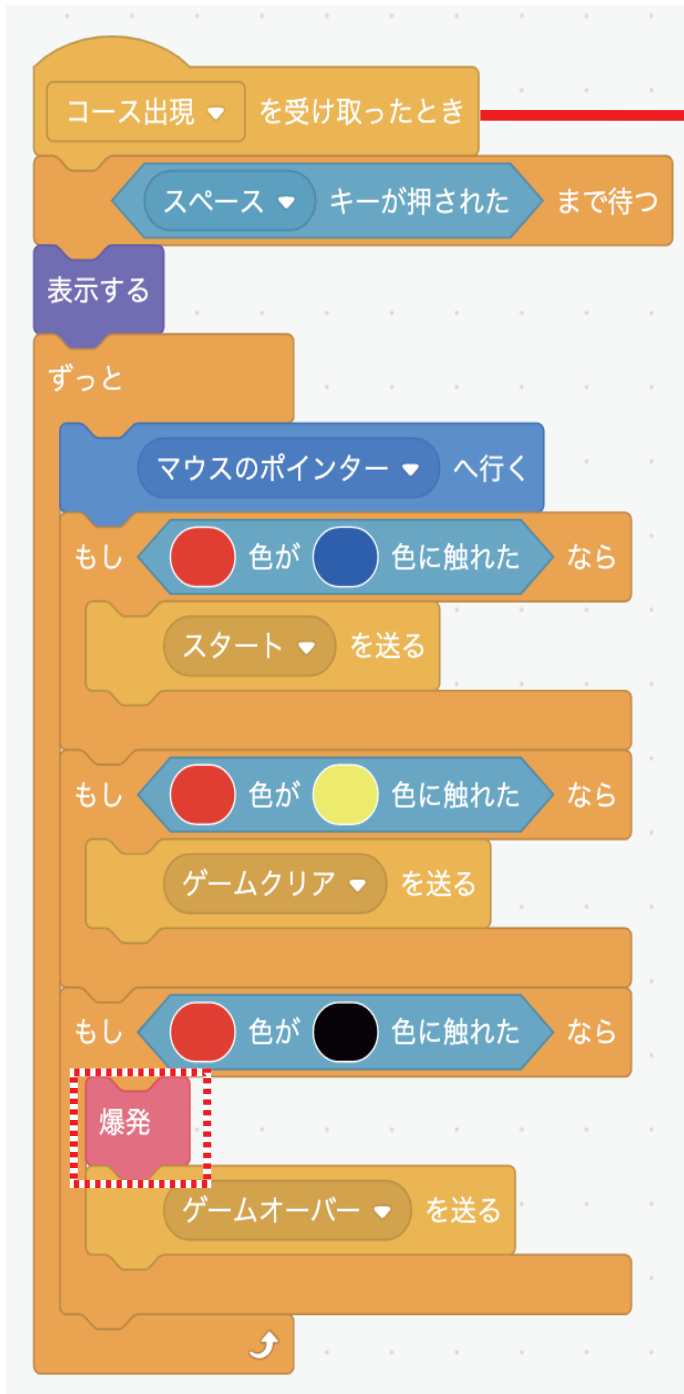
の部分を追加して、「爆弾」が「ゴールエリア」についたかを判断します。



The script starts with a 'Course Appearance' message block. It then waits for the space key to be pressed. A 'Show' block displays 'Bomb'. A 'Forever' loop contains three 'If mouse pointer touches color' blocks. The first checks for red and sends a 'Start' message. The second checks for yellow and sends a 'Game Clear' message. The third checks for black and sends a 'Game Over' message. A 'Explosion' block is added to the end of the loop, which sends a 'Game Over' message.

