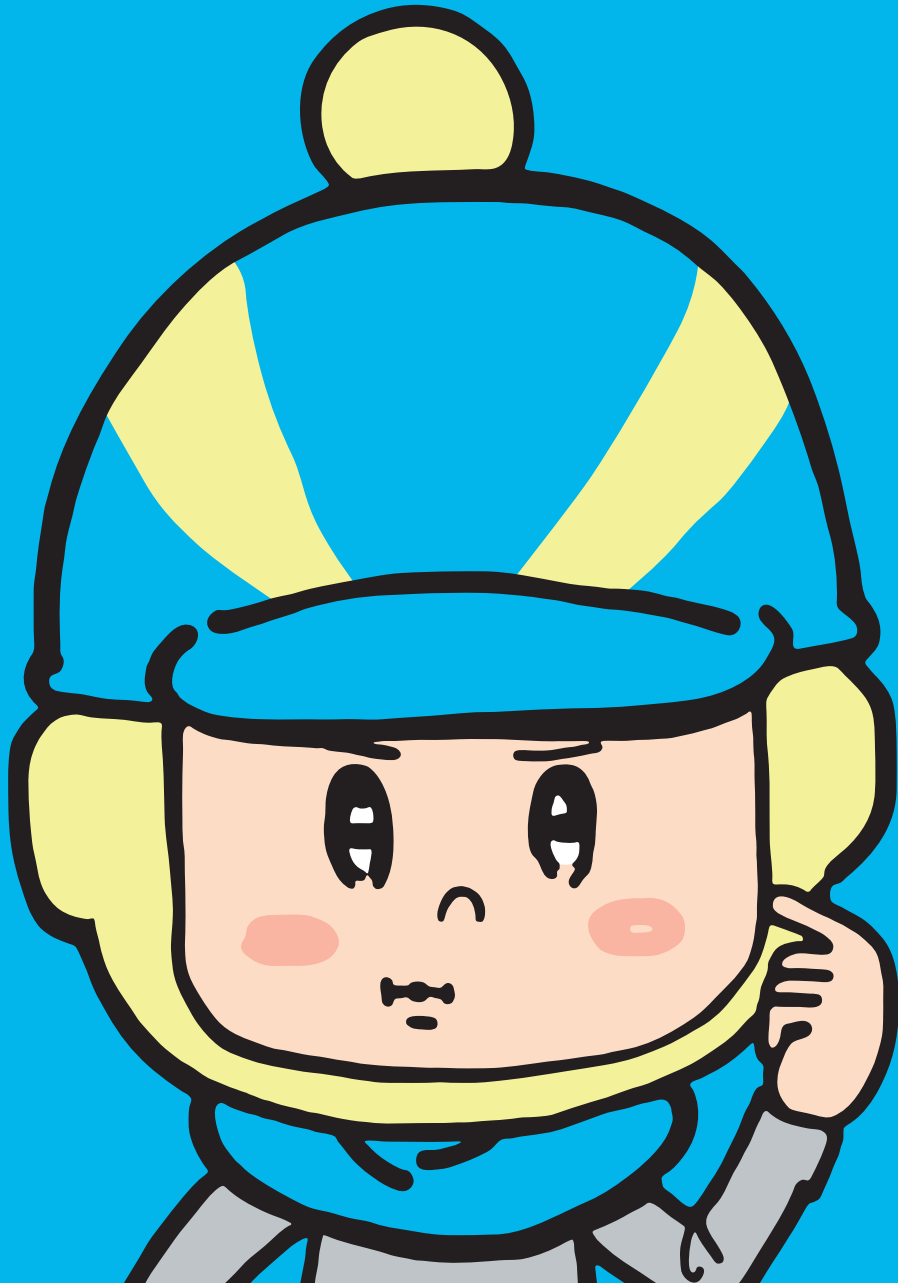


ゲームを作ろう!

