幕のコードを追加しよう。



The image shows two Scratch code blocks. The left block is a "when green flag clicked" event block with a "forever" loop. Inside the loop, it contains: a "wait 2 to 4 seconds" block, a "go to some location" block, a "repeat 12 times" block, a "repeat 4 times" block, and a "repeat 30 times" block. The "repeat 30 times" block contains a "create clone of self" block. The right block is a "when cloned" event block. It contains: a "set size to 20%" block, a "repeat until end reached" block, a "repeat 1 times" block, a "step 5" block, and a "delete this clone" block.

弾幕をよける[2]

2. 弾幕をよける宇宙船をつくる。

○ 弾幕を出すスプライトを追加しよう。



○ 当たったときに出る音を追加しよう



○ 宇宙船のコードを追加しよう。



弾幕をよける[3]

3. 改良・拡張してみよう。

○ いろいろな弾幕にしてみよう。

○ 逃げ切る制限され時間をつけてみよう。

○ 何回か当たったらゲームが終わるようにしててみよう。

空中階段



空中にある階段をのぼってゴールを目指そう。

1. 背景・階段の sprites を準備する。
2. ネコのコスチュームを変更する。
3. ネコのコードを作る
4. 改良・拡張してみよう。



空中階段[1]

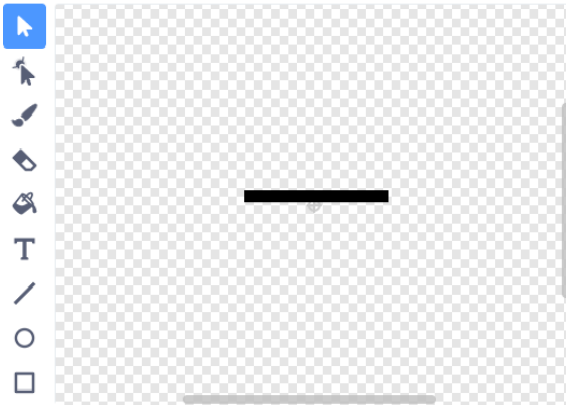
1. 背景・階段の спраイトを準備する。

○ 背景を準備する。



背景に選択します。
地面のあるもの
を選びます。

○ 階段の спраイトを作る。



階段のコスチュー
ムは作成します。

階段の спраイト
は複製して 2 個準
備します。

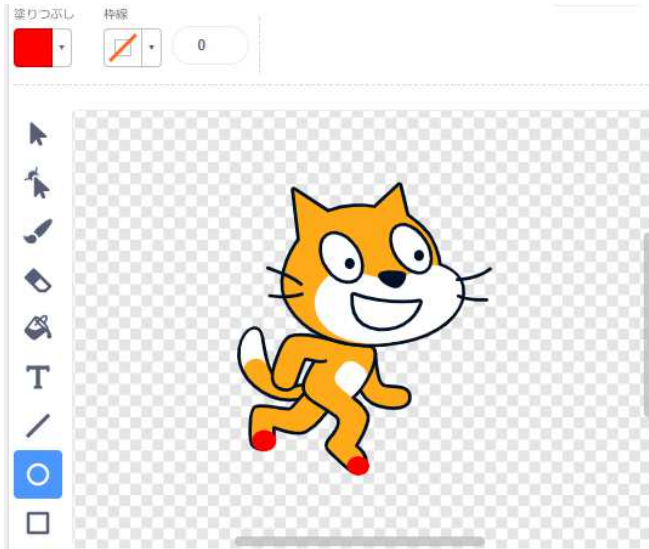
○ ゴールの спраイトを選ぶ。



ゴールになるよ
うな спраイトを選
びます。

空中階段[2]

