

小学生 から始める プログラミング教室



Kid's
programmer
-master course-

ベーシックコース

ナ

勇者の冒険 スクラッチ 3.0 ver. ～クローンを利用して要素を増やそう～



リ

“逆転のアイデア”で
主人公を動かすしくみを作りよ！

1回目授業	年	月	日
2回目授業	年	月	日

名前

練習ファイルには、必要に応じて
一部の命令ブロックが記述してある場合があります。



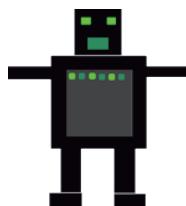
- ◆文科省 ICT活用教育アドバイザー事務局掲載
学校ICT化サポート事業者
- ◆長期コースによる、プログラミングの普及

変数

■ 変数とは

プログラムの世界では、状況にあわせて変化させたい値を「変数」として決めることができます。例として「ゲームの制限時間」や「体力」などがあげられます。

変数例



主人公 A の変数例

- 体力：100
- 素早さ：100
- 攻撃力：100
- 守備力：100

変数例

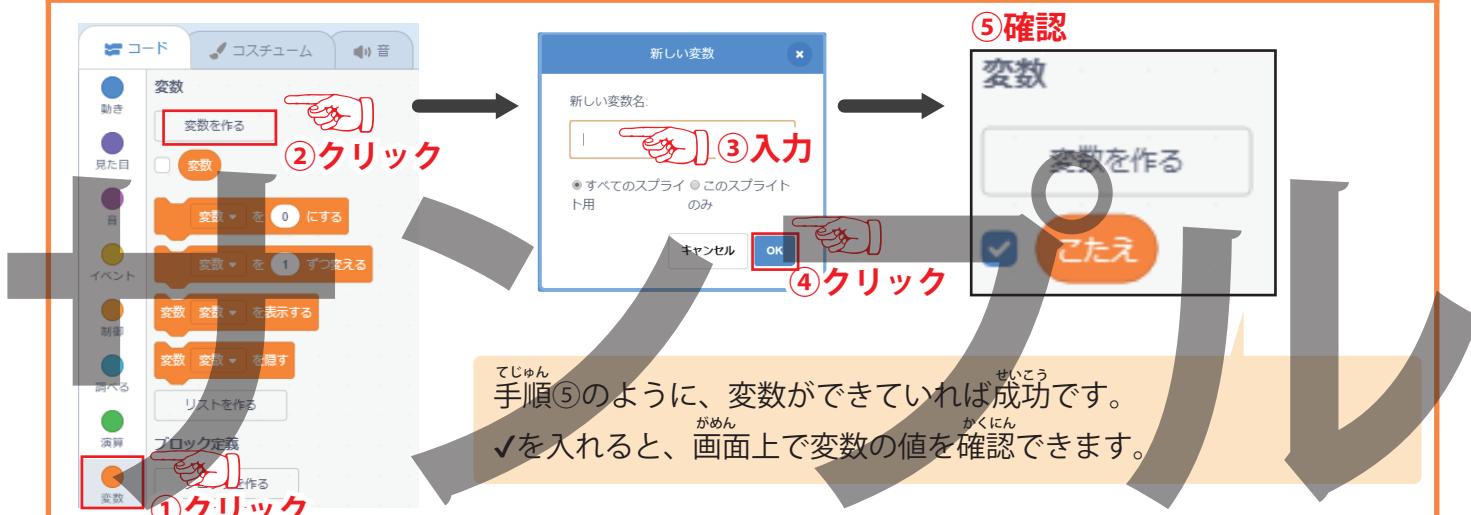


主人公 B の変数例

- 体力：200
- 素早さ：200
- 攻撃力：200
- 守備力：200

キャラクターごとに変数の値を変えることで、設定値を変えることができます。

■ 作り方



■ 表示の種類について



表示方法を変えられます。



変数は、スクラッチ 1.4 ver. にもありました。プログラミングを実行する上で必要な構文となるので必ず覚えて、使えるようになりましょう。

かんすう
関数

■ 関数とは

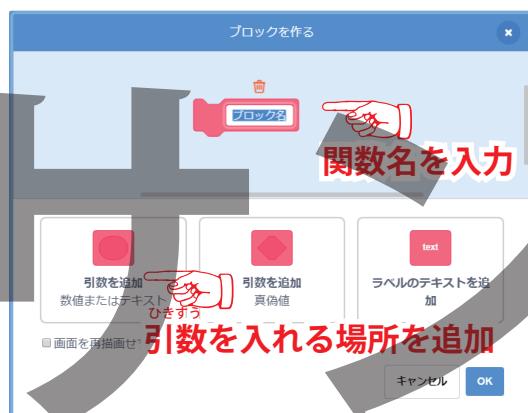
スクラッチ 3.0 ver. では、オリジナル関数（命令のかたまり）を定義することができます。定義するには、下のようにします。

■ 作り方

- ① 「ブロックを作る」をクリック



- ② 関数を定義



ブロック名・・・関数名
ひきすう ざいりょう
引数・・・関数に使う材料

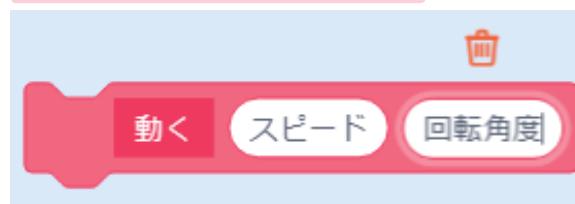
- ③ 確認



■ 使い方

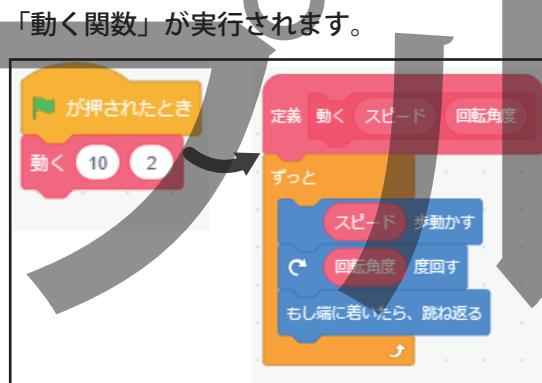
作成例

関数名「動く」
かいてんかくど
引数「スピード」「回転角度」

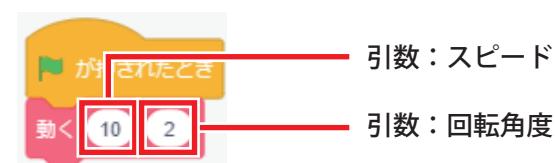


使用例

旗がクリックされたとき、
「動く関数」が実行されます。



引数に値を入力して、関数を実行させます。



動作例

対象のスプライトが、
10 歩ずつ動きながら、2 度ずつ回転しながら
移動し続けます。



関数のブロックを使えるようになると、プログラムをより短く書くことができます。



かんせい 完成ゲームを確認しよう

「チャレンジ > 2日目_チャレンジ完成 .sb3」を開こう

操作手順 1 フラッグを押してゲームをスタートします

主人公を動かして、60秒でどれだけ「宝石」を取れるか挑戦するゲームです。
体力が1より少なくなったり、ステージから落ちたりしたら、ゲームオーバーです。



操作手順 2 宝石や回復アイテムを、たくさん取ります

宝石・敵・回復アイテムは、ランダムに出てきます。

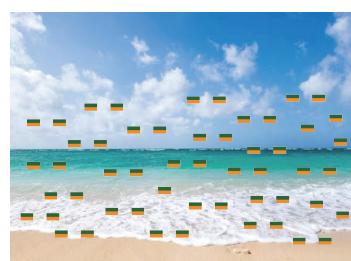
剣をふって、敵を“げきは”（うちやぶること）したときも、宝石の数が増えます。



操作手順 3 取った宝石の数が10を超したら、次のステージに進みます



⇒ 主人公のスピードが速くなります
⇒ ステージの背景や、陸地の形が変わります





みかた テキストの見方

よこ
ブロックの横に、その意味が書いてあります



もくひょう 今月の目標

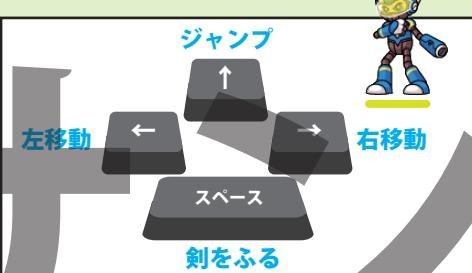
- いろ じょうけん
色を条件に動きを分ける
しくみを学ぼう！
- さくらんぼ ほうこう
スプライトが動く方向を
逆転させる計算を理解しよう！
- いどう
主人公が移動しているように
見せるしくみを知ろう！
- 「クローン」を学ぼう！