— 初級編 —



ゲーム

1

森の射撃訓練

指令

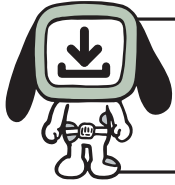
- Scratchの使い方を覚えよう
- 処理の順序を考えよう



ゲームの説明書

最初のゲームは「森の射撃訓練」です。エルフを操作して、つぎつぎ現れる的を矢で射抜きましょう！ 当てた数は、画面の左上に星印で表示されます。





ゲームデータのアクセス先

<https://scratch.mit.edu/projects/172353414/>

インターネットが使える環境で、上記のURLにアクセスしてください。画面が表示されたら「中を見る」のボタンをクリックします。

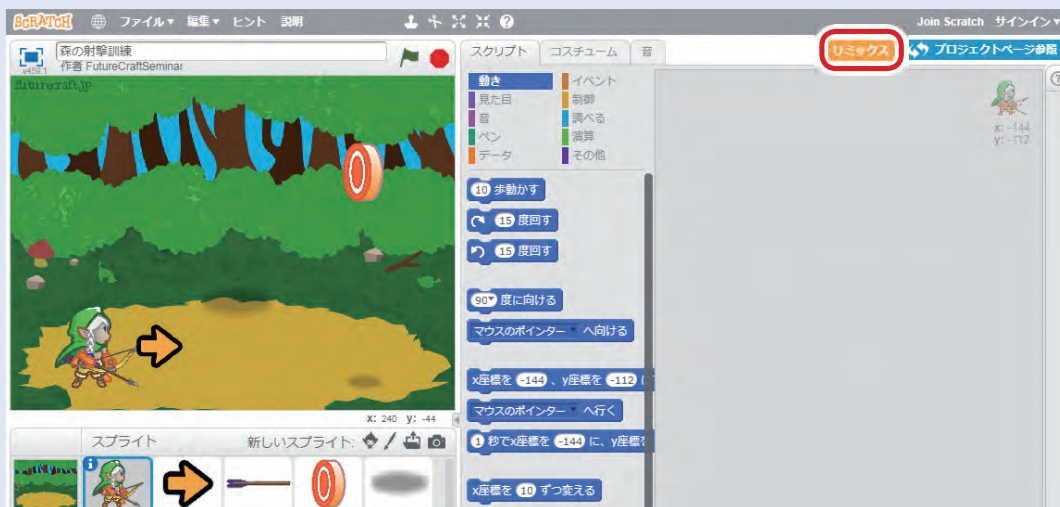


エディター画面が表示されたらゲーム作りを開始できます。



リミックスしてみよう！

Scratchのアカウントを持っている場合は、右上のリミックスボタンを押して自分のプロジェクトとしてコピーしましょう。



ゲームデータをScratchの画面に読み込んだら、画面の右上にある旗をクリックして、ゲームを開始しましょう。どうなるでしょうか。



あれ!? なにをしても矢が撃てない! 矢印も動かせないぞ??
どうやって操作するんだ?



ふふん。だから言ったじゃない。ゲームはまだ未完成なの。
エルフにプログラミングして、ゲームを完成させないとイケないのよ。



なんだよそれ! めんどくさいなー。どうすればいいんだよ。



未完成のゲームだったら、自分で完成させ
ちゃえばいいのよ。次のページから、一緒に
ゲームを完成させちゃいましょう!

次ページから
ゲームを完成
させよう!

Let's GO

STEP

1

STEP 2

STEP 3

矢を撃てるようにする

ゲーム完成度 0 10 20 30 40 50 60 70 80% 90 100(%)

1 プログラミングを始めよう！

まずは、次の図を目標にプログラミングをしていきましょう。

完成スクリプト例

```

        がクリックされたとき
        ずっと
        次のコスチュームにする
        0.5 秒待つ
    
```

1 「エルフ」のSpriteを選択する

スクリプト コスチューム 音

- 動き
- 見た目
- 音
- ペン
- データ
- イベント
- 制御
- 調べる
- 演算
- その他

10 歩動かす

15 度回す

15 度回す

90 度に向ける

マウスのポインターへ向ける

x座標を -144、y座標を -112

マウスのポインターへ行く

1 秒でx座標を -144 に、y座標を

x座標を 10 ずつ変える

x座標を 0 にする

y座標を 10 ずつ変える

y座標を 0 にする

「Sprite」リストで、「エルフ」というSpriteをクリックして選択する

2 「旗」がクリックされたとき」ブロックを追加する

① 「スクリプト」を選択

② 「イベント」を選択

③ ブロックをスクリプトエリアにドラッグ

3 「ずっと」ブロックを追加する

① 「制御」を選択

② ブロックをスクリプトエリアにドラッグし、「旗」がクリックされたとき」ブロックにくっつける

HINT

スプライトを確認してから作業しよう！
うっかり別のスプライトを選択していませんか？ よく確認してから作業しましょう。

Scratchではこのように、処理を表す「ブロック」を組み立ててプログラミングします。ここで作ったのは「**旗がクリックされたとき**」に処理を始めて、ゲームしている間「ずっと」何かを続ける、というスクリプトです。このブロックがないと、スプライトは「いつ処理を始めればいいのか」「いつ処理を終わらせればいいのか」がわかりません。

このゲームでは、エルフのコスチュームが変わると矢が飛んで行く仕組みになっています。次の手順を参考にして、0.5秒ごとにコスチュームが切り替わるようにしてみましょう。

4 「次のコスチュームにする」と「～秒待つ」ブロックを追加する

1 「見た目」から「次のコスチュームにする」ブロックをドラッグ。「ずっと」ブロックにくっつける

2 「制御」から「～秒待つ」ブロックをドラッグ。「次のコスチュームにする」ブロックにくっつける

3 半角英数字で「0.5」と入力

スプライトの見た目を替えるブロックは、薄紫色の「見た目」カテゴリーにあるわよ。

このコスチュームで矢を撃つようにしている

ずっと繰り返す

HINT

コスチュームとは

「コスチューム」とは、スプライトの見た目や外観のことです。スプライトに複数のコスチュームを割り当て、これらを連続で切り替えるようにスクリプトを作れば、スプライトをアニメーションのように動かせるようになります。

2 うごかして試してみよう

スクリプトが完成したら、緑の旗をクリックしてゲームを開始してみましょう。0.5秒毎にエルフのコスチュームが変わり、矢が飛んでいきます。

1 「エルフ」の sprites を選択する



旗をクリック

2 矢が飛んでいく



0.5秒ごとにコスチュームが変わり、矢が飛んでいく

う、うごく。
こいつ…動くぞ!