2. ネコのコスチュームを変更する。



ネコのコスチュームの足先に赤に肉球をつけます。2つのコスチュームにつけます。

この赤い肉球は、階段への着地の判定に使います。

3. ネコのコードを作る。

○ 縦方向の移動する大きさを記録する変数を作る。

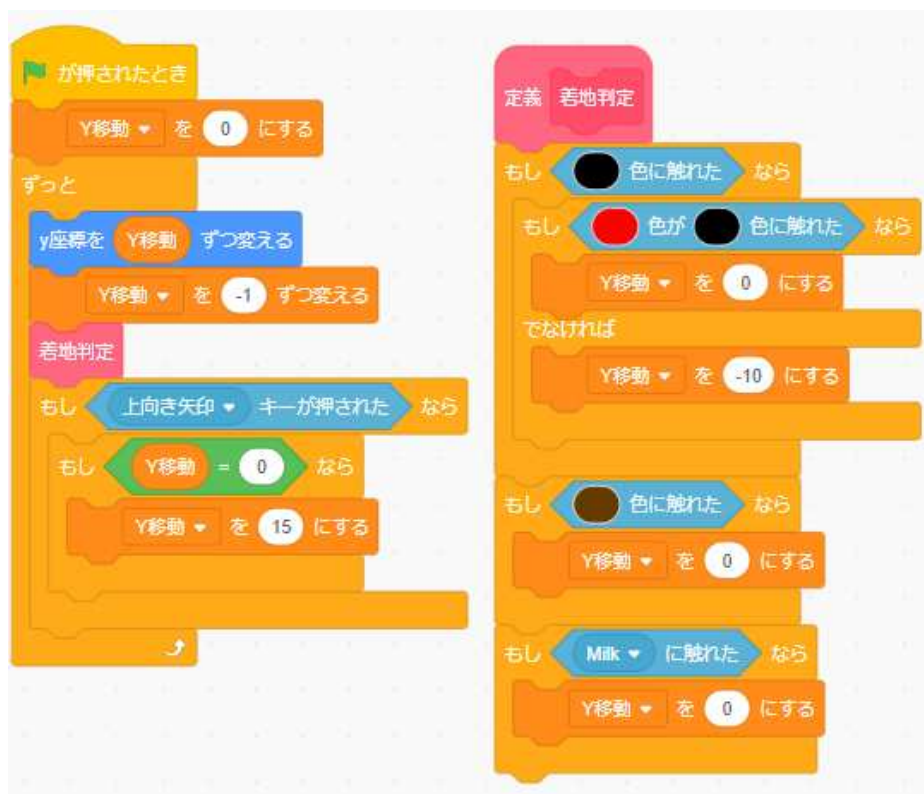
A screenshot of the Scratch 'New Variable' dialog box. The dialog has a blue header with the title '新しい変数' and a close button. Below the title, there is a text input field containing 'Y移動'. There are two radio buttons: the first is selected and labeled 'すべてのスプライト用', the second is labeled 'このスプライトのみ'. At the bottom, there is a checkbox for 'クラウド変数 (サーバーに保存)' and two buttons: 'キャンセル' and 'OK'. An arrow points from the '変数' block in the code area to this dialog box.

「Y 移動」の変数をつくります。

○ 横方向の移動とゴールのコードを作る。



○ 縦方向の移動のコードを作る。

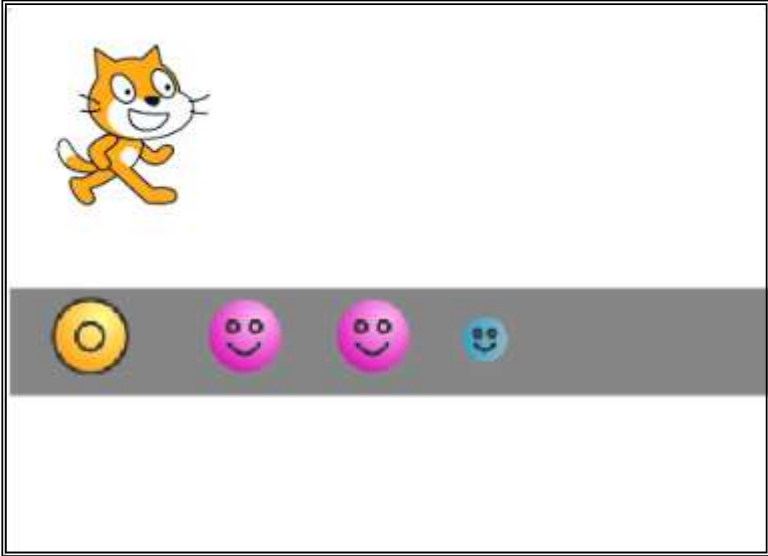


スクリプトが大きくなったので、ブロック定義(ピンクのブロック)を使って、スクリプトを分けてあります。

4. 改良・拡張してみよう。

- いろいろな階段の配置にしてみよう。階段を多くする場合はネコを小さくした方がいいよ。
- ゴールした時の効果を考えてみよう。
- 複数画面のゲームにしてみよう。
- 落ちたら死ぬ場所を考えてみよう。

太鼓ゲーム



みんなの大好きな太鼓のゲームを作ってみよう。

次の順番で作っていくよ。

1. 背景と音を出そう。
2. ドンの音符の sprites を作ろう。
3. 的の sprites を作ろう。
4. タイミングがあったらニャーと鳴くようにする。
5. カットの sprites を追加しよう。
6. 改良・拡張してみよう。



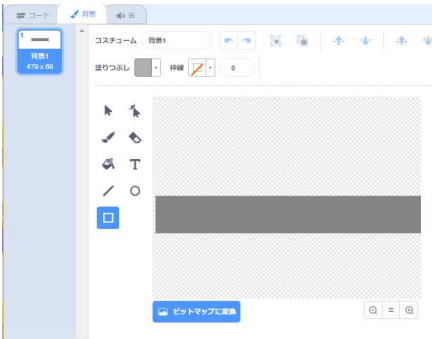
Go.Ota CoderDojo 市川真間

本資料は、CoderDojo 水戸の教材を参考にしています。

太鼓ゲーム[1]