学習の手順

1

主人公の基本動作を作ります。
色 (■: 足 ■: 地面) によって
動きを分けるプログラムにします。

左・右・上キーに合わせて、
主人公を動かします。



ジャンプ		落下動作
60	-3 -3 -3	

学習項目

- いろ じょうけんぶんき
・色によって、条件分岐を行う
しくみ
- いろ
・キーで主人公を動かすしくみ



1
日目

3
章

2

$3 \times 1 = 3$ (右方向に移動)

学習項目

- いろ
・陸地を移動させるしくみ
(後ろのものを動かすことで、
手前のものが動いているように見せる)



1
日目

4
章

3



学習項目

- じゅうよう
★ 重要
・クローンを使って
要素を増やすしくみ



1
日目

2
日目

5
章

6
章 ~

ほかのアイテムも、陸地にそって移動するしくみをつくります。

また、クローンを使って要素を増やす方法を学びます。

取った宝石の数が 10 を超えたら主人公の速さや背景を変えるプログラムも、追加します。



しよう へんすう 使用変数 / スプライト

使用変数

p.1 「変数」を確認

●主人公のスピード	主人公が移動するスピード
●主人公の体力	主人公の今の体力
●宝石の数	取った宝石の数
●残り時間	ゲームをプレイできる時間
●向き	主人公が向いている方向
★敵の攻撃力	主人公の体力を減らす数（「敵」だけにある変数）
★回復量	主人公の体力を増やす数（「回復アイテム」だけにある変数）

スプライト

●主人公	陸地を冒険する勇者。左右に進んだりジャンプしたりできます
●陸地	主人公やアイテム、敵が動き回る地面
●宝石	主人公が集める、きれいな石
●回復アイテム	主人公の体力を取りもどすアイテム
●ゲームオーバー	主人公の体力がなくなったり、ステージから落ちたときの表示
●タイムアップ	残り時間がなくなつたときの表示
●敵	主人公にダメージを与えるモンスター
●スピードアップ	宝石の数が 10 より多くなつたときに出でくる表示



コスチューム

スプライトには「コスチューム」と呼ばれる、見た目があります。
それぞれのコスチュームはすでに作つてあるので、確認してみましょう。

●スピードアップ	●陸地	●主人公	●タイムアップ	●ステージ	●敵	●宝石	●回復アイテム
スピードアップ スピードアップ 304 x 57	陸地1 陸地1 480 x 248	右向き 右向き 83 x 117	右こうげき3 右こうげき3 135 x 160	タイムアップ! タイムアップ! 418 x 79	背景1 背景1 480 x 360	敵1 敵1 180 x 115	宝石1 宝石1 95 x 98
スピードアップ スピードアップ 304 x 57	陸地2 陸地2 454 x 175	左向き 左向き 82 x 117	左こうげき 左こうげき 128 x 148	背景2 背景2 480 x 360	敵2 敵2 129 x 116	宝石2 宝石2 92 x 92	回復1 回復1 100 x 100
スピードアップ スピードアップ 304 x 57	陸地2 陸地2 454 x 175	右こうげき 右こうげき 127 x 146	左こうげき2 左こうげき2 146 x 173	背景3 背景3 70 x 103	敵3 敵3 70 x 103	宝石3 宝石3 100 x 100	回復2 回復2 100 x 100
スピードアップ スピードアップ 304 x 57	陸地2 陸地2 454 x 175	右こうげき2 右こうげき2 145 x 172	左こうげき3 左こうげき3 135 x 161	背景4 背景4 262 x 184	ヒット ヒット 200 x 200	宝石4 宝石4 72 x 98	回復3 回復3 72 x 98



1日目



ステージと画面表示の作成

1 ステージを選ぼう

ステージでは、ゲームに必要な変数を決めます。
残り時間をカウントするしくみも作ります。



ステージ



旗 (はじめのボタン) が押されたとき

- 背景を「背景1」にする
- 主人公の体力を 50 にする
- 主人公のスピードを 3 にする
- 宝石の数を 0 にする
- 残り時間を 60 にする

背景1

- ずっと
- 1 秒ごとに
 - 残り時間を 1 ずつ減らす

もし 残り時間が 0 なら

- タイムアップを送る



プログラムを確かめよう

をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

旗を押したときに、残り時間が 1 秒きざみで減りましたか？

60 ~
数字が減る

残り時間 0



2 タイムアップとゲームオーバーの作成



1 タイムアップを選ぼう

ステージから「タイムアップ」の命令が送られてくると、表示されます。

タイムアップ!!

タイムアップ



旗（はじめのボタン）が押されたとき

・隠す



タイムアップを受け取ったとき

・表示する
・すべてを止める



2 ゲームオーバーを選ぼう



穴埋め問題にチャレンジ

右の説明を見ながら、①に何が入るか考えて、プログラムを完成させよう！

上の「タイムアップ」と似ているので、複製（コピー）して作るとかんたんです。

（複製のしかたは p.9 にあります）

ゲームオーバー

ゲームオーバー



旗（はじめのボタン）が押されたとき

・隠す

①

ゲームオーバーを受け取ったとき

・表示する
・すべてを止める



たし プログラムを確かめよう

をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

残り時間が0になったら、「タイムアップ!!」が表示されましたか？

●ためしてみよう！

タイムアップのコスチュームを、変更してみよう！



主人公の作成



えら 主人公を選ぼう

主人公

さいしょ 最初に、押したキーに合わせて主人公が左右に向く動きを作ります。



ゲームオーバーを受け取ったとき

・隠す

左向き矢印 キーが押されたとき

- ・コスチュームを「左向き」にする
- ・変数「向き」を1にする

ひだりむ 左向き

みぎむ 右向き

右向き矢印 キーが押されたとき

- ・コスチュームを「右向き」にする
- ・変数「向き」を2にする



たし プログラムを確かめよう

をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

右キーを押すと、主人公は「右向き」になりましたか？

左キーを押すと、主人公は「左向き」になりましたか？

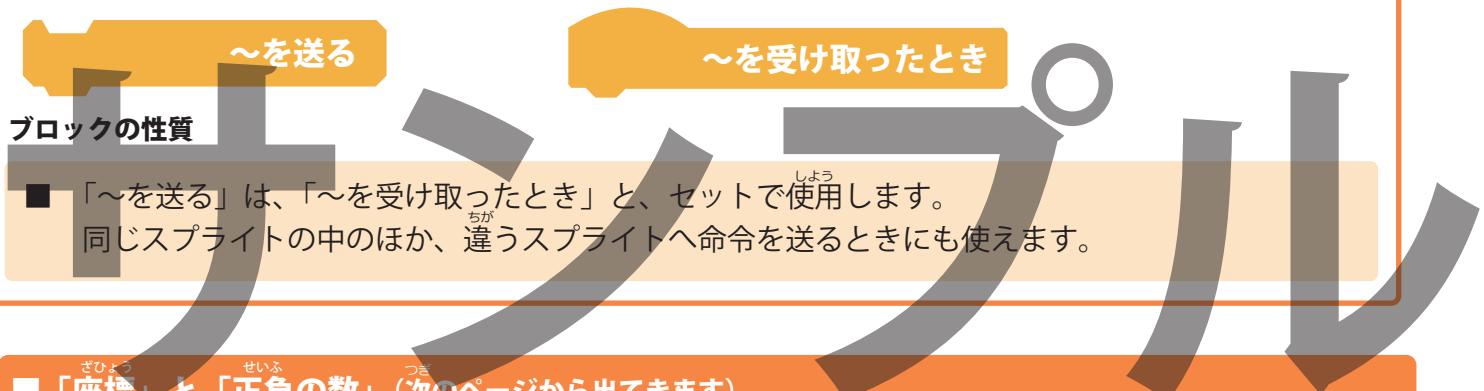


■ プログラムの複製（違うスプライトへの複製）

複製したいプログラムの上でクリックします。画面右下に表示されているスプライトへ、ドラッグで移動しましょう。移動したいスプライトに重なったら、マウスから指をはなします。