HINT

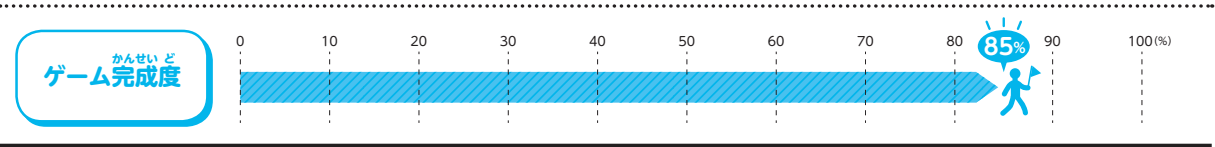
スクリプトがうまく動かないときは……

スクリプトがうまく動かない場合、ブロック同士がきちんとくっついていなかったり、数字が全角で入力されている可能性があります。もう一度確認してみましょう。

STEP 2

STEP 3

狙いをつけられるようにする



1 矢が飛んで行く方向を変えよう

矢が撃てるようにはなりましたが、まだ矢が飛ぶ方向を変えられません。次はきちんと狙いをつけられるようにしましょう。

このゲームでは、矢はエルフが向いている「向き」の方へ向かって飛んでいきます。エルフの「向き」を変えることができれば、狙いをつけられそうです。

Level1で作ったエルフのスク립トに、次のようにスク립トを追加してみましょう。

完成スク립ト例

Level1で作ったこのスク립トは残したまま

空いているスペースにスク립トを追加しよう！

1 「エルフ」の sprites を選択する

※実は ➡ は、エルフの「向き」の方向に連動して動いているだけです。

2 「がクリックされたとき」ブロックを追加する

① 「スクリプト」を選択

② 「イベント」を選択

③ ブロックをスクリプトエリアにドラッグ



3 「ずっと」ブロックを追加する

① 「制御」を選択

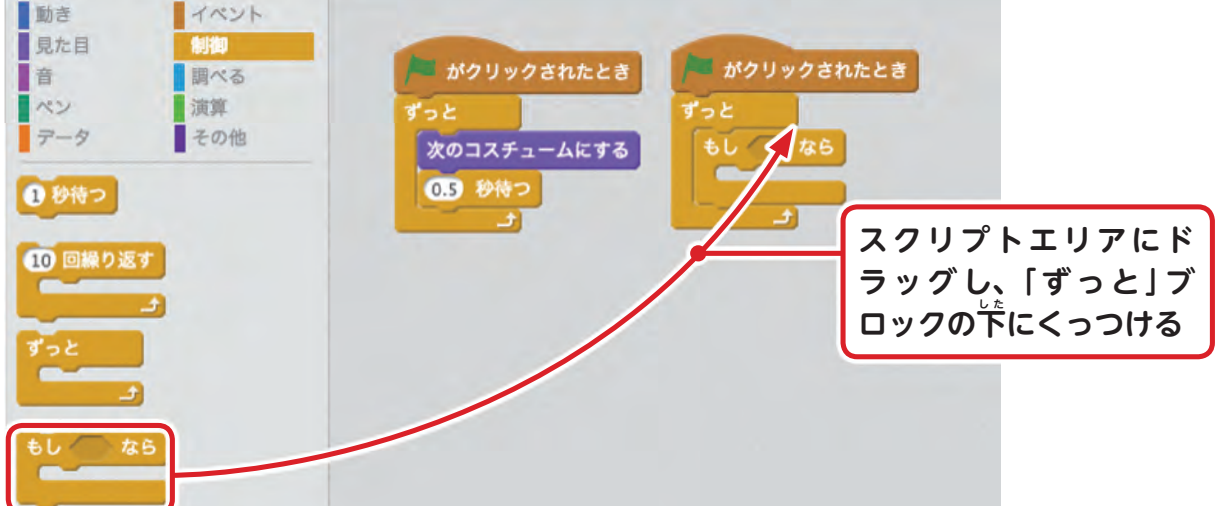
② スクリプトエリアにドラッグし、「がクリックされたとき」ブロックの下にくっつける