1.背景と音を出そう。

○ 背景に音符の流れるレーンの追加しよう



○ 流す音を選ぼう



ループの中から好きな音を選ぼう

○音が流れるようにしよう

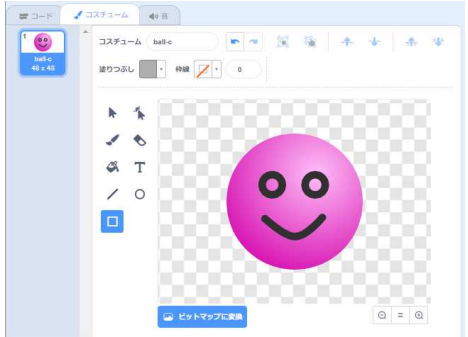
背景のコードを追加しよう。



旗をクリックすると音楽が流れるよ

太鼓ゲーム[2]

- 2. ドンの音符のSpriteを作ろう。
- ドンのコスチュームを作る。



[Spriteを選ぶ]か[描く]でSpriteを作ろう。

- ドンの動かそう/コードを追加しよう。



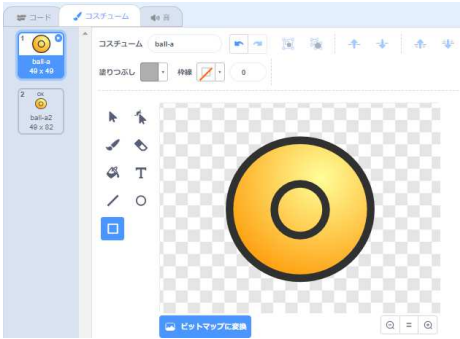
旗をクリックすると、ドンの音符が出てきて動くよ



ヒント: [端に触れた]はこのブロックを使います。

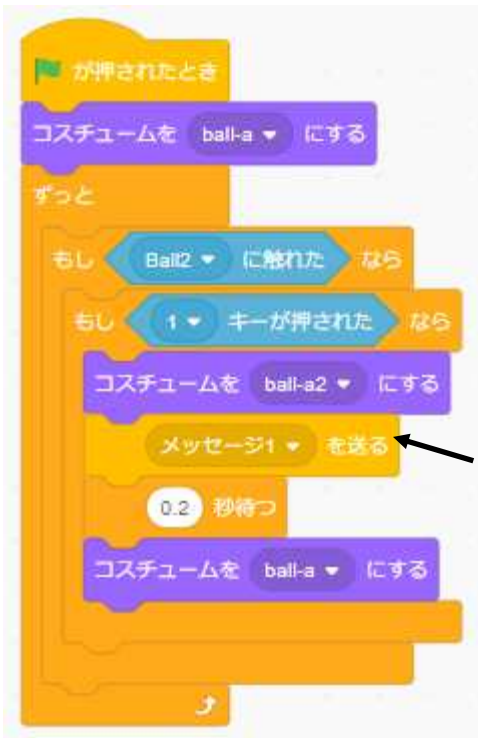
太鼓ゲーム[3]

- 3. 的のSpriteを作ろう。
- 的のコスチュームを作ろう



的のコスチュームデザインを作ります。タイミングが合った時に表示するコスチュームを作っておくよ。

○的のコードの追加/タイミングの判断

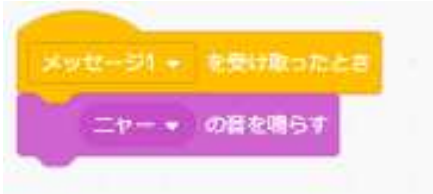


旗で動かして、音符が的のところで、1のキーを押すとののコスチュームが変わるよ。

次のネコに鳴かせる時に使います。

太鼓ゲーム[4]

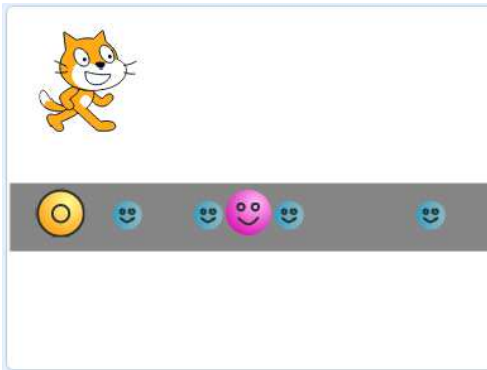
4. タイミングがあったらニャーと鳴くようにする。
○ネコに鳴くコードを追加する。



4. カットの sprites を追加するする。



ドンの sprites を [複製] して、カットの sprites を作ってみよう。



的のコードにカットのタイミングの判定を追加してみよう。例えば、カッは 2 のキーと押したときに判断するようにしてみよう

太鼓ゲーム[5]

