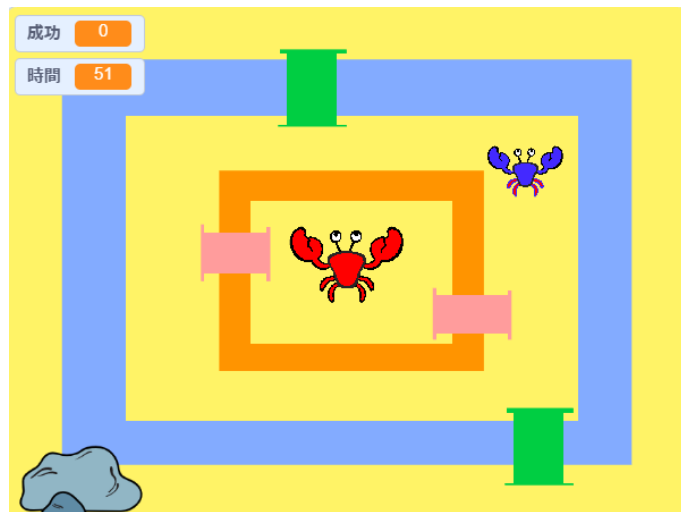


# カニ友レスキュー



## スプライトを用意しよう

自分で描いてもライブラリから持ってきてもOKです



【青カニ】  
カニ友を助けに行く



【カニ友】  
青カニに助けをもらう



【外堀】  
落ちるとアウト



【内堀】  
落ちるとアウト



【外の橋1】  
ランダム出現



【外の橋2】  
ランダム出現



【内の橋1】  
ランダム出現



【内の橋2】  
ランダム出現



【岩屋】  
カニ友をここまで連れてくると成功  
(ポイントゲット)

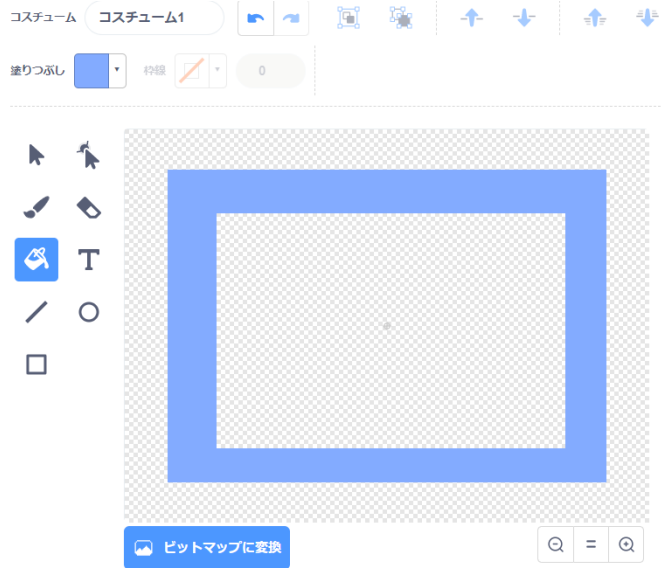
### ○こんなゲームです

- ▶ 青カニが、お堀の中にとらわれているカニ友を助け出して、岩屋まで無事に届けるゲームです
- ▶ お堀は二重になっていて、落ちるとスタート地点に戻ってしまいます
- ▶ 橋はランダム出現するので手ごわいです
- ▶ 制限時間内で何匹のカニ友をレスキューできるかチャレンジしてください