4 「もし~なら」ブロックを追加する

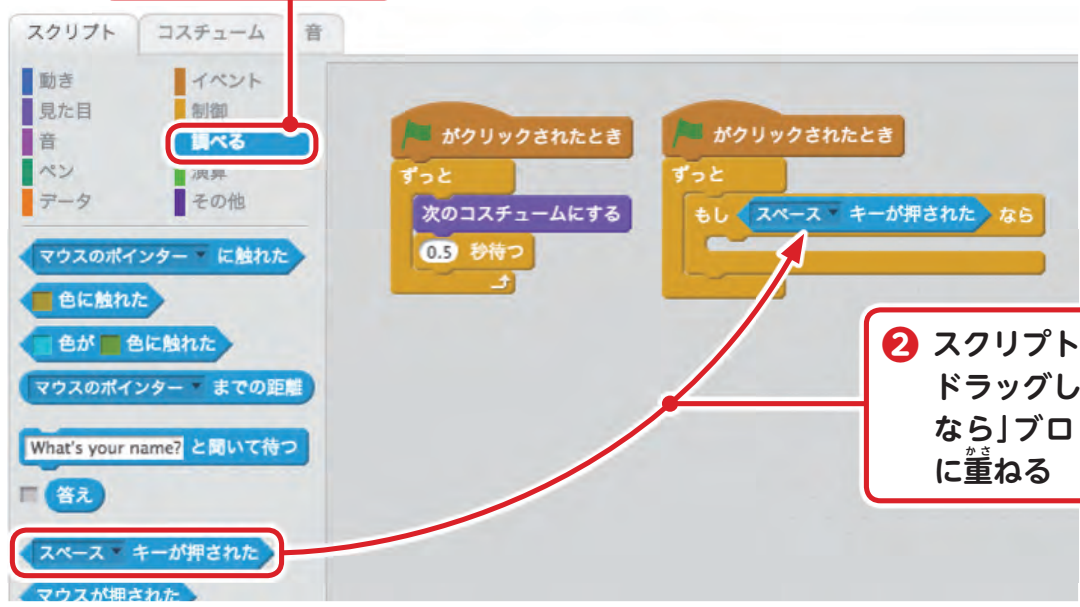
① 「もし~なら」ブロックを追加する

② スクリプトエリアにドラッグし、「ずっと」ブロックの下にくっつける



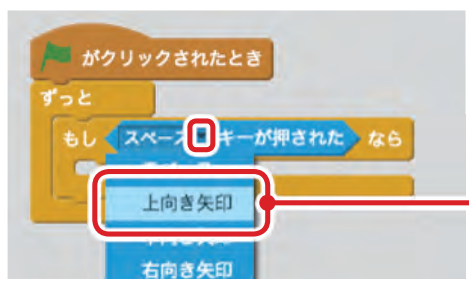
5 「上向き矢印キーが押された」ブロックを追加する

① 「調べる」を選択



② スクリプトエリアにドラッグし、「もし～なら」ブロックの上重ねる

スペース キーが押された



③ ▼をクリックし、メニューから「上向き矢印」を選択

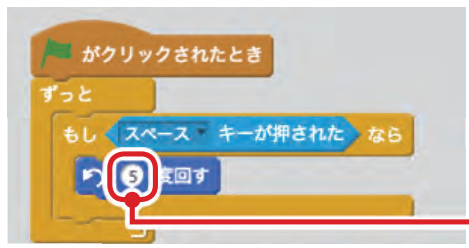
6 「時計の逆回りに15度回す」ブロックを追加する

① 「動き」を選択



② スクリプトエリアにドラッグし、「もし～なら」ブロックの下につける

15度回す

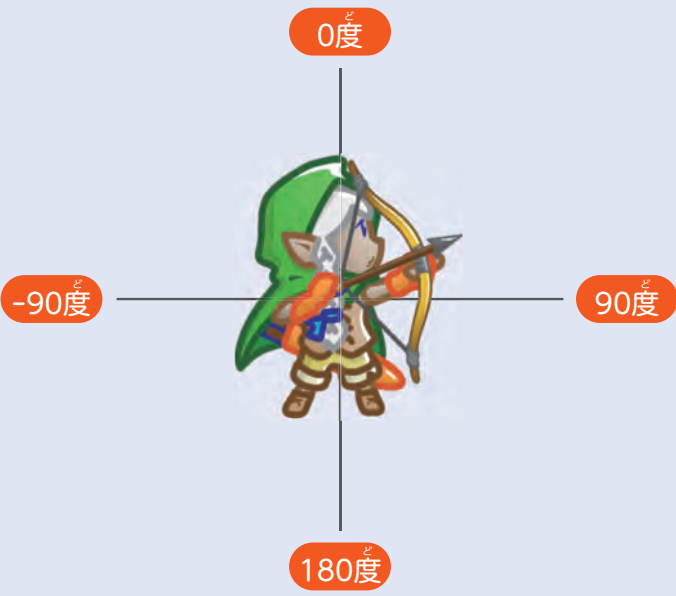


③ 半角英数字で「5」と入力

POINT

スプライトの「向き」について

スプライトはそれぞれ数値として「向き」という値を持っています。画面の上を向いていると0度、右が90度、下が180度、左が-90度となります。



90▼ 度に向ける

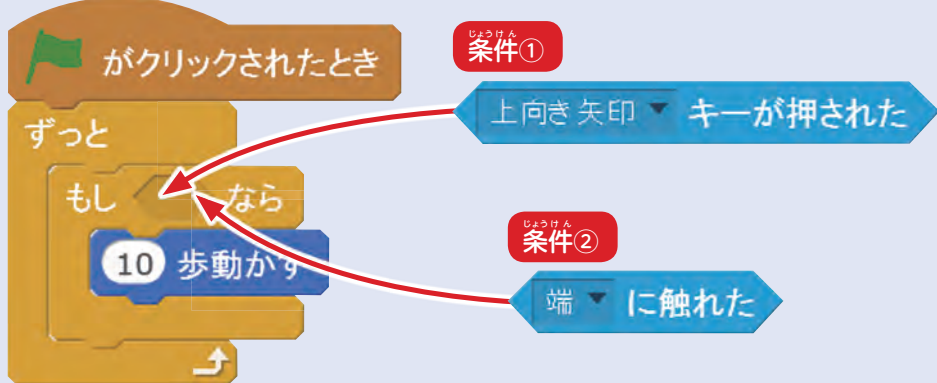
- (90) 右
- (-90) 左
- (0) 上
- (180) 下


「○○度に向ける」のブロックで確認できる

POINT


「もし～なら」について

「もし～なら」は特定の条件のときに処理を分岐させるブロックです。菱形の穴に"条件"を示す調べる系のブロックを入れて使います。条件が一致した場合にだけ、「もし～なら」ブロックで囲まれた処理を行います。



この例では、条件①の場合  キーが押されると、条件②の場合スプライトが端に触ったときに10歩進みます。条件に一致していない場合は進みません。キーを押したときにキャラクターを操作したいなど、何かのときだけ処理を行いたい場合に利用します。

2 動かして試してみよう

スクリプトが完成したら、をクリックしてゲームを開始してみましょう。



1 ゲームを実行する



をクリック


2 矢を上方向にしか操作できない



 キーを押すたびに、矢の向きが上方向に動かせるが、 キーを押しても下方向には動かせない

う、動きが
なんだかおかしい!



キーボードの  キーを押したら、エルフの向きが変わって、矢を撃つ方向が上向きに変わったはずですが。

ところが下に向けようと  キーを押しても動きません。なぜでしょうか？




HINT

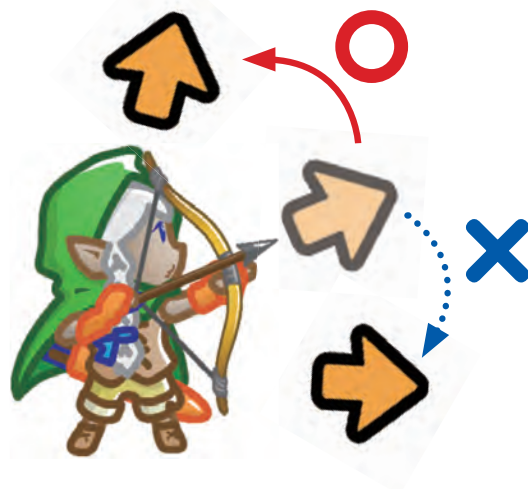
矢印キーを押したときに、どうしてエルフのSpriteの向きは変わらないの？

見た目上エルフの向きが変わってはいませんが、これは見た目は変化しないように設定されているからです。ここでは、「そういう設定がされているんだ」と思うようにしましょう。