“コース出現”メッセージが送られてきたとき

- ・スペースキーを押すまで待ちます。
- ・表示します。

・ずっと くりかえし ます。

・爆弾が、マウスポインターの位置にいきます。

<もし、爆弾のふちがスタートエリアのラインにふれたら>

・“スタート”メッセージをプログラム全体に送ります。

<もし、爆弾のふちがゴールエリアにふれたら>

・“ゲームクリア”メッセージをプログラム全体に送ります。

<もし、爆弾のふちがコースにふれたら>

・ブロック定義爆発を処理します。

・“ゲームオーバー”メッセージをプログラム全体に送ります。

確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

・スタート後、スタートエリアでスペースキーを押して「爆弾」を表示させます。

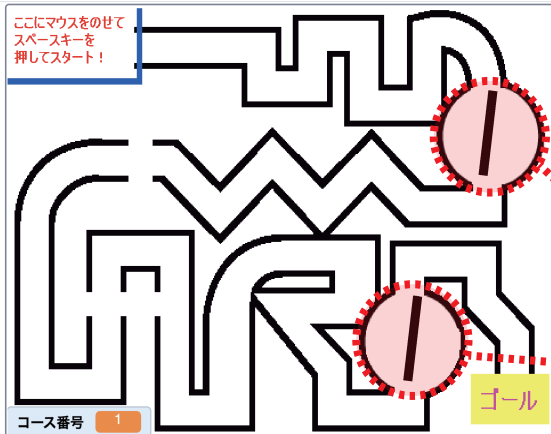
□「爆弾」がコースにふれたとき、爆発しましたか？

4 「回転板」を作ろう

1 「回転板」のプログラムを作ります



① 「回転板」の説明です。



下記の座標を中心として回転して、爆弾の邪魔をします。

回転板名	x 座標	y 座標
回転板 1	182	73
回転板 2	116	-98

② 「回転板」の基本の動きを“回転板 1”で作ります。



ゲームがスタートしたとき

- ・かくします。

“コース出現”メッセージが送られてきたとき

- ・位置を $(x, y) = (182, 73)$ にします。
- ・表示させます。
- ・ずっと くりかえし ます。
- ・0.3 秒待ちます。
- ・右回りで 15° 回します。

確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

- ・スタートボタンを押して、ゲーム実行します。
- 「回転板」は、“回転板 1”の位置に表示されましたか？
- 「回転板」は、回転していますか？

③ 「回転板」のクローンを使い、「回転板1」を作ります。

②で作ったプログラム

ゲームがスタートしたとき

- ・本体は、かくしたままにします。

“回転板1”の位置だけ指示し、クローンを作る

“回転板1”の動きは、クローンされてから指示する。

“コース出現”メッセージが送られてきたとき

- ・位置を $(x,y) = (182,73)$ にします。
- ・“自分自身”のクローンを作ります。

クローンが作られたとき

- ・クローンを表示します。
- ・ずっと、繰り返します。
 - ・0.3秒待ちます。
 - ・右回りで15°回します。

☑確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

- ・スタートボタンを押して、ゲーム実行します。
- 「回転板」は、“回転板1”の位置に表示されましたか？
- 「回転板」は、回転していますか？

④ 次に、を追加して、“回転板 2” を作ります。



コース出現 ▼ を受け取ったとき

x座標を 182 、y座標を 73 にする

自分自身 ▼ のクローンを作る

x座標を 116 、y座標を -98 にする

自分自身 ▼ のクローンを作る

“コース出現” メッセージが送られてきたとき

- ・位置を $(x,y) = (182,73)$ にします。
- ・“自分自身” のクローンを作ります。
- ・位置を $(x,y) = (116,-98)$ にします。
- ・“自分自身” のクローンを作ります。

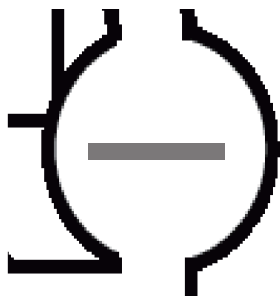
☑確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