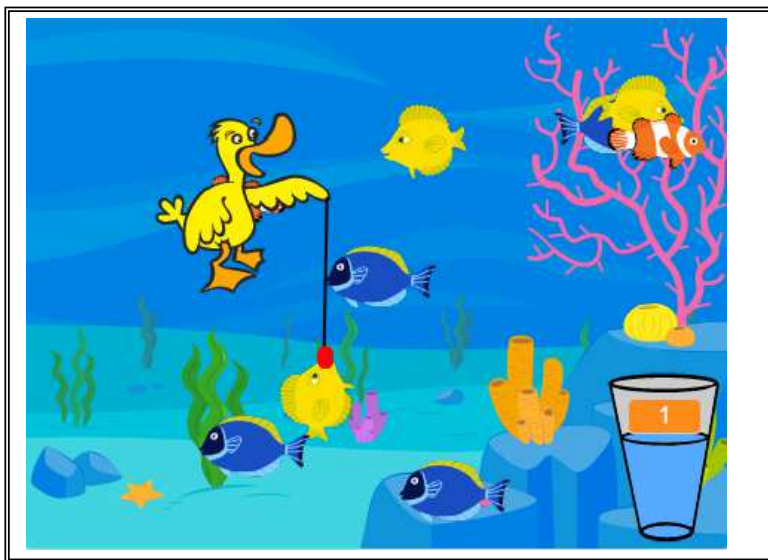
5. 改良・拡張してみよう。

- 成功した時の得点を表示してみよう。
- 何回か失敗したら、終了してみよう。
- タイミングが合った時の効果をもっと良くしてみよう。
- 音楽を自分で打ち込んでみよう。



拡張機能の[音楽]を使うよ。

さかな釣り



あひるが魚釣りをするよ。何匹つれるかな。

1. 魚を入れるコップの準備。
2. あひるを作る。
3. さかなを作る
4. 改良・拡張してみよう。



さかな釣り[1]

1. 魚を入れるコップの準備。。

- 背景を準備する。



背景に選択します。
赤色を使っていないものを選びます。

- コップのSpriteを選択する。。



- 釣った魚の数を数える変数の作成。

A screenshot of the '新しい変数' (New Variable) dialog box. The title bar is blue with the text '新しい変数' and a close button. The main area is white. It contains a text input field with the text '釣った数'. Below the input field, there are two radio buttons: 'すべてのスプライト用' (selected) and 'このスプライトのみ'. At the bottom, there is a checkbox for 'クラウド変数 (サーバーに保存)' and two buttons: 'キャンセル' and 'OK'.

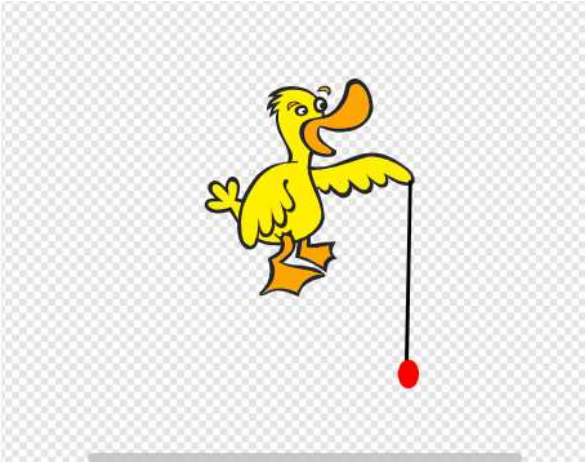
コップの位置に変数の表示を移動させます。

「釣った数」の変数をつくりま
す。

さかな釣り[2]