3 矢が下向きにも回転するようにプログラミングしよう

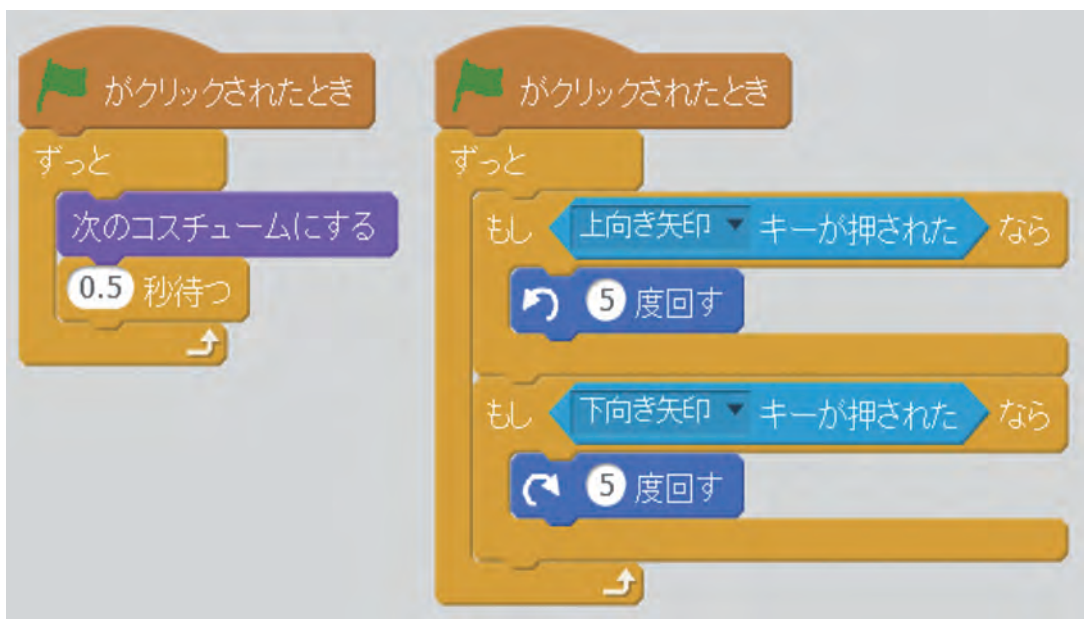
ここまでで組み上げたのは、 キーが押されたときに、反時計回りにSpriteの向きを変えるスクリプトです。


まだ下方向（時計回り）に動かすスクリプトが組み上がっていないため、狙いを下に向けられません。



P7の「矢が飛んで行く方向を変えよう」を参考にして、「下向き矢印キーが押された」なら「時計回りに5度回す」スクリプトを組みんでみましょう。

完成スクリプト例



スクリプトが組めたら、ゲームを開始して動きを試みましょう。 キーを押したときに、矢印が下のほうを向くはずですよ。

4 プログラミングを実行しよう

矢をうつ。上下に方向を変える。ができていますでしょうか。ここまでできたらゲームで遊んでみて、何点取れるか挑戦してみましょう。

1 ゲームを実行する



旗をクリック

2 矢を上下方向に操作できるようになった



↑ ↓ で矢の向きを自由に操作できるようになった

よし! 上も下も狙えるぞ! 満点取ってやるぜ!



HINT

例と違っていても大丈夫!

例と多少違っていても、矢が撃てて、上下に狙えれば大丈夫です。「必ずこうでなくてはならない」ということではありません。うまく動かないときは、もう一度スクリプトを見直してみましょう。

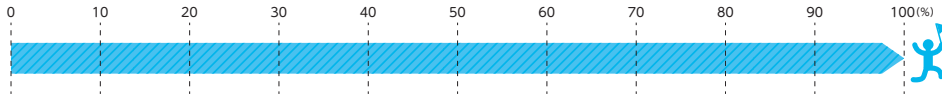
- 最初のタイミングと、どのくらい処理を続けるかのブロックはあるでしょうか。
- ブロック同士はきちんとくっついていますか。

STEP

3

自分のタイミングで 矢を撃てるようにする

ゲーム完成度



1 キーを押したときに矢が撃てるようにしよう

ゲームをプレイしていて、矢が勝手に飛ぶのが気になってきませんか？ そう思ったら、スクリプトを改造して自分の操作で矢を撃てるようにしましょう。



このゲームでは、エルフのコスチュームが、「撃つ」になったときに矢が発射される仕組みになっています。いままで組んできたスクリプトは、下の図のように、ゲームが始まるとずっとエルフのコスチュームが自動的に切り替わり、「撃つ」のコスチュームに切り替わった時点で矢が放たれていました。

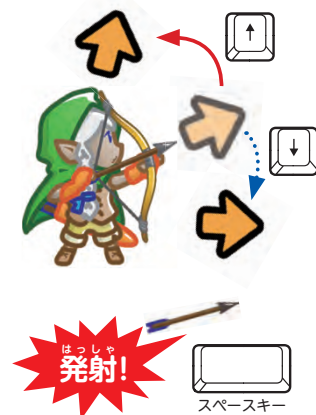
改造前のエルフのコスチューム



COLUMN

入力方法を変えてみよう！

このゲームでは、エルフがキーボードの  キーで上を向き、 キーで下を向きます。矢の発射はスペースキーです。しかし「もし～なら」の条件になっている「～キーが押されたとき」は変えることができます。パソコンゲームでは移動や方向を変える操作を左手で行うため、上・下・左・右をそれぞれ「W」「S」「A」「D」キーに割りつけられていることが多いです。もしゲームプレイがしづらいい場合は好みのキーに変えてみてもよいでしょう。



そこで、ここではスペースキーを押したら、エルフが矢を放つようにスクリプトを改造していきましょう。

改造前の「次のコスチュームにする」ブロックの前に、「もし～なら」ブロックと「スペースキーが押された」ブロックが必要になるのがわかります。またスペースキーを押す前、つまりエルフが矢を放つ動作を行う前の「棒立ち」状態のコスチュームもあったほうが自然です。エルフの動作の流れを図にまとめると、