- ・スタートボタンを押して、ゲーム実行します。
- 「回転板」は、“回転板 2” の位置に表示されましたか？
- 「回転板」は、回転していますか？

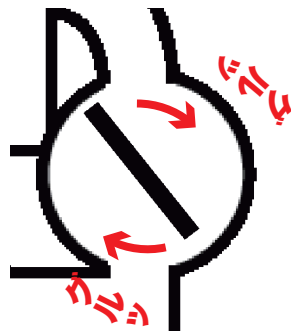
Scratch の説明：
クローンとは

- ・同じスプライトを画面上で複数動かすときに使います。
- ・ローカル変数やメッセージを受け取ったときの処理もコピーされるので、それを含めて設計する必要があります。

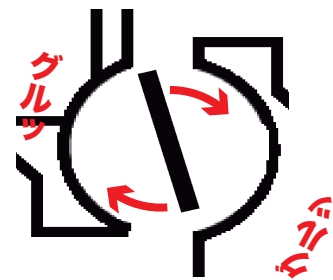


実体（本物）：
隠れて何もしていない

クローン1（回転板1）：
実際に表示され、動いている



クローン2（回転板2）：
実際に表示され、動いている



- ・クローンを使う場合は、本体は使いません。
(使わない方がプログラムが簡単になるからです。)

- ⑤ ④で作ったプログラムをもっと読みやすく、あとで回転板が3、4、と増やしやすくするために、ブロック定義「回転板を動かす」を作成して、置き換えます。

②で作ったプログラム

```

    コース出現 ▼ を受け取ったとき
    x座標を 182 、 y座標を 73 にする
    自分自身 ▼ のクローンを作る
    x座標を 116 、 y座標を -98 にする
    自分自身 ▼ のクローンを作る
  
```

置き換え

```

    コース出現 ▼ を受け取ったとき
    回転板を動かす 182 73
    回転板を動かす 116 -98

    定義 回転板を動かす x座標 y座標
    x座標を x座標 、 y座標を y座標 にする
    自分自身 ▼ のクローンを作る
  
```

“コース出現”メッセージが送られてきたとき

- ・ 回転板を動かすブロックを、
(x座標 = 182)(y座標 = 73) で処理します。
- ・ 回転板を動かすブロックを、
(x座標 = 116)(y座標 = -98) で処理します。

“回転板を動かす”ブロックを実行したときの動き (x座標)(y座標)

- ・ 位置を (x,y)=(“x座標” , “y座標”) にします。
- ・ 自分自身のクローンを作ります。

☑確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

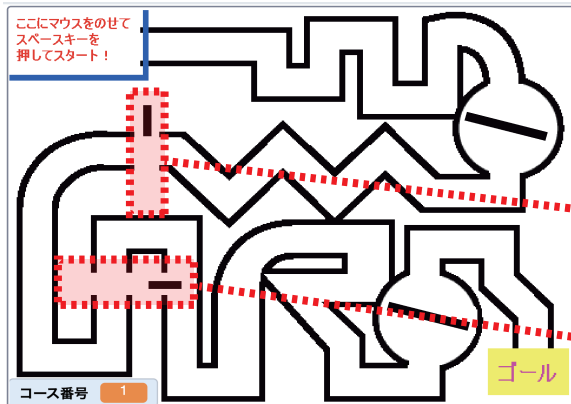
- ・ スタートボタンを押して、ゲーム実行します。
- 「回転板」は、“回転板1”と“回転板2”の位置に表示されましたか？
- どちらの「回転板」も、回転していますか？