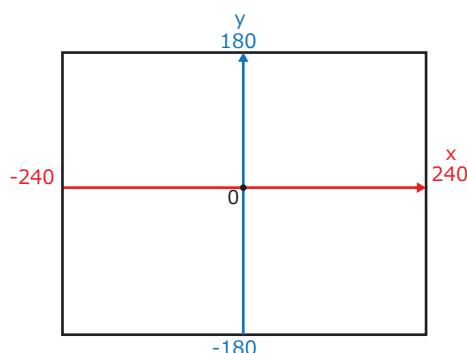


■ 「～を送る」の性質（スクラッチ 3.0 ver.）



■ 「座標」と「正負の数」（次のページから出てきます）

座標

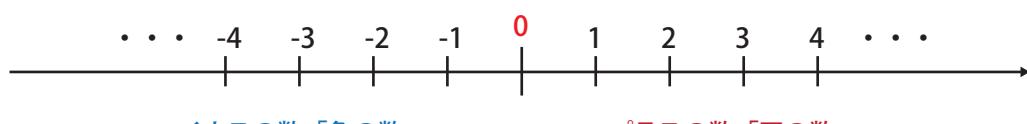


座標とは、位置を数字で管理する手法です。
x（横方向）とy（縦方向）で表し、
スプライトの位置を決定します。

画面の中心は、「x(0)・y(0)」と決められています。
スクラッチでは、座標の範囲はこのようになります。
x座標は「-240～240」
y座標は「-180～180」

正負の数

0より小さい数を「**負の数（マイナス）**」、0より大きい数字を「**正の数（プラス）**」といいます。
普段の生活で使っている0より大きい数はすべて「プラス」の数です。プラスの記号は通常、特に表記されず、省略されています。





■主人公がジャンプするしくみ

もし、主人公が着地している状態で、上キーが押されたら、ジャンプします。

もし

着地状態

足(黄緑)が地面(緑)に
触れている状態

かつ



主人公のジャンプ動作 60

ジャンプする

足(黄緑)が地面(緑)から
はなれて、
y座標が 60 変わります

■主人公が地面に落ちるしくみ

主人公が地面に着地をしていなければ(ジャンプ中なら)、地面に向かって落ちます。

でなければ(ジャンプしているなら)

主人公のジャンプ動作 -3

落下動作

足(黄緑)が地面(緑)に
向かって落ち続けます
(y座標が -3 ずつ
変わります)

■ジャンプする関数を知ろう！

主人公が「ジャンプ」または「落下」するときは
「ジャンプの高さ(変えるy座標の数)」を使います。

定義 主人公のジャンプ動作 ジャンプの高さ

y座標を ジャンプの高さ ずつ変える

もし、主人公のy座標が -175 より下になつたら、
ゲームオーバーを送る

ジャンプ	落下動作
60	-3 -3 -3





2

主人公の基本動作を作る

主人公は **足の色** と **地面の色** を判断して、「ジャンプ」や「落下」をします。



主人公

旗（はじめのボタン）が押されたとき

コスチュームを 左向き ▾ にする

表示する

x座標を 0 、y座標を 0 にする

ずっと

もし 色が 色に触れた かつ 上向き矢印 ▾ キーが押された なら

主人公のジャンプ動作 60

でなければ

色が 色に触れた まで繰り返す

主人公のジャンプ動作 -3

もし 主人公の体力 < 1 なら

ゲームオーバー ▾ を送る

ずっと

もし 「足 が 地面 に触れていて」
「上キー」が押されたなら

・「主人公のジャンプ動作」関数
の値を 60 にして実行

ジャンプ

でなければ

「足 が 地面 に触れる」まで繰り返す

・「主人公のジャンプ動作」関数
の値を -3 にして実行

落下

もし 主人公の体力が 1 より小さいなら

・ゲームオーバーを送る

■ 色 を選ぶ方法

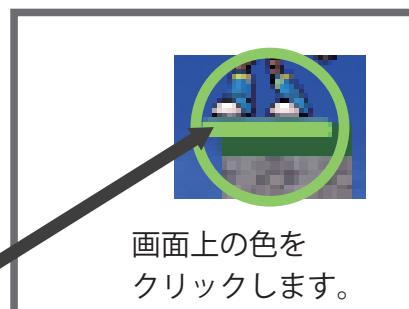
色に触れた

色をクリック



← 【色番号を指定する場合】
ここで番号を指定します

← 【画面上の色を使う場合】



画面上の色を
クリックします。



3 主人公がジャンプする関数を作る

ジャンプも落下も、「ジャンプの高さ」の数字を使うしくみです。
ステージから落ちたら、「ゲームオーバー」を表示する命令を送ります。



主人公



主人公のジャンプ動作が実行されたとき

・y 座標を「ジャンプの高さ」^{たか} ずつ変える

もし y 座標が -175 より小さいなら

・ゲームオーバーを送る

4 プログラムを確かめよう

をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

主人公が着地しているときに上キーを押したら、主人公がジャンプしましたか？

●ためしてみよう！

主人公のジャンプ力を 200 にしてみよう！

4 陸地の作成

1 陸地の移動スピード・方向の計算

左・右キーを押すと、陸地はそれと逆の方向に移動します。

移動するスピードと方向は、「主人公のスピード」×「方向（1 または -1）」で決めています。

左キーで「方向」が 1（右）に、右キーで「方向」が -1（左）になります。

もし、主人公のスピードが 3 なら…



「主人公のスピード」×「方向」 ずつ、
x 座標を変えます。



を押したとき 方向 1 正の方向

3 × 1 = 3 (右方向に移動)



を押したとき 方向 -1 負の方向

3 × -1 = -3 (左方向に移動)