改造後のエルフのコスチューム



改造前のスクリプトに対して、赤枠部分のスクリプトを追加しなければいけません。上の図のとおりブロックを組み直すと、以下ようになります。

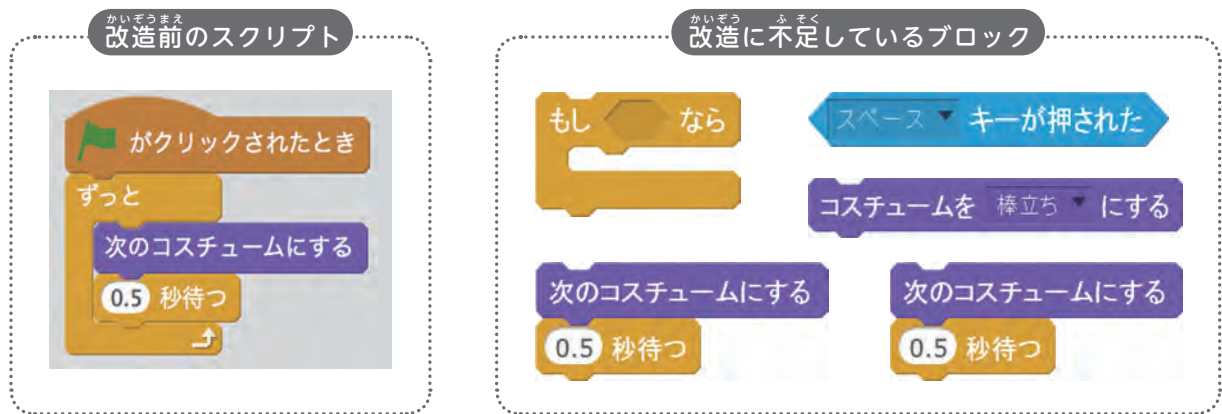
完成スクリプト例

改造前

▶

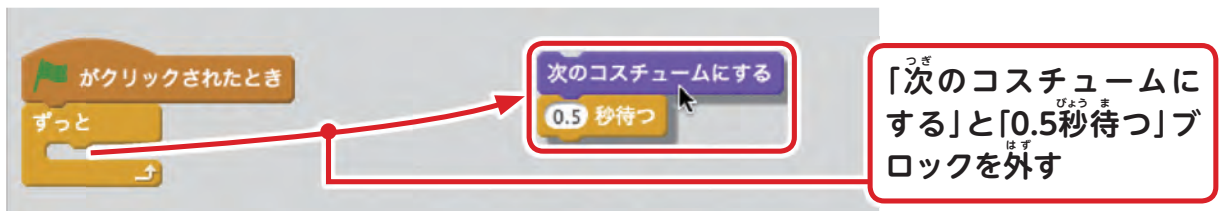
改造後

最初のスクリプトを改造する点に注意しましょう。

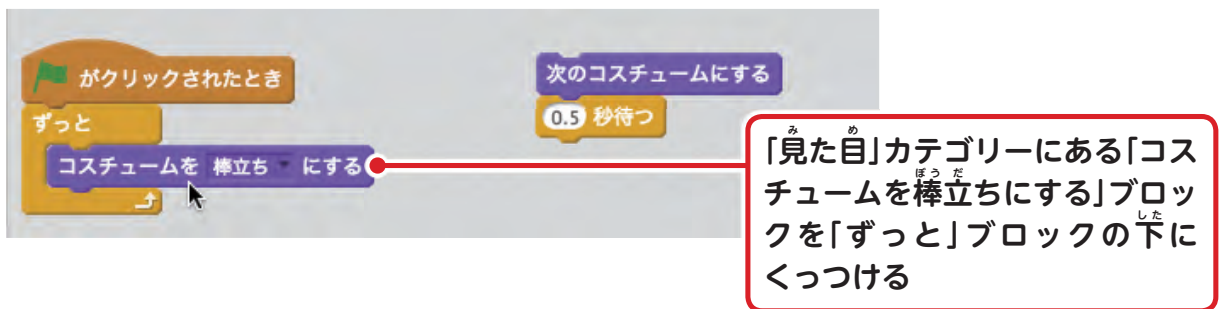


スクリプトは主に「イベント」のカテゴリにある上が丸いブロックを始めとして、上から順番に処理されます。最初から順番に一個ずつ確認して、足りないブロックを追加していきます。