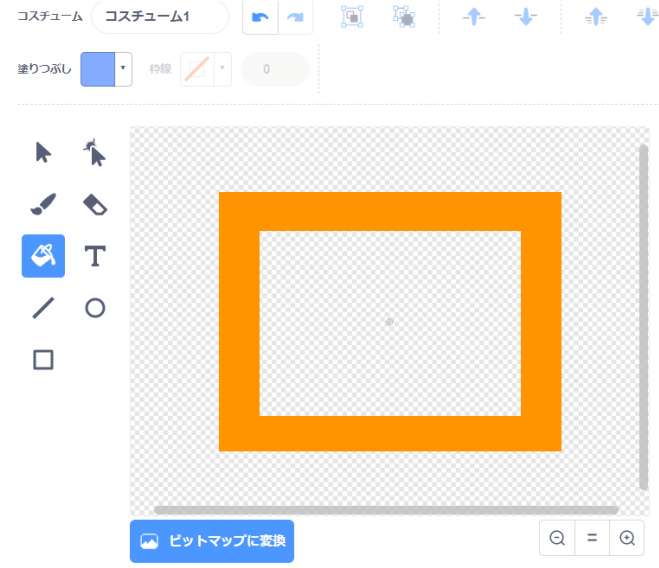
# 堀を作る



## ○スプライトで外堀を描きます



## ○スプライトで内堀を描きます



中心を画面の真ん中にします

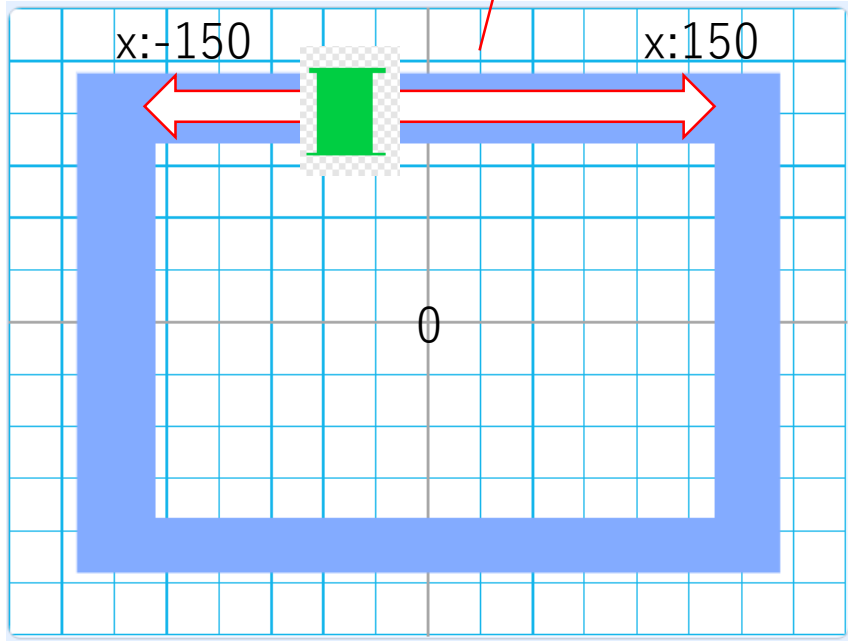


# 外の橋1のプログラム



○ タテは上に固定して  
左右を乱数にする

この範囲に「外の橋1」  
を出現させたい



出現位置は、

x座標を **-150** から **150** までの乱数、y座標を **125** にする

ヨコ (x座標が [-150~150] のどこか)  
タテ (y座標が [125] あたり)

数字は描いた  
絵によって  
異なります

```
が押されたとき
隠す
ずっと
  x座標を [ ]、y座標を [ ] にする
  [-150 から 150 までの乱数]
  1 秒待つ
  表示する
  3 秒待つ
  隠す
```

```
が押されたとき
隠す
ずっと
  x座標を [-150 から 150 までの乱数]、y座標を 125 にする
  1 秒待つ
  表示する
  3 秒待つ
  隠す
```