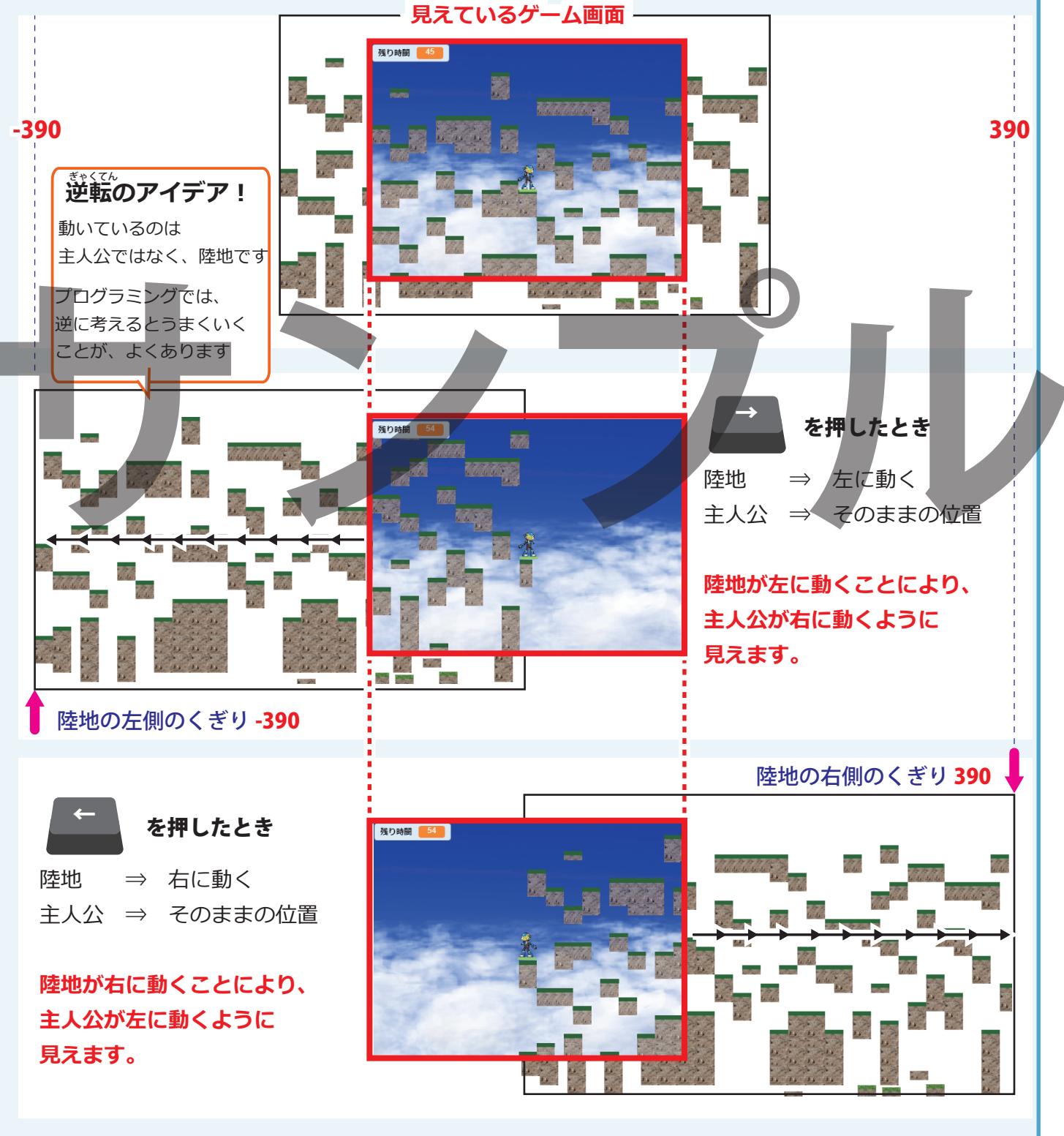


2 陸地のプログラムのしくみ

主人公をキーで動かしているのではありません。主人公はその場でジャンプしているだけです。陸地を左右に動かしているので、まるで主人公が左右に動いているように見えるのです。

■ 陸地が動くしくみ

あいだ
x 座標 -390 ~ 390 の間を、陸地が動きます。





3

陸地を選ぼう！

陸地を動かすプログラムを作ります。

陸地



ゲームオーバーを受け取ったとき

- 隠す



旗（はじめのボタン）が押されたとき

- 0.1 秒待つ
- コスチュームを「陸地1」にする
- 画面のいちばん後ろへ移動する
- 表示する
- 最初の位置を決める

ずっと

もし「右キー」が押されたなら

- 「陸地の左右移動」関数の値を -1 にして実行

陸地を左にずらす

もし「左キー」が押されたなら

- 「陸地の左右移動」関数の値を 1 にして実行

陸地を右にずらす



陸地の左右移動が実行されたとき

もし x 座標が -390 より大きく
 x 座標が 390 より小さいなら

- x 座標を「主人公のスピード」 × 「方向」ずつ変える



プログラムを確かめよう

■ をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

- 左右のキーを押したとき、陸地が動いていましたか？

●ためしてみよう！

- ばいそく主人公がまるで倍速で動いているように、いどうそくど陸地の移動速度を2倍にしてみよう！

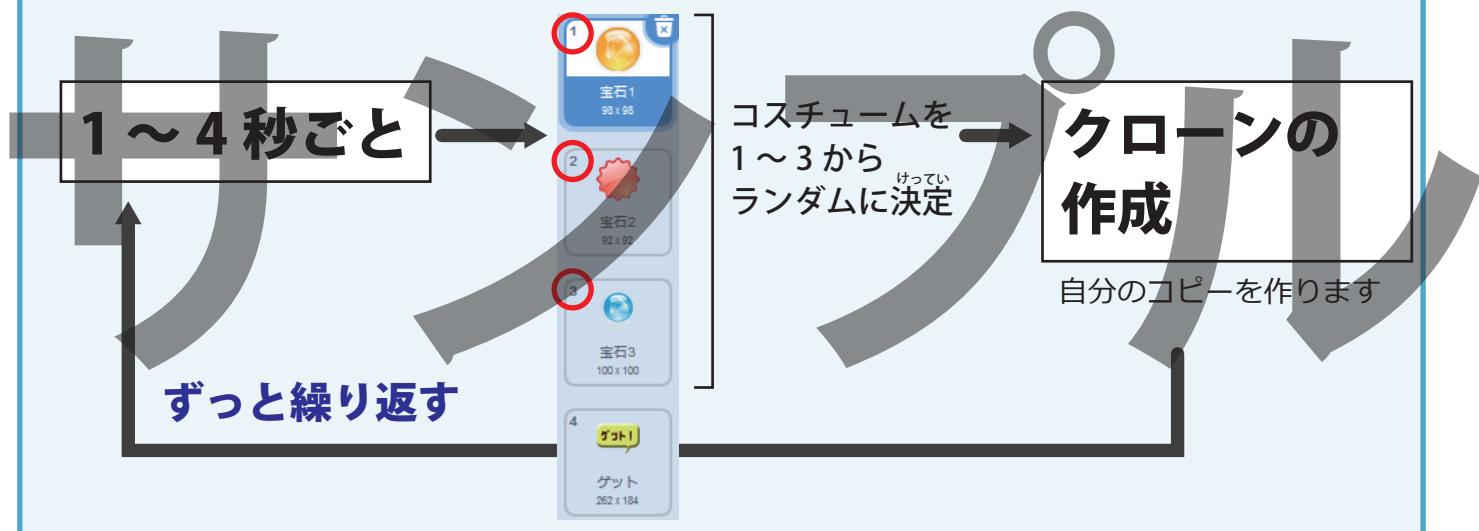


宝石の作成

■「クローン」で宝石をたくさん出す

スタートの合図で、宝石は1～4秒ごとに「クローン」を作ります。

これはひとつのスプライトからコピーをいくつも作って、画面にたくさん出す方法です。



旗（はじめのボタン）が押されたとき

・隠す



ずっと

- ・1～4秒ごとに
- ・コストチュームを1～3からランダムに決める
- ・クローンを作る



■宝石（敵・回復アイテム）が移動するしくみ① ⇒ 陸地と一緒に移動させる！

「宝石」「敵」「回復アイテム」は、左・右キーを押したとき、陸地と一緒に動くようにします。もし陸地と一緒に動かなかったら、左・右キーを押したときに、宝石がその場に取り残されたり、主人公につかまらず逃げていくように見えたりしてしまいます。

x座標を **主人公のスピード** * **方向** ずつ変える

「主人公のスピード」×「方向」ずつ、
x座標を変えます。



を押したとき **方向**

1 正の方向

(右方向に移動)



を押したとき **方向**

-1 負の方向

(左方向に移動)

おな
陸地と同じ動き

(方向の決め方は

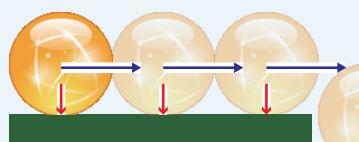
p.12と同じです)

Scratch script for the gem movement mechanism:

- 克隆されたとき**:
 - x座標を **-240** から **240** までの乱数にする
 - y座標を **-180** から **180** までの乱数にする
 - 表示する
- ずっと**:
 - 1 歩動かす
 - もし **右向き矢印** キーが押されたなら
 - 宝石の左右移動 **-1**
 - もし **左向き矢印** キーが押されたなら
 - 宝石の左右移動 **1**
 - 色に触れたまで繰り返す
 - 宝石の落下動作
- もし 主人公** に触れたなら
 - 宝石が主人公に触れる動作
- もし 端** に触れたなら
 - 隠す