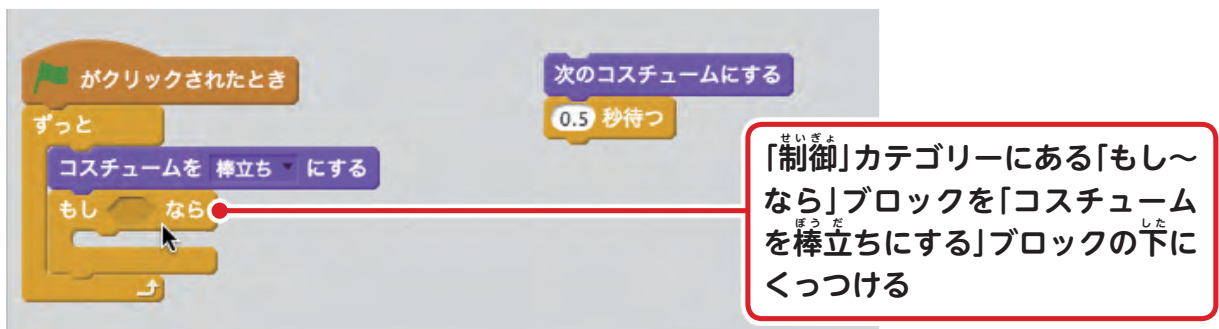
1 「次のコスチュームにする」「0.5秒待つ」ブロックを外す



2 「コスチュームを棒立ちにする」ブロックを追加する



3 「もし～なら」ブロックを追加する



4 「スペースキーが押された」ブロックを追加する

「調べる」カテゴリーにある「スペースキーが押された」ブロックを「もし～なら」ブロックの上にくっつける

5 ブロックを複製する

「次のコスチュームにする」と「0.5秒待つ」を右クリックし、「複製」を選択

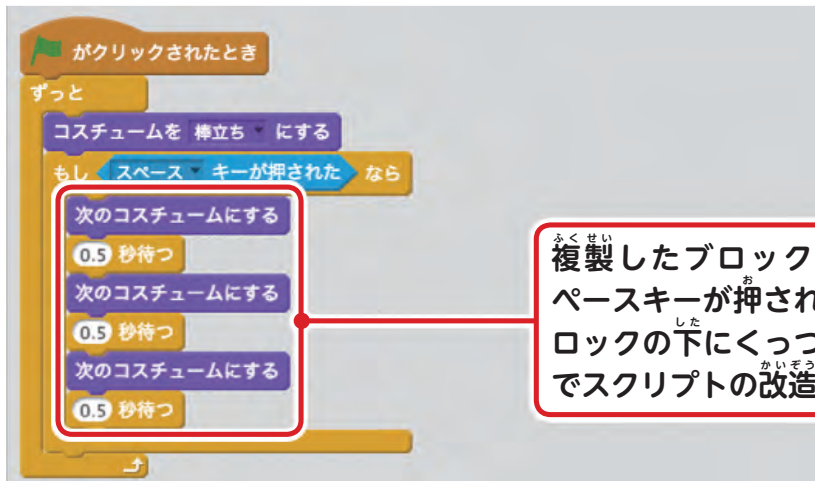
6 複製したブロックをくっつける

複製した「次のコスチュームにする」と「0.5秒待つ」をくっつける

7 5～6の操作を繰り返す

5～6の操作を繰り返し、左図のようにブロックを作る

8 複製したブロックを「もしスペースキーが押されたなら」ブロックにくっつける



複製したブロックを「もしスペースキーが押されたなら」ブロックの下にくっつける。これでスクリプトの改造は完了

HINT

ブロックを付ける位置を間違えてしまったときは……

ブロックを放すタイミングを誤って、思っていたのとは違う位置にブロックがくっついてしまうことがよくあります（図1）。

Scratchはつかんだブロックの下についているブロックも一緒にくっついてきます。このまま思ったおりに組み上げるのは難しいです（図2）。

誤って組み込んだブロックは、一度スクリプトエリアの空いているところに置いて、一番下のブロックから分解しましょう（図3）。

分解したブロックを一つずつ正しいと思う箇所へ組みこんでいきます。もしまた間違えたら間違えた部分を外してやりなおします（図4）。

操作を誤ってしまうとあわててしまいがちですが、落ち着いて綺麗に整理して、自分にとってわかりやすい状態にしてから改めて組み立て始めることが大切です。

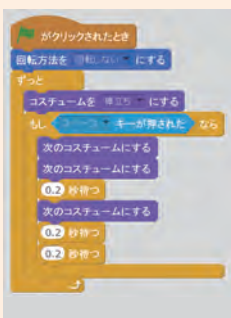


図1

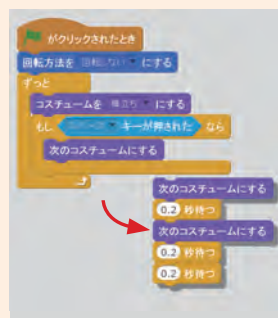


図2

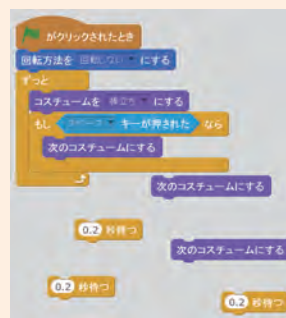


図3

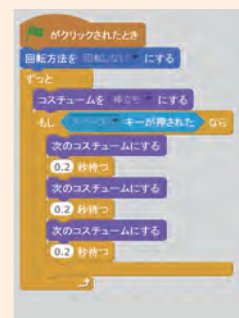


図4

POINT

スクリプトの処理の順番は？

スクリプトは先頭から順番に処理されます。わからなくなったら始めから順に確認していきましょう。声に出しながらブロックを読んでいくとわかりやすいです。

2 動かして試してみよう

ここで組んだスクリプトでは「スペースキーが押された」ことを条件にしています。キーボードのスペースキーを押してみましょう。

1 ゲームを実行する



旗をクリック

2 スペースキーを押すと矢が放たれる



スペースキーを押すと、矢が放たれるようになった

ずっと最初のコスチューム「棒立ち」になっていますが、スペースキーが押されたときだけでなく、コスチュームを3回変えることで、アニメーションしながら矢が発射されます。



HINT

全角入力モードと、半角入力モードの違いに気をつけて！

全角入力モードになっていると、スペースキーが反応しません。スペースキーが動かないときは入力モードも確認してみてください。

同じ処理の繰り返しがあるときは、「～回繰り返す」を使おう！

このスクリプトをよく見ると「次のコスチュームにする」と「0.5秒待つ」という同じ処理を3回繰り返しています。

同じ処理を3回も入力するのは「面倒くさいなあ」と感じたあなたはプログラマーに向いています。実は「～回繰り返す」のブロックを使って次のように簡単に表現することができます。

「もし～なら」ブロックの中から、繰り返している部分を抜き出して分解します。

「～回繰り返す」の数字の箇所を「3」と入力して、「もし～なら」ブロックの中に入れ、「次のコスチュームにする」と「0.5秒待つ」のブロックを1つだけ戻せば完成です。