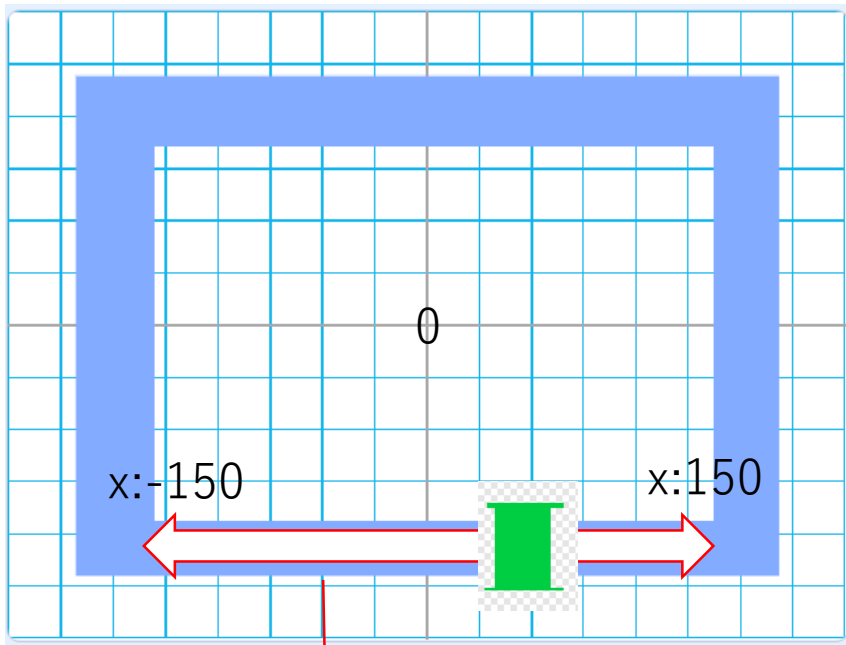
1秒おきに3秒出現

左右 (ヨコ) はx座標、上下 (タテ) はy座標って言います

## 外の橋2のプログラム



○ タテは下に固定して左右を乱数にする



この範囲に「外の橋2」  
を出現させたい

左右（ヨコ）はx座標、上下（タテ）はy座標って言います

数字は描いた  
絵によって  
異なります

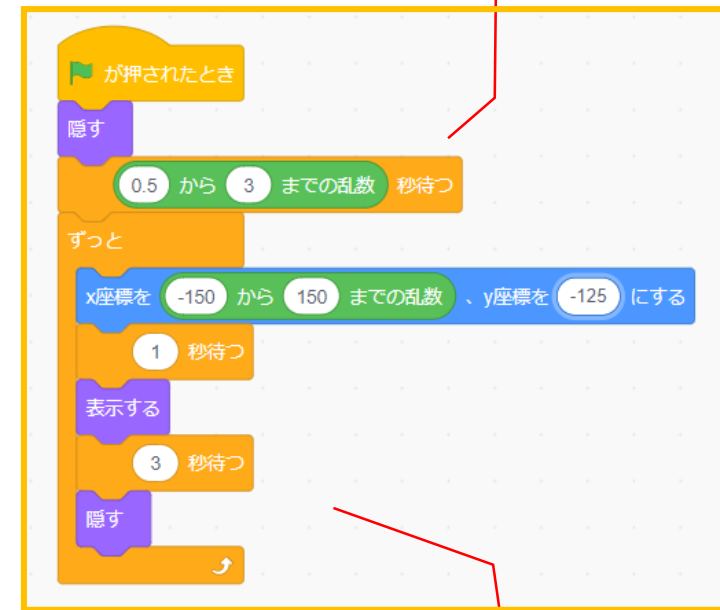
出現位置は、

x座標を  から  までの乱数、y座標を  にする

ヨコ（x座標が [-150~150] のどこか）  
タテ（y座標が [-125] あたり）



外の橋1と出現の  
タイミングをずらす



1秒おきに3秒出現

# 内の橋1のプログラム



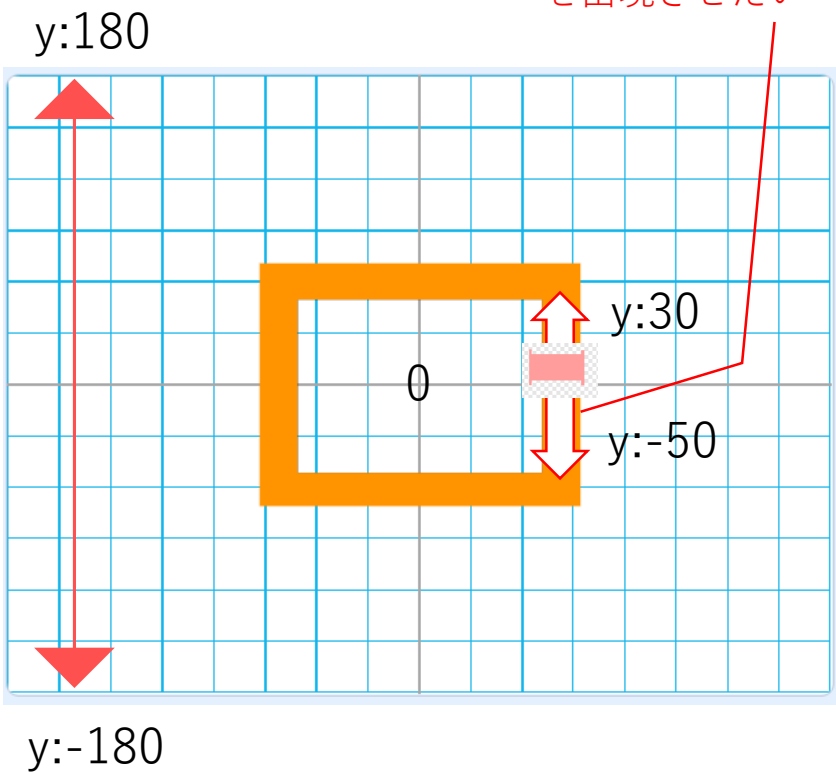
数字は描いた  
絵によって  
異なります

出現位置は、

x座標を 80、y座標を 30 から -50 までの乱数 にする

○ ヨコは右に固定して  
上下を乱数にする

この範囲に「内の橋1」  
を出現させたい



ヨコ (x座標が [80] あたり)  
タテ (y座標が [-50~30] のどこか)

外の橋1と出現の  
タイミングをずらす

```
が押されたとき
隠す
秒待つ (0.5 から 3 までの乱数)
ずっと
  x座標を [ ]、y座標を [ ] にする
  (30 から -50 までの乱数)
  1 秒待つ
  表示する
  3 秒待つ
  隠す
```

```
が押されたとき
隠す
(0.5 から 3 までの乱数) 秒待つ
ずっと
  x座標を 80、y座標を 30 から -50 までの乱数 にする
  1 秒待つ
  表示する
  3 秒待つ
  隠す
```