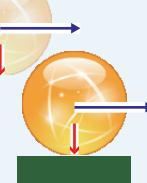


■宝石（敵・回復アイテム）が移動するしくみ② ⇒ 地面にそって移動させる！



ずっと1歩ずつ右に移動

ずっと緑（地面）に落下している



「宝石」「敵」「回復アイテム」は、
地面の上をずっと右に移動しながら、
地面に向かってずっと落下しています。
こうすると、地面にそって移動していきます。



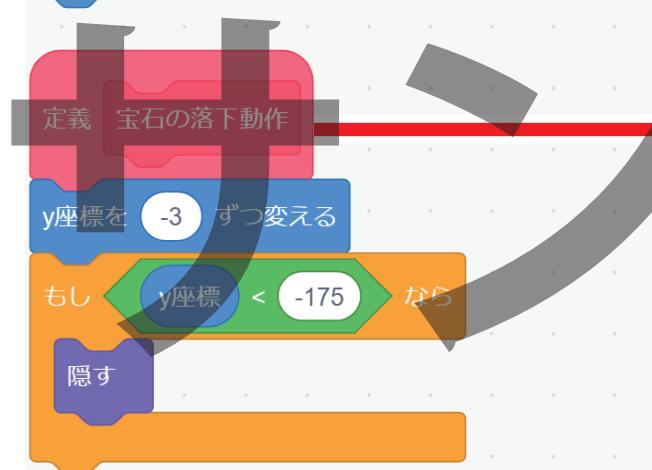
宝石の左右移動が実行されたとき



宝石



・x座標を「主人公のスピード」×「方向」ずつ変える



宝石の落下動作が実行されたとき

・y座標を -3 ずつ変える (下へ落ちる)
もし y 座標 < -175 なら
隠す



宝石が主人公に触れる動作が実行されたとき



・宝石の数を 1 ずつ増やす
・コスチュームを「ゲット」にする
・0.5秒待つ
・隠す

ゲット

ゲット！



ゲームオーバーを受け取ったとき

・隠す



プログラムを確かめよう

FLAG をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

- 1～4秒ごとに、3種類の宝石がランダムに出てきましたか？
- 宝石が主人公に触れたら、「宝石の数」が1つ増えて、その宝石が消えましたか？

●ためしてみよう！

- 宝石が0.3秒ごとに出現するようにプログラムを変更してみよう！



早くできた生徒は取り組もう

チャレンジ1：改造してみよう！

練習ファイル チャレンジ > 1日目_チャレンジ練習.sb3

完成ファイル チャレンジ > 1日目_チャレンジ完成.sb3

★ ① 残り時間が20秒になったら、主人公のスピードが5になるプログラムを作ろう！

ヒント：主人公のプログラムに「もし～なら」のブロックを使って作成しよう！

★ ② 主人公が宝石を取ったとき、体力が「10」回復する命令を追加しよう！