2. あひるを作る。

○ あひるのコスチュームを変更する。



アヒルのコスチュームに釣り糸と赤いエサを追加します。

この赤いえさは、さかなが釣れたかの判定に使用します。

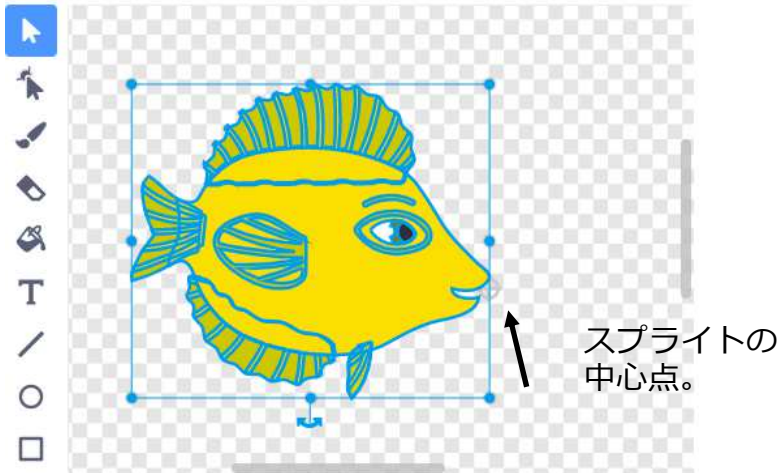
○ あひるのコードを追加する。



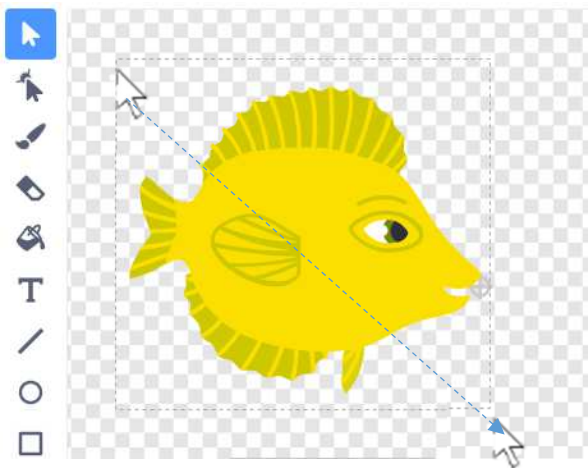
さかな釣り[2]

3. さかなを作る

- さかなのコスチュームを変更する。



スプライトの中心に口元がくるように、コスチューム全体を移動させます。4種類のさかながいるので、すべて移動させます。



ヒント:

移動させるとき選択を使って、コスチューム全体を囲むようにして指定します。

さかな釣り[2]

○ さかなのコードを作る

The image shows two Scratch code blocks for a fishing game. The left block is a script triggered by a click, and the right block is a custom function named '釣りチェック' (Fishing Check).

Left Block (Click Event):

- When clicked, set '釣った数' (Number Caught) to 0.
- Hide the sprite.
- Repeat 10 times:
 - Create a clone of the self.
- When a clone is created:
 - Set size to 60%.
 - Set rotation to '左右のみ' (Left and Right only).
 - Repeat 1 to 4 times:
 - Return to the next costume.
 - Go to a random location.
 - Turn 90 degrees.
 - Show the sprite.
 - Forever loop:
 - Move 5 steps.
 - If the edge is reached, rotate 2 to 2 degrees.
 - If the edge is reached, return to the start.

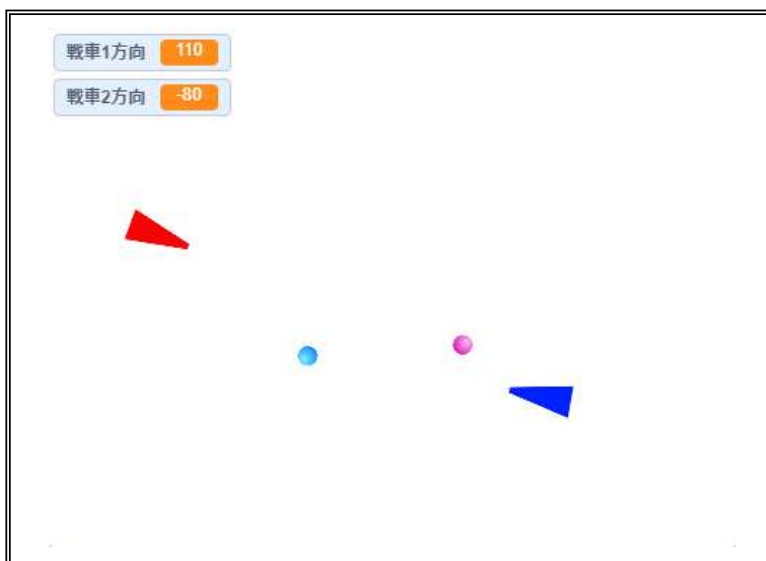
Right Block (Custom Function: 釣りチェック):

- If the red flag is touched:
 - If a random number between 1 and 100 is less than 2:
 - Set rotation to '自由に回転' (Free rotation).
 - Turn 0 degrees.
 - Forever loop:
 - Turn a random number between -5 and 5 degrees.
 - Set x-coordinate to Duck's x-coordinate + 55.
 - Set y-coordinate to Duck's y-coordinate - 105.
 - If 'Glass Water' is touched:
 - Increase '釣った数' (Number Caught) by 1.
 - Delete this clone.

4. 改良・拡張してみよう。

- 魚がされる確率を変えてみよう。釣った後に魚がにげられるようにしてみよう。
- 時間制限をつけてみよう。
- いろいろな魚を増やしてみよう。釣った魚で得点が変わるようにしてみよう。
- 魚以外のいろいろなものを捕まえるゲームにしてみよう。

戦車対戦



二人で戦車をあやって対戦しましょう。

1. 戦車を動かすようにします。
2. 玉を発射します。
3. 玉にあたった時のアクションを追加します。
4. 改良・拡張してみよう。



戦車対戦[1]