5 「よこぎり板」を作ろう

1 「よこぎり板」のプログラムを作ります



① 「よこぎり板」の説明です。



下記の座標と方向で、±40 の範囲で動き、コースをよこぎります。

回転板名	x 座標	y 座標	方向
よこぎり板1	-122	50	上下
よこぎり板2	-135	-70	左右

② 「よこぎり板」の基本の動きを“よこぎり板1”で作ります。



ゲームがスタートしたとき

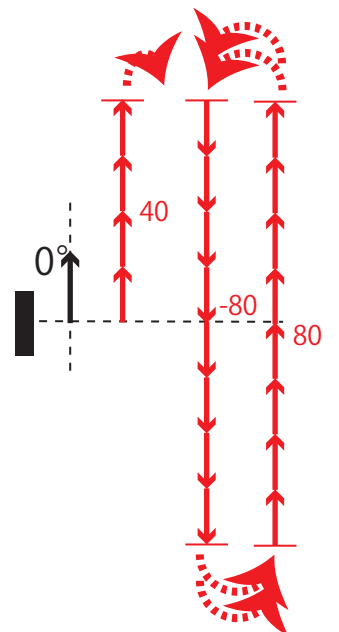
- ・かくします。

“コース出現”メッセージが送られてきたとき

- ・位置を $(x, y) = (-122, 50)$ にします。
- ・ 0° の方向に向けて動くようにセットします。
- ・表示します。

・40回繰り返します。
・1歩動きます。

・ずっと繰り返します。
・80回繰り返します。
・-1歩動きます。
・80回繰り返します。
・1歩動きます。



確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

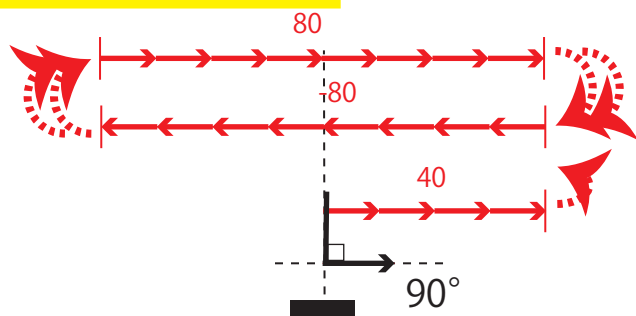
- ・スタートボタンを押して、ゲーム実行します。
- 「よこぎり板」は、「よこぎり板1」の位置に表示されましたか？
- 「よこぎり板」は、縦に動いてコースをよこぎりますか？
- 「よこぎり板」は、縦に動いてコースをよこぎったあと帰ってきますか？

③ 次に、別で「よこぎり板」の基本の動きを“よこぎり板2”でも作ってみます。



“コース出現”メッセージが送られてきたとき

- ・位置を $(x, y) = (-135, -70)$ にします。
- ・90°の方向に向けて動くようにセットします。
- ・表示します。
- ・40回繰り返します。
 - ・1歩動きます。
- ・ずっと繰り返します。
- ・80回繰り返します。
 - ・-1歩動きます。
- ・80回繰り返します。
 - ・1歩動きます。



確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

- ・③で作ったプログラムだけ実行します。
- 「よこぎり板」は、「よこぎり板2」の位置に表示されましたか？
- 「よこぎり板」は、横に動いてコースをよこぎりますか？
- 「よこぎり板」は、横に動いてコースをよこぎったあと帰ってきますか？



④ ②の“よこぎり板1”と③の“よこぎり板2”のプログラムは、同時に動かしたいので、クローンで動かすように変更します。

②で作った“よこぎり板1”

このScratchコードは、コース出現を受け取ったときに実行される。最初のブロックは「x座標を -135、y座標を -70 にする」と「90 度に向ける」で、この部分が「位置と向き」として赤い点線で囲まれている。続いて「表示する」ブロックがあり、その下に「40 回繰り返す」のループがあり、「1 歩動かす」ブロックが繰り返される。さらに「ずっと」のループがあり、「80 回繰り返す」のループがあり、「-1 歩動かす」ブロックが繰り返され、その後「80 回繰り返す」のループがあり、「1 歩動かす」ブロックが繰り返される。この「ずっと」ループの部分が「共通の部分」として青い点線で囲まれている。

③で作った“よこぎり板2”