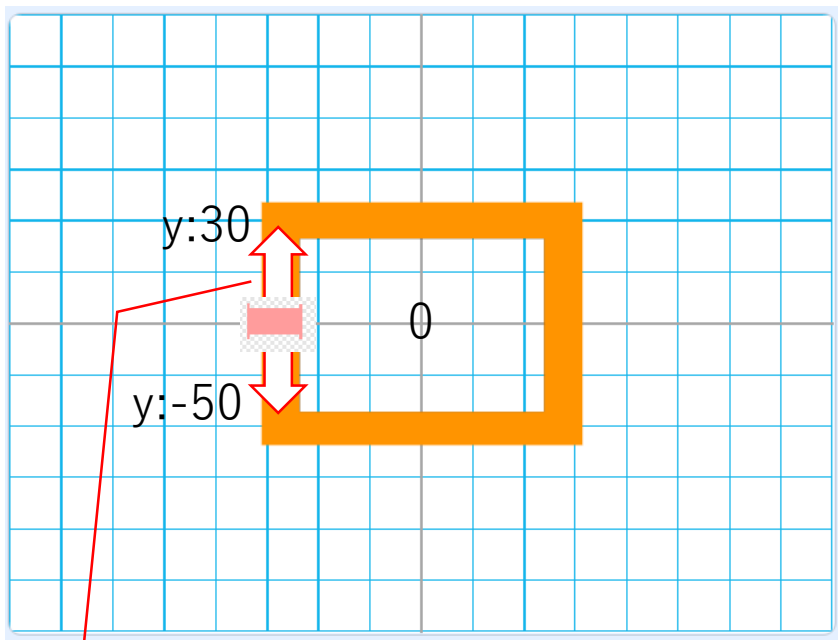
1秒おきに3秒出現

左右 (ヨコ) はx座標、上下 (タテ) はy座標って言います

## 内の橋2のプログラム



○ ヨコは左に固定して上下を乱数にする



この範囲に「内の橋2」  
を出現させたい

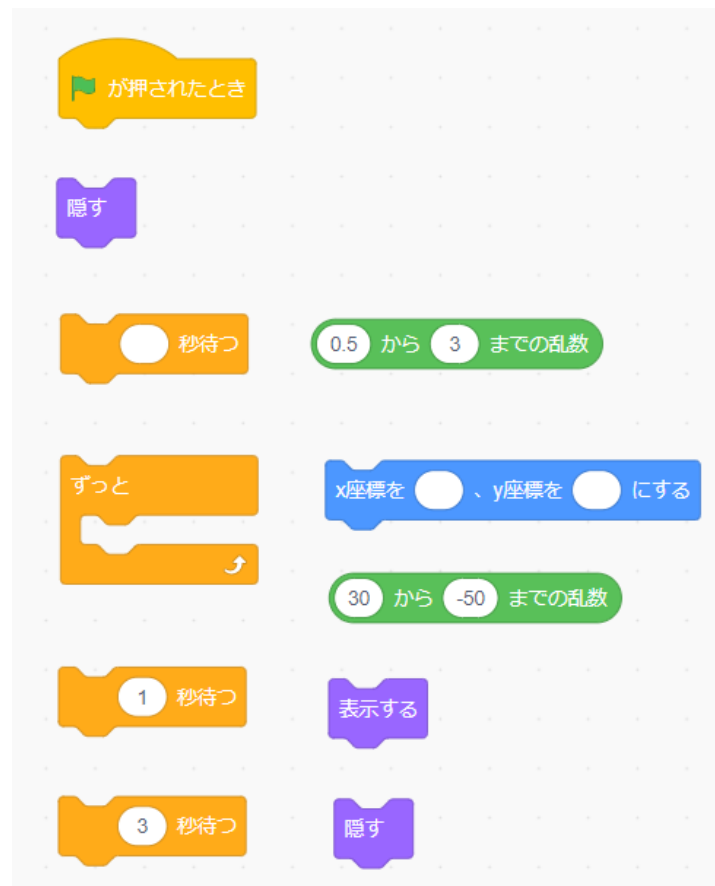
数字は描いた  
絵によって  
異なります

出現位置は、

x座標を -80 、y座標を 30 から -50 までの乱数 にする

ヨコ (x座標が [80] あたり)  
タテ (y座標が [-50~30] のどこか)

外の橋1と出現の  
タイミングをずらす



1秒おきに3秒出現

左右 (ヨコ) はx座標、上下 (タテ) はy座標って言います

## 青カニのプログラム ①



### タイプA

出現位置

○「上下左右に動かす」  
のコードは？（復習です）

タイプAは簡単、タイプBは  
ちょっと複雑。どっちもOK

```
が押されたとき
x座標を 210 、 y座標を 160 にする
ずっと
もし 上向き矢印 ▼ キーが押された なら
  y座標を 3 ずつ変える
もし 下向き矢印 ▼ キーが押された なら
  y座標を -3 ずつ変える
もし 右向き矢印 ▼ キーが押された なら
  x座標を 3 ずつ変える
もし 左向き矢印 ▼ キーが押された なら
  x座標を -3 ずつ変える
```