1. 戦車を動かすようにします。

○ 玉の発射方法を決める変数を準備する。

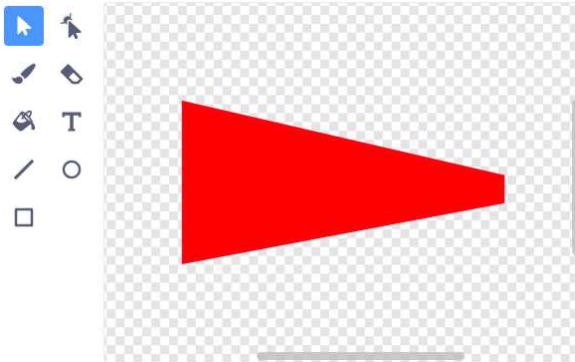


玉の発射方向を決める変数を準備します。

戦車 1 方向: 左側の赤い戦車用

戦車 2 方向: 右側の青い戦車用

○ 戦車のスプライトを作る。



左側の赤い戦車の
コスチュームを作
ります

ヒント:

2台の戦車はほとん
ど同じです。赤い戦
車を作ってから複製
して、青い戦車に
変更しましょう。

戦車対戦[2]

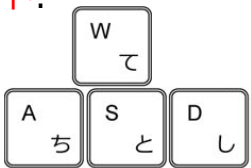
○ 戦車を動かすコードを作る。

左の赤い戦車(スプライト1) 右の青い戦車(スプライト2)

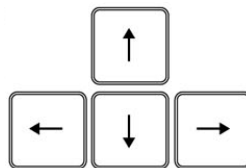
```
when green flag clicked
  set x to -200
  set y to 0
  rotate 90 degrees
  loop forever
    when key w is pressed
      move 5 steps
    when key d is pressed
      rotate 10 degrees clockwise
      set direction to right
    when key a is pressed
      rotate 10 degrees counter-clockwise
      set direction to left
```

```
when green flag clicked
  set x to 200
  set y to 0
  rotate -90 degrees
  loop forever
    when key up arrow is pressed
      move 5 steps
    when key right arrow is pressed
      rotate 10 degrees clockwise
      set direction to right
    when key left arrow is pressed
      rotate 10 degrees counter-clockwise
      set direction to left
```

ヒント:



Sで玉発射



下矢印で玉発射

戦車対戦[3]

3. 玉のコードを作る。

左の赤い戦車の玉
赤い玉のSprite

右の青い戦車の玉
青い玉のSprite

```
when clicked flag clicked
  set size to 30%
  hide
  forever loop
    if key pressed s
      create clone of self
      wait 0.5 seconds
```

```
when clicked flag clicked
  set size to 30%
  hide
  forever loop
    if key pressed down arrow
      create clone of self
      wait 0.5 seconds
```

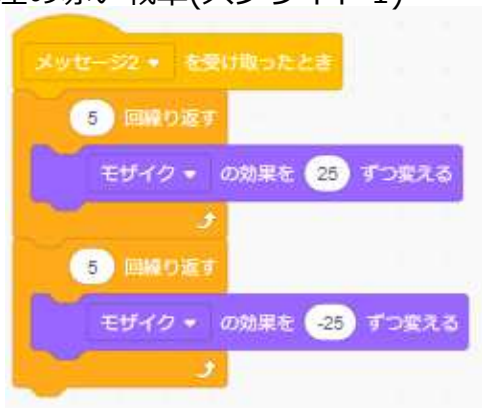
```
when cloned
  sprite1 move
  turn 90 degrees
  show
  forever loop
    10 steps
    if touched sprite2
      say message1
      delete clone
    if touched end
      delete clone
```

```
when cloned
  sprite2 move
  turn 90 degrees
  show
  forever loop
    10 steps
    if touched sprite1
      say message2
      delete clone
    if touched end
      delete clone
```

戦車対戦[3]

3. 玉にあたった時のアクションを追加します。

左の赤い戦車(スプライト 1)



右の青い戦車(スプライト 2)