ヒント：宝石のプログラムに「～を～ずつ変える」ブロックを加えて作成しよう！



早くできた生徒は取り組もう

チャレンジ2：みんなの意見を取り入れよう

プログラミングのお仕事では、お客様のお願いを聞いて取り入れることが多くあります。

先生やお友だち、おうちの人などに今日作ったゲームを見せて、

もっとよくするアイデアを聞いてみよう！（おうちで聞いたことも、忘れずにメモしておいてね！）

例）宝石のほかにも、すてきなアイテムを増やしたいな！



1日目

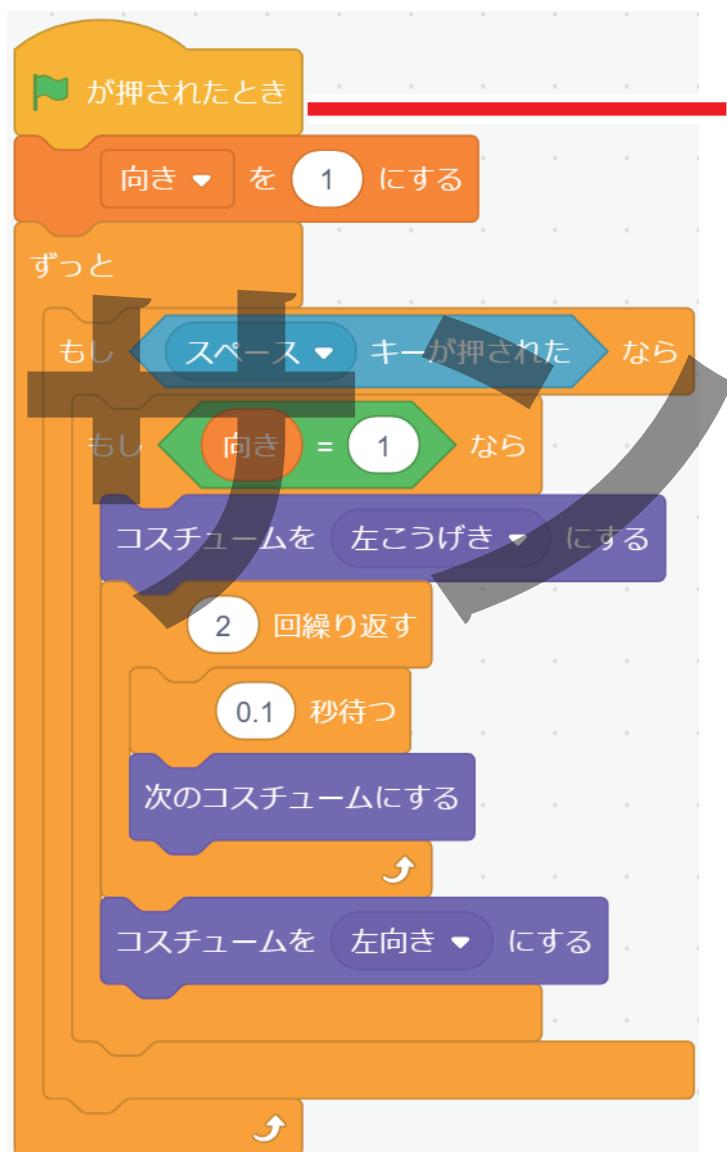
終了です



スペースキーを押したときに、主人公が剣をふるるようにしましょう。



主人公



旗（はじめのボタン）が押されたとき

・変数「向き」を1にする

ずっと

もし「スペースキー」が押されたなら

もし「向き」が1（左向き）なら

・コスチュームを「左こうげき」にする
2回繰り返す
・0.1秒後に次のコスチュームにする

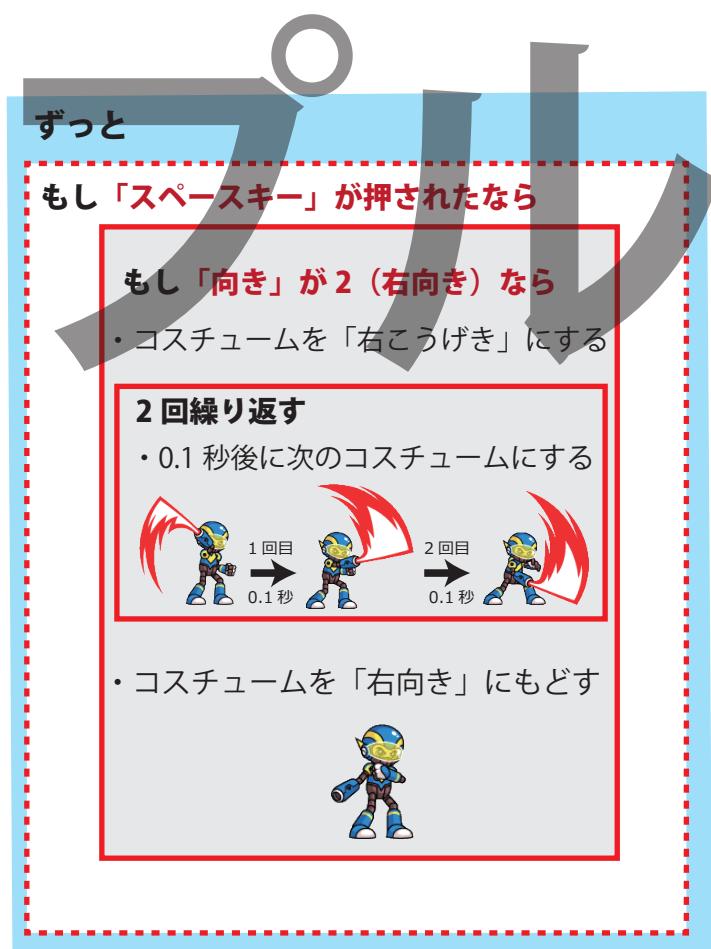
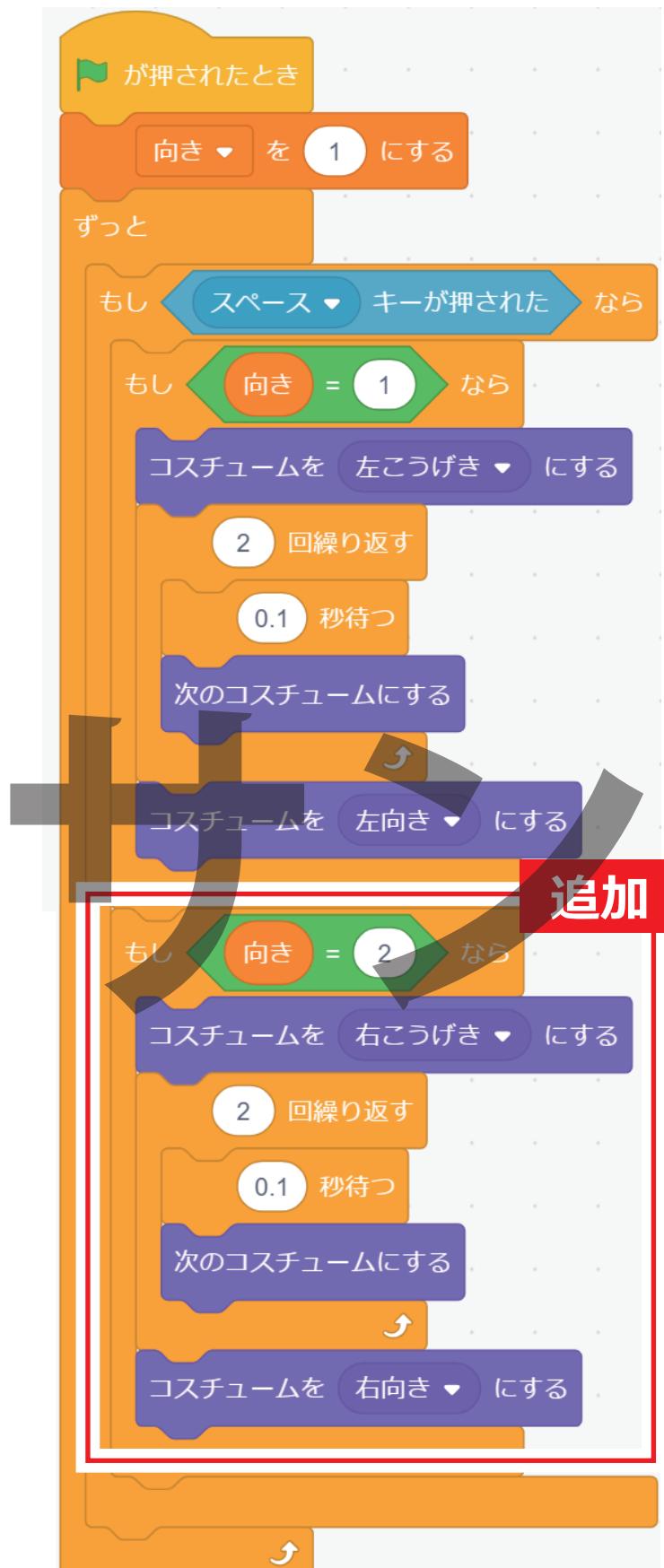


・コスチュームを「左向き」にもどす



をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

主人公が左を向いた状態でスペースキーを押すと、左こうげきの動きになりましたか？



プログラムを確かめよう

 をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

□ 主人公が右を向いた状態でスペースキーを押すと、「右こうげき」の動きになりましたか？



7

敵の出現

1

敵を選ぼう

敵のプログラムは、宝石のプログラムを複製して作ると、かんたんです。

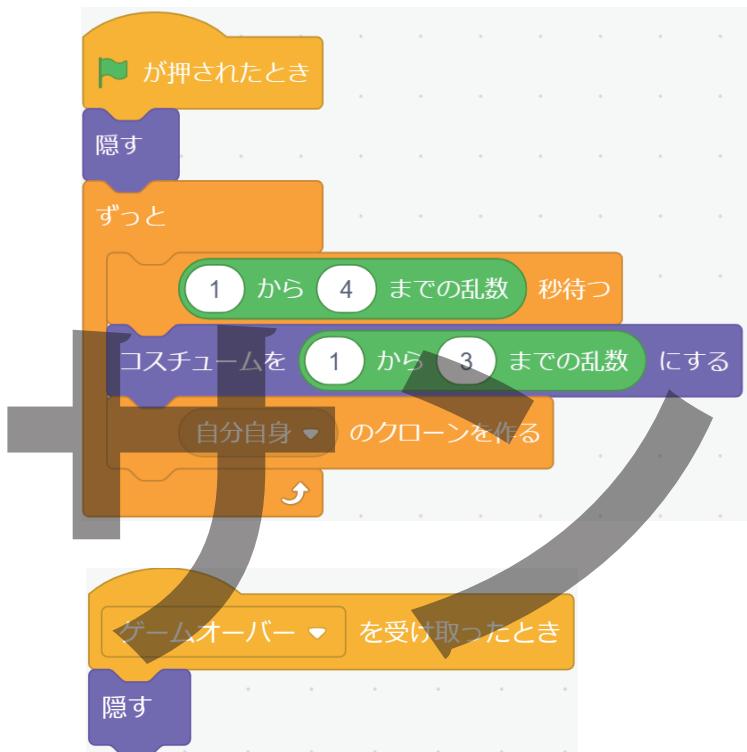
(複製のしかたは p.9 にあります)