### タイプB

```
が押されたとき
x座標を 210 、 y座標を 160 にする
ずっと
  上向き矢印 ▼ キーが押された ではない まで繰り返す
  y座標を 3 ずつ変える
  下向き矢印 ▼ キーが押された ではない まで繰り返す
  y座標を -3 ずつ変える
  右向き矢印 ▼ キーが押された ではない まで繰り返す
  x座標を 3 ずつ変える
  左向き矢印 ▼ キーが押された ではない まで繰り返す
  x座標を -3 ずつ変える
```

## 青カニのプログラム ②



○ 「堀に落ちたらスタートに戻る」のコードは？

【新しいメッセージ】  
で作ります

Scratch code blocks for the initial program:

- が押されたとき (When green flag clicked)
- ずっと (Forever loop)
- もし (If) なら (then)
- 失敗 を送る (Send failure message)
- 隠す (Hide)
- x座標を 210、y座標を 160 にする (Set x and y coordinates)
- 1.5 秒待つ (Wait 1.5 seconds)
- 表示する (Show)

Logic blocks:

- または (or)
- 色に触れた (Touched color)

右と左とどちらかが  
正解ならOK

(スプライトで衝突判定)

Scratch logic block: 外堀 に触れた または 内堀 に触れた

「堀に落ちたら」は外堀と  
内堀の色で判定します

でもOKです

Scratch code blocks for the revised program:

- が押されたとき (When green flag clicked)
- ずっと (Forever loop)
- もし (If) なら (then)
- 失敗 を送る (Send failure message)
- 隠す (Hide)
- 1.5 秒待つ (Wait 1.5 seconds)
- x座標を 210、y座標を 160 にする (Set x and y coordinates)
- 表示する (Show)

Logic block: 色に触れた または 色に触れた

「失敗」メッセージを受け取る相手は誰でしょう？



# カニ友のプログラム



○青カニにタッチされると  
青カニと合体します

Scratch code blocks for the initial state of the crab:

- が押されたとき (When clicked)
- 大きさを 60 %にする (Set size to 60%)
- x座標を 0 、y座標を -20 にする (Set x to 0, y to -20)
- ずっと (Forever loop) containing:
  - もし (If) containing:
    - 青カニ に触れた (Touched blue crab)
    - 大きさを 25 %にする (Set size to 25%)
- 青カニ へ行く (Go to blue crab)



Scratch code blocks for the merged state of the crab:

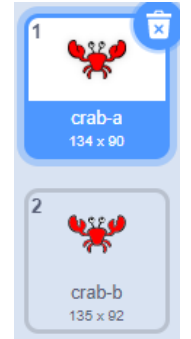
- が押されたとき (When clicked)
- 大きさを 60 %にする (Set size to 60%)
- x座標を 0 、y座標を -20 にする (Set x to 0, y to -20)
- ずっと (Forever loop) containing:
  - もし (If) containing:
    - 青カニ に触れた (Touched blue crab)
    - 大きさを 25 %にする (Set size to 25%)
    - 青カニ へ行く (Go to blue crab)

サイズが小さくなって  
青カニと合体する

○アニメ風の待ちポーズ

Scratch code blocks for an anime-style waiting pose:

- が押されたとき (When clicked)
- ずっと (Forever loop) containing:
  - 0.05 秒待つ (Wait 0.05 seconds)
  - 次のコスチュームにする (Next costume)



○脱出に失敗すると最初の状態に戻る

Scratch code blocks for returning to the initial state after a failed escape:

- 失敗 を受け取ったとき (When failed)
- 大きさを 60 %にする (Set size to 60%)
- x座標を 0 、y座標を -20 にする (Set x to 0, y to -20)

# 岩屋のプログラム



```
Scratch code blocks:
- when green flag clicked
- set x coordinate to -190, y coordinate to -140
- loop:
  - if touched by カニ友
    - increase success by 1
  - if touched by カニ友
    - send challenge message
```