4. 改良・拡張してみよう。

○ いろいろな形の戦車を作ってみよう。

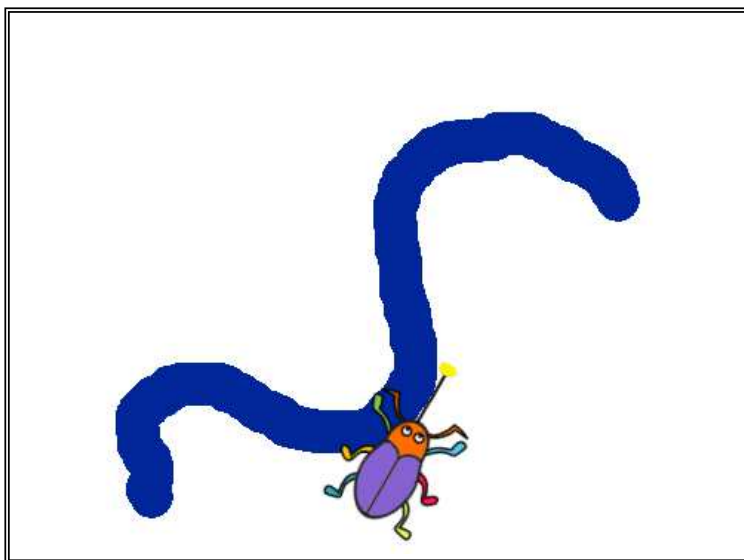
○ 得点をつけてみよう。

○ 勝利条件をつけてみよう。

○ 防御や攻撃方法を考えてみよう。

○ 片方をコンピュータで対戦できるようにしてみよう。

ライトレーサー



線にそって、虫がピコピコ動くよ。

次の順番で作っていくよ。

1. 背景を作ろう。
2. 虫のコスチュームを変更しよう。
3. 虫のコードを作ろう。
4. 改良・拡張してみよう。



ライトレーサー[1]

1. 背景を作ろう。



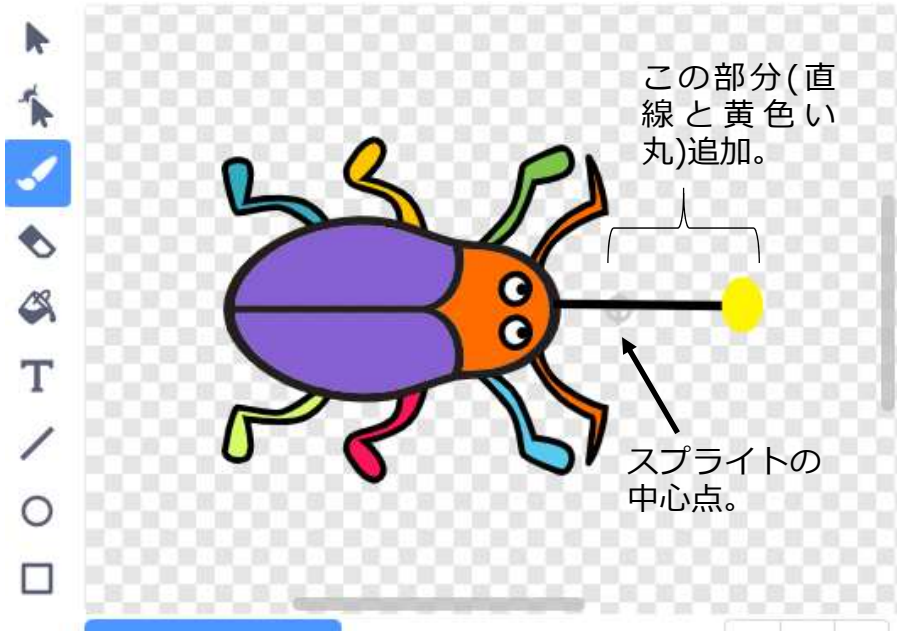
虫が歩く太めの線を引きます。

ヒント:

ビットマップの方が描きやすいかも。

ライトレーサー[2]

2. 虫のコスチュームを変更しよう。



虫のスプライトに線を検知するための触覚(直線と黄色い丸を追加します)

おへそみたいな示しのスプライトの中心点が、この図に位置になるように調整します。

ヒント:

うまく動かない場合は、この中心点の位置を調整します。

ライトレーサー[3]

3. 虫のコードを作ろう。



ヒント:

[XX色に触れた]の色指定については、スポイトで吸い取ると正しく指定できます。

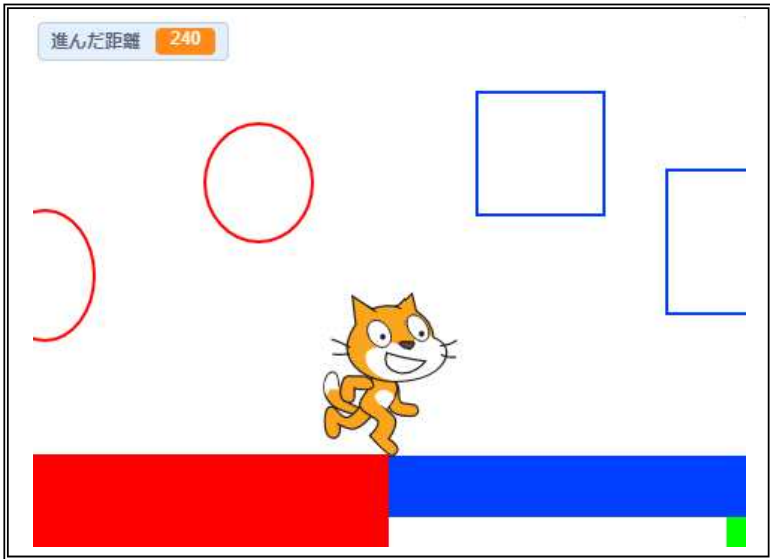
5. 改良・拡張してみよう。

○ いろいろな線をトレースさせてみよう。

○ いろいろなものでトレースしてみよう。

○ トレースの動きを変化させよう。

横スクロール基礎



ゲームでよくある背景や地面が動く横スクロールの基礎的なプログラムです。内容を理解してどんどん応用してみてね。

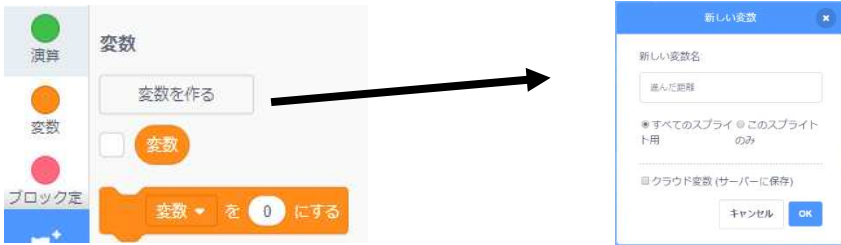
1. ネコのコードを作る。
2. スクロール用のスプライトを準備する。
3. スクロール用のコードを作る
4. 改良・拡張してみよう。



横スクロール基礎[1]

1. ネコのコードを作る。

○ 移動した距離を入れる変数を準備する。



横スクロールの場合は、ネコの画面上の位置は固定され、背景や地面となるスプライトが動きます。これらのスプライトは、この[進んだ距離]の値をもとに表示位置を変えていきます。

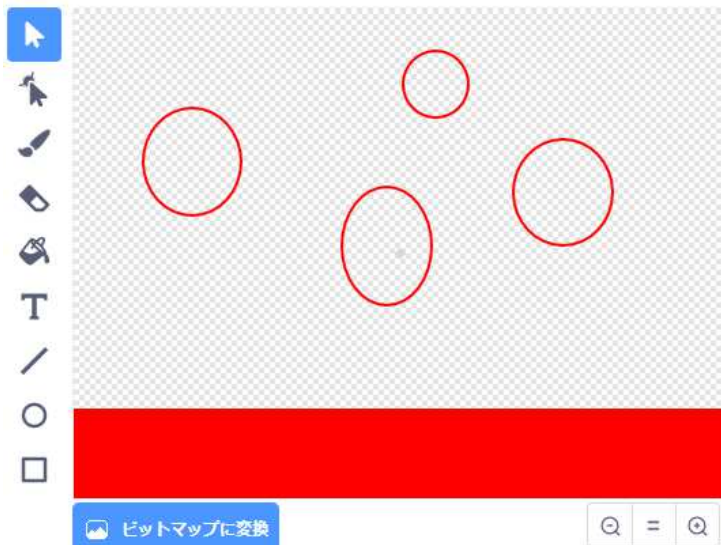
○ ネコのコード。



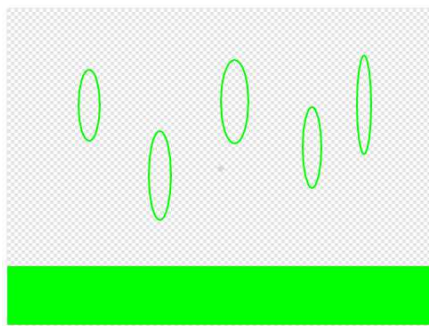
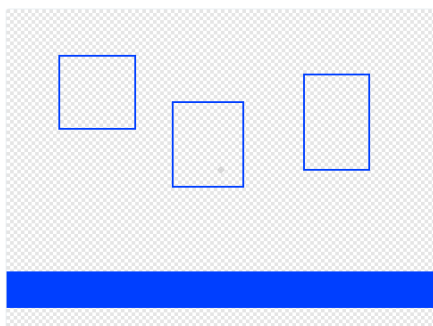
横スクロール基礎[2]

2. スクロール用のスプライトを準備する。

赤、青、緑の三個のスプライトを作っていきます。



地面になる部分は、コスチュームの作成画面いっぱいに書きます。背景になる部分には適当な模様を描いておきます。同様に、新しく青と緑のスプライトもつくります。



横スクロール基礎[3]

3. スクロール用のコードを作る

赤のSprite



青のSprite



緑のSprite



ヒント:

それぞれのコードでSpriteのX座標の位置が異なります。

4. 改良・拡張してみよう。

- いろいろな背景を作ってみよう。
- たくさんの背景を作ってみよう。
- 複数のSpriteを組み合わせてみよう。

計算機



数を入れていくと、最後にネコが合計を計算してくれるよ。
次の順番で作っていくよ。

1. 変数・リストを準備する。
2. ネコのコードを作る。
3. 改良・拡張してみよう。



計算機[1]

1. 変数・リストを準備する。
 - 入力した数を記録するリストを作る。



- 変数を作る。



合計の結果をいれる[合計]と、どの数を足すか決める[カウンター]の変数を作成します。

計算機[2]

2. ネコのコードを作る。



計算機[5]

4. 改良・拡張してみよう。

- テンキーで数字を入力できるようにしてみよう。
- 四則演算ができる電卓を作ってみよう。