敵



宝石のプログラムを複製して作ろう！



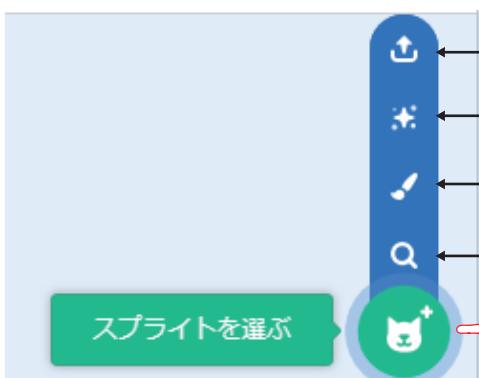
「敵」と「回復アイテム」のしくみは
「宝石」のプログラムと、ほぼ同じです。



コストチュームを
1～3から
ランダムに決定

■ 改造するとき便利！ 新しいスプライトを追加する方法

ゲーム完成後、改造してスプライトを増やすときは、スクラッチに入っているものを使うと便利です。



マウスを乗せる

画面の右下の「スプライトを選ぶ」マークから、
好きなスプライトを追加できます。





宝石のプログラムを複製して作ろう！



敵

クローンされたとき

x座標を -240 から 240 までの乱数、y座標を -180 から 180 までの乱数 にする

表示する

ずっと

1 歩動かす

敵を左にずらす

もし 右向き矢印 ▾ キーが押された なら

敵の左右移動 -1

敵を右にずらす

もし 左向き矢印 ▾ キーが押された なら

敵の左右移動 1

敵の落下動作

色に触れた まで繰り返す

敵の落下動作

主人公に触れたら

もし 主人公 ▾ に触れた なら

もし 色に触れた なら

もし剣に触れたなら

・「敵げきは」関数を実行

でなければ

そうでなければ

敵ヒット

・「敵ヒット」関数を実行

もし 端 ▾ に触れた なら

隠す

端に触れたら

…宝石プログラムから複製した後に、
変更または追加するところ

定義 敵の左右移動 方向

x座標を 主人公のスピード * 方向 ずつ変える

定義 敵の落下動作

y座標を -3 ずつ変える

もし y座標 < -175 なら

隠す

定義 敵げきは

「敵げきは」関数が
実行されたとき

宝石の数 ▾ を 1 ずつ変える

・宝石を 1 増やす
・隠す

定義 敵ヒット

「敵ヒット」関数が
実行されたとき

主人公の体力 ▾ を -10 ずつ変える

コスチュームを ヒット ▾ にする



0.5 秒待つ

・主人公の体力を
10 減らして、
ヒットの表示にする

・隠す



プログラムを確かめよう

をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

- 敵が主人公の剣に当たると宝石の数が増えましたか？
- 敵が主人公に触れたら、主人公の体力が 10 減りましたか？

●ためしてみよう！

- 敵が 0.5 秒ごとに出現するようにしてみよう！
- 敵にあたると主人公の体力が -50 減るようにしてみよう！



回復アイテムの出現



回復アイテムを選ぼう

回復アイテムも、宝石のプログラムを複製して作りましょう。



宝石のプログラムを複製して作ろう！



穴埋め問題にチャレンジ

右の説明を確認しながら、①に何が入るか考えて、プログラムを完成させよう！

が押されたとき

隠す

ずっと

1 から 4 までの乱数 秒待つ

①

自分自身 ▾ のクローンを作る

ゲームオーバー ▾ を受け取ったとき

隠す



回復アイテム



コストを
1 ~ 2 から
ランダムに決定



宝石のプログラムを複製して作ろう！

回復アイテムのしくみは、p.15～p.17の「宝石」のプログラムと同じです。



回復アイテム



…複製した後に、変更するところ

クローンされたとき

x座標を -240 から 240 までの乱数、y座標を -180 から 180 までの乱数 にする

表示する

ずっと

1 歩動かす

回復アイテムを左にずらす

もし 右向き矢印 ▾ キーが押された なら

回復アイテムの左右移動 -1

回復アイテムを右にずらす

もし 左向き矢印 ▾ キーが押された なら

回復アイテムの左右移動 1

回復アイテムの落下動作

色に触れた まで繰り返す

回復アイテムの落下動作

主人公に触れたら

もし 主人公 ▾ に触れた なら

回復アイテムが主人公に触れる動作

端に触れたら

もし 端 ▾ に触れた なら

隠す

体力回復



定義 回復アイテムの左右移動 方向

x座標を 主人公のスピード * 方向 ずつ変える

定義 回復アイテムの落下動作

y座標を -3 ずつ変える

もし y座標 < -175 なら

隠す

定義 回復アイテムが主人公に触れる動作

主人公の体力 ▾ を 5 ずつ変える

コスチュームを 体力回復 ▾ にする

0.5 秒待つ

隠す



たし プログラムを確かめよう

旗をクリックして、ゲームを実行させよう。できていたら✓をつけよう。

1~4秒ごとに、2種類の回復アイテムがランダムに出てきましたか？

回復アイテムは端に触れたら、消えましたか？

●ためしてみよう！

回復アイテムが主人公にふれたら、主人公の体力が50増えるようにしてみよう！

9

かくちょうようそ 拡張要素を追加しよう！

ここでいう「拡張」とは、ゲームをグレードアップすることです。
「宝石の数」が10より大きくなったら、変化を2つ加えます。



「宝石の数」が10より大きくなったら

- ① 主人公のスピードを速くします。
- ② 「ステージ」と「陸地」のコスチュームを変えます。

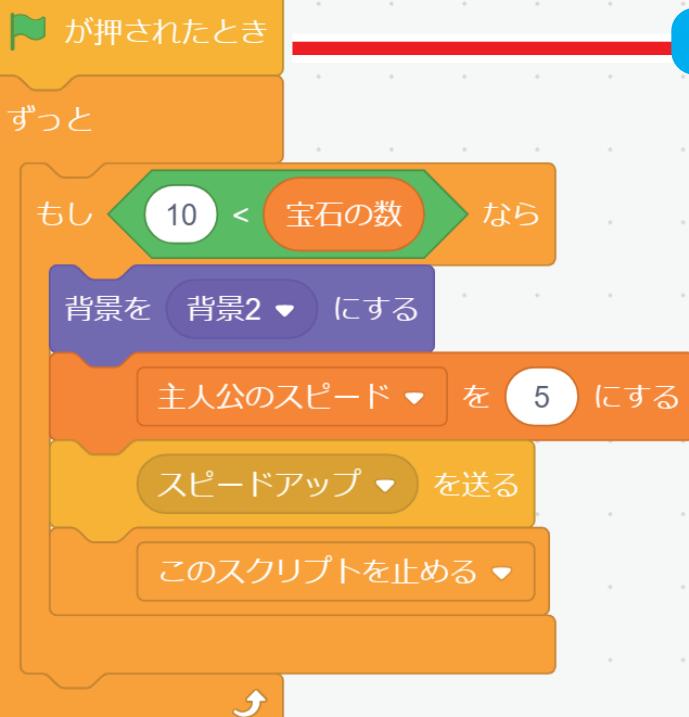
1

ステージを選ぼう

宝石の数が10より大きくなったら、ステージの背景を変え、主人公のスピードを3から5に変えます。

ステージ

このプログラムを追加します



旗（はじめのボタン）が押されたとき

ずっと

もし 宝石の数が 10 より大きいなら

- ・背景を「背景2」にする
- ・主人公のスピードを5にする
- ・スピードアップを送る
- ・このスクリプト(プログラム)を止める

背景2



「スピードアップ」に送る

スピードアップ



2 スピードアップを選ぼう

宝石の数が 10 より大きくなったときに、ステージから命令が送られてきます。
「スピードアップ」の文字を、画面の右から左へと動かしながら表示します。

スピードアップ
スピードアップ

■「スピードアップ」の文字が動くしくみ

ステージからスピードアップの命令を受け取ったら、

「スピードアップ」の文字が画面の右 (x 座標 400) から左 (x 座標 -450) へと、3 秒で移動します。

The diagram illustrates the movement of the text "スピードアップ" (Speed Up) from right to left. It shows a horizontal timeline with markers at -450, 400, and the center. The text "3秒で移動" (Move in 3 seconds) is positioned in the center. Below the timeline, the text "スピードアップ" is shown in red, with arrows indicating its movement from right to left. The background shows a game stage with a dragon and a castle.