変数で「成功」  
を作ります

=

```
Scratch code blocks:
- when green flag clicked
- set x coordinate to -190, y coordinate to -140
- loop:
  - if touched by カニ友
    - increase success by 1
    - send challenge message
```

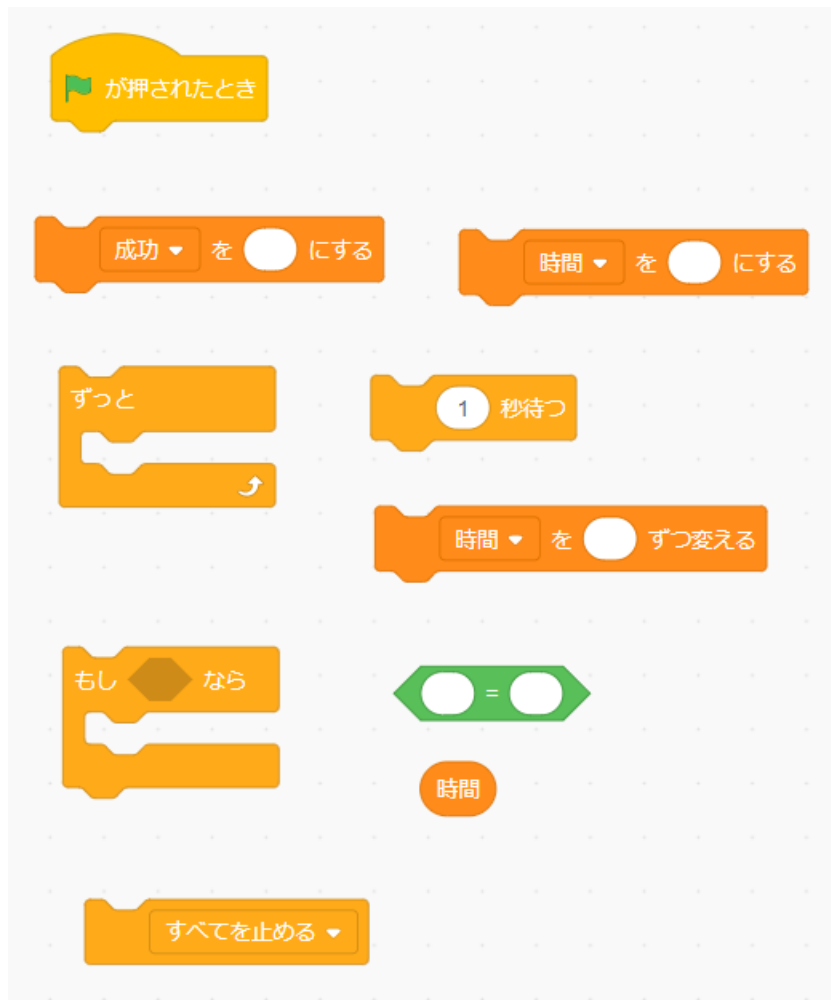
岩屋の位置  
(好きなところでOK)

コードの意味は  
わかるかな？

[[新しいメッセージ]]  
で作ります

## ステージのプログラム

- 『ステージ』 で変数を初期化、タイマーもセットします



タイマーが「カエルの道路横断」のときのちょっと違います

## 仕上げ

- 「チャレンジ」メッセージを受け取る青カニとカニ友に追加します

### 青カニに追加するコード



### カニ友に追加するコード



### ○レベルチェック

- 必要なブロックを見つけられる
- 見本を見ながらブロックを並べられる
- 間違ったところを自分で見つけられる、直せる
- ブロックの意味（コマンド）がなんとなく分かる
- ペイントエディタが使える（初歩レベル）
- 制作途中で作品を上書き保存できる

### ○関連する学習

x座標、y座標

正の数、負の数

順次処理

衝突判定

条件分岐

メッセージ

コスチュームの切り替え

ステージの切り替え

ライブラリの活用

BGM・音の入れ方

デバッグ

キーボード操作