このScratchコードは、コース出現を受け取ったときに実行される。最初のブロックは「x座標を -122、y座標を 50 にする」と「0 度に向ける」で、この部分が「位置と向き」として緑色の点線で囲まれている。続いて「表示する」ブロックがあり、その下に「40 回繰り返す」のループがあり、「1 歩動かす」ブロックが繰り返される。さらに「ずっと」のループがあり、「80 回繰り返す」のループがあり、「-1 歩動かす」ブロックが繰り返され、その後「80 回繰り返す」のループがあり、「1 歩動かす」ブロックが繰り返される。この「ずっと」ループの部分が「共通の部分」として青い点線で囲まれている。


```

    コース出現 を受け取ったとき
    x座標を -122 、y座標を 50 にする
    0 度に向ける
    自分自身 のクローンを作る
    x座標を -135 、y座標を -70 にする
    90 度に向ける
    自分自身 のクローンを作る

    クローンされたとき
    表示する
    40 回繰り返す
    1 歩動かす
    ずっと
    80 回繰り返す
    -1 歩動かす
    80 回繰り返す
    1 歩動かす
  
```

“コース出現”メッセージが送られてきたとき

- ・位置を $(x, y) = (-122, 50)$ にします。
- ・ 0° の方向に向けて動くようにセットします。
- ・自分自身のクローンを作ります。
- ・位置を $(x, y) = (-135, -70)$ にします。
- ・ 90° の方向に向けて動くようにセットします。
- ・自分自身のクローンを作ります。

クローンが作られたとき

- ・表示します。
- ・40回繰り返します。
 - ・1歩動きます。
- ・ずっと繰り返します。
 - ・80回繰り返します。
 - ・-1歩動きます。
 - ・80回繰り返します。
 - ・1歩動きます。

確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□にをつけよう。

・スタートボタンを押して、ゲーム実行します。

“よこぎり板1”は、先ほどと同じように、正しく動いていますか？

“よこぎり板2”は、先ほどと同じように、正しく動いていますか？

⑤ ④で作ったプログラムをブロック定義「よこぎり板を動かす」で置き換えます。

④で作ったプログラム



```
コース出現 ▾ を受け取ったとき
x座標を -122 、y座標を 50 にする
0 度に向ける
自分自身 ▾ のクローンを作る
x座標を -135 、y座標を -70 にする
90 度に向ける
自分自身 ▾ のクローンを作る
```

置き換え



コース出現 ▾ を受け取ったとき

よこぎり板を動かす -120 50 0

よこぎり板を動かす -135 -70 90

“コース出現”メッセージが送られてきたとき

- ・よこぎり板を動かすブロックを、
(x座標=-120)(y座標=50)(角度=0)で処理します。
- ・よこぎり板を動かすブロックを、
(x座標=-135)(y座標=-70)(角度=90)で処理します。

定義 よこぎり板を動かす x座標 y座標 角度

x座標を x座標、y座標を y座標 にする

角度 度に向ける

自分自身 ▾ のクローンを作る

“よこぎり板を動かす”ブロックを実行したときの動き
(x座標)(y座標)(角度)

- ・位置を(x,y)=(“x座標”，“y座標”)にします。
- ・自分自身のクローンを作ります。

☑確認してみよう！

本項で作成したプログラムを実行して、動きを確認しよう。確認できたら□に☑をつけよう。

- ・スタートボタンを押して、ゲーム実行します。
- “よこぎり板1”は、先ほどと同じように、正しく動いていますか？
- “よこぎり板2”は、先ほどと同じように、正しく動いていますか？

6 コースの流れを作ろう

1 ステージのプログラムを作ります

いまできているプログラムを変更して、ステージでは、トップ画面とコース画面の背景を表示するのみにします。



変更



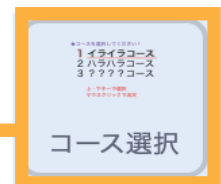
ゲームがスタートしたとき

- ・背景を“タイトル画面”にします。
- ・マウスがクリックされるのを待ちます。
- ・0.3秒待ちます。
- ・背景を“コース選択背景”にします。
- ・“コース選択”メッセージを送ります。

2 「コース選択」のプログラムを作ります

「コース選択」では、コースの種類を表示して選択できるようにします。

① スタート時は、表示を初期設定して、コース番号を1にしておきます。



ゲームがスタートしたとき

- ・かくします。
- ・初期設定として、コース番号を1にします。