穴埋め問題にチャレンジ

上の図を確認しながら、①～⑤に何が入るか考えて、「スピードアップ」のプログラムを完成させよう！

スクリプト

旗 (はじめのボタン) が押されたとき

- 隠す

スピードアップを受け取ったとき

- 表示する
- x座標を ① 、y座標を ② にする
- ③ 秒でx座標を ④ に、y座標を ⑤ に変える
- 隠す
- このスクリプトを止める ▾

① ② ③ ④ ⑤



3

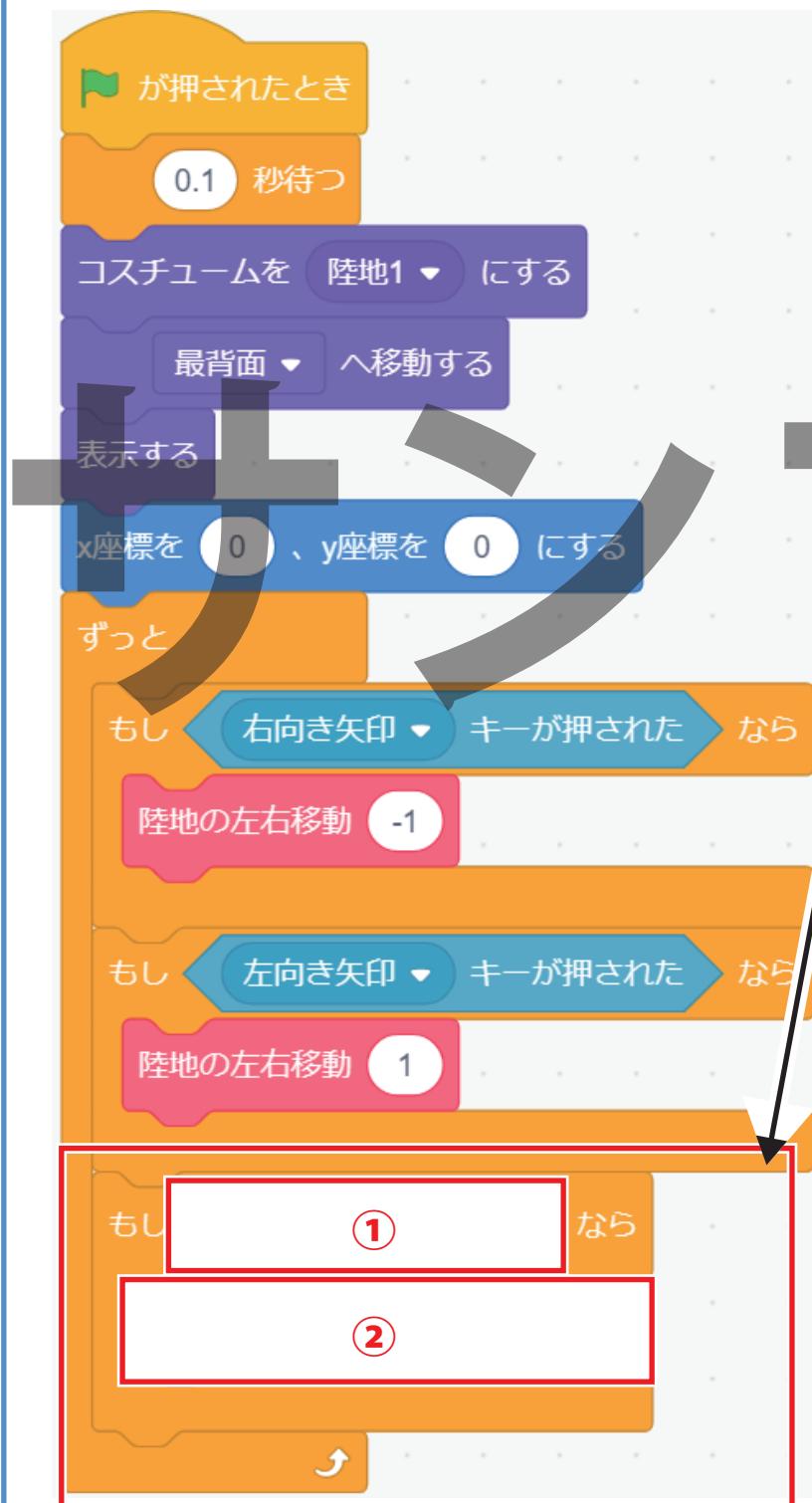
陸地を選ぼう

陸地のコスチュームを変えます。

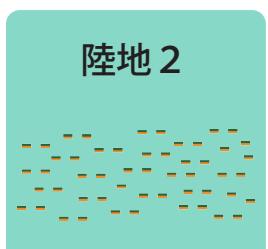
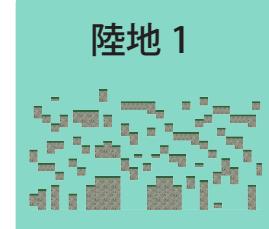


あなう 穴埋め問題にチャレンジ

すでに作ってあるプログラムに、下の部分を追加して、陸地を変化させよう！



もし、
「宝石の数」が
10 より大きくなったら
コスチュームを
「陸地 2」にします





早くできた生徒は取り組もう

チャレンジ：もっとグレードアップしよう！

練習ファイル チャレンジ > 2日目_チャレンジ練習.sb3

完成ファイル チャレンジ > 2日目_チャレンジ完成.sb3

さらにプログラムを追加して、ゲームをおもしろくしましょう。

クローンで増やしたアイテムの「回復量」「敵の攻撃力」を、
コスチュームの番号によってそれぞれ違う数に変えます。



アイテムごとに、
違う回復量にします



敵ごとに、
違う攻撃力にします

正解は次のページ→

ショートカットキー：やりなおす

もしブロックを組みまちがえたら、**Ctrl + Z** を押すと、ひとつ前にもどることができます。
そこからもう一度やりなおしをしたい場合は、**Ctrl + Shift + Z** を押しましょう。





回復アイテムを選ぼう



回復アイテム

…追加または変更しよう！

```
当りのとき
  隠す
  ずっと
    1 から 4 までの乱数 秒待つ
    コスチュームを 1 から 2 までの乱数 にする
    もし コスチュームの 番号 ▾ = 1 なら
      回復量 ▾ を 5 にする
    もし コスチュームの 番号 ▾ = 2 なら
      回復量 ▾ を 10 にする
    自分自身 ▾ のクローンを作る
```

敵を選ぼう



敵

…追加または変更しよう！

```
当りのとき
  隠す
  ずっと
    1 から 4 までの乱数 秒待つ
    コスチュームを 1 から 3 までの乱数 にする
    もし コスチュームの 番号 ▾ = 1 なら
      敵の攻撃力 ▾ を -5 にする
    もし コスチュームの 番号 ▾ = 2 なら
      敵の攻撃力 ▾ を -10 にする
    もし コスチュームの 番号 ▾ = 3 なら
      敵の攻撃力 ▾ を -15 にする
    自分自身 ▾ のクローンを作る
```

定義 回復アイテムが主人公に触れる動作

```
主人公の体力 ▾ を 回復量 ずつ変える
コスチュームを 体力回復 ▾ にする
0.5 秒待つ
隠す
```

定義 敵ヒット

```
主人公の体力 ▾ を 敵の攻撃力 ずつ変える
コスチュームを ヒット ▾ にする
0.5 秒待つ
隠す
```



早くできた生徒は取り組もう

チャレンジ：みんなのアイデアをもとに改造しよう！

ほかの人の意見や自分の考えをもとに、もっと楽しめるゲームに改造しましょう。
終わったら、みんなにわかるように発表しましょう。

かいぞうれい 改造例 (追加アイテムの例)



■追加アイテム 時計

⇒ 残り時間を 10 増やすアイテムです！



■追加アイテム はさみ

⇒ 残り時間を 10 減らすアイテムです！



できるプログラマーへの道：「わりこみ」してもいいんです！？

プログラムのとちゅうで動きを変える

「キーが押されたなら (押されたとき)」という命令は、よく使いますね。

キャラクターが動いているとちゅうで、キーが押されると
プログラムが「あ、押されたな～」と気づいて、動きを変えます。



このように、プログラムを動かしているとちゅうで、
何かをきっかけに別のプログラムを動かすことを、「わりこみ」とコンピューター用語でいいます。
きっかけになった操作は、「イベント」といいます。

「イベント」の確認でコンピューターは大忙し

スクラッチでイベントを作るのはかんたんですが、コンピューターはむずかしい仕事をやっています。

ふつう、コンピューターはいくつもの命令をいちどに動かすことはできません（※）。

なのでプログラムを動かしながら、
ときどき「キーが押されたかな？」と確認しているのです。



アクションゲームなど、タイミングが大事なゲームを作るときは
イベントを確認する順番も、重要です。

たとえば 2 人対戦のゲームだったら、確認する順番によって勝ち負けが変わるかもしれません。

「わりこみ」に注意しよう

複雑なプログラムを作るようにになったら、イベントによる「わりこみ」を
コンピューターでどう動かすかが、たいせつになります。
「わりこみ」には注意が必要なことがあるんだな、と頭に入れておきましょう。

（※）近年はマルチコア・マルチスレッドのコンピューターが普通ですが、スクラッチはシングルスレッドで実行されています。

ベーシックコース

勇者の冒険 スクラッチ 3.0 ver.

～クローンを利用して要素を増やそう～



バージョンによる表示の違い

スクラッチ 3.0 ver. はバージョンによって、一部の命令の表示やデザインが違うことがあります。どちらの表示でも、プログラムには問題ありません。



このテキスト

プログラム内 「座標」 の表記について

スクラッチ 3.0 ver. のファイル読み込み時、「座標」の表記が英語になる場合がありますが、プログラムには問題ありません。



◆文科省 ICT活用教育アドバイザー事務局掲載
学校ICT化サポート事業者

◆長期コースによる、プログラミングの普及