どうでしょう。すっきり簡単になりましたね。

① いったん外す

② 「3」と入力

③ ドラッグ

④ ドラッグ





HINT

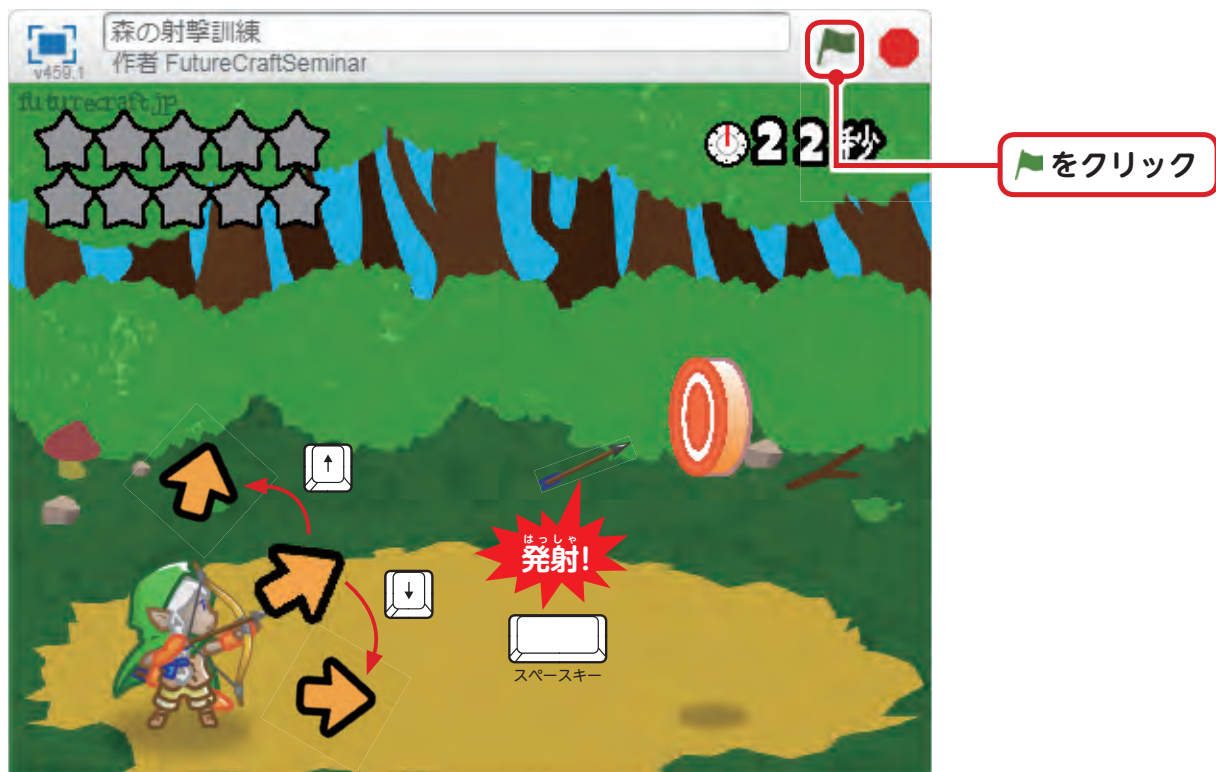
不要なブロックは どうすれば消せるの？

不要なブロックはブロックパレットに戻すことで、消すことができます。

不要なブロックは、このエリアにドラッグする

3 動かして試してみよう

これで、  キーで狙いをつけて、スペースキーで矢が撃てるようになりました。おめでとう。「森の射撃訓練」完成です。ゲームを遊んで何点取れるかチャレンジしましょう！



完成スクリプト例

```

    旗がクリックされたとき
    ずっと
    コスチュームを 棒立ち にする
    もし スペース キーが押された なら
        3 回繰り返す
        次のコスチュームにする
        0.5 秒待つ
    旗がクリックされたとき
    ずっと
    もし 上向き矢印 キーが押された なら
        5 度回す
    もし 下向き矢印 キーが押された なら
        5 度回す
    
```


ゲームを改造して完全クリアにチャレンジしよう!

パーフェクトチャレンジ!

Perfect
条件

得点を10点以上獲得せよ!

ゲームが終わると、得点に応じて BAD/OK/GOOD/Perfect の結果が表示されます。「森の射撃訓練」の Perfect の条件は、10点以上獲得することです。頑張っ**て** Perfect を目指しましょう!



ぐわあああ! 8点までいったのに! このゲーム難しすぎるぞ!



それなら、ゲームを改造してみればいいじゃない。



そんなことしていいのか? チートじゃん!



いいに決まってるじゃない。このゲームの開発者はあんたでしょ。
どんなゲームにするかは自分で決めればいいのよ。

※チートとはコンピュータゲームにおいて本来とは異なる動作をさせる行為です。「ズル」あるいは「騙す」という意味の英単語cheatからきています。

完成させたゲームはそのままではPerfectは難しいでしょう。スクリプトを改造して、たとえば矢の速度を速くしたり、的が小さくならないようにすることができます。

改造のヒント①

ゲーム作りではエルフのスクリプトを組みましたが、改造ではその他のスプライトも変えてみましょう。スプライトリストで矢を選択してみます。スクリプトエリアが矢のスクリプトになったのがわかります。



「エルフ」が選択されている

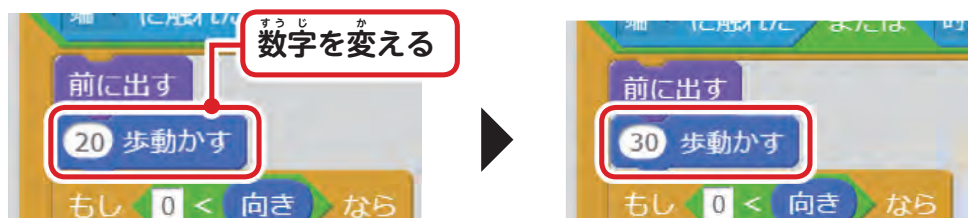


「矢」が選択されている

矢の速度を上げる改造をしてみましょう。「矢の速度が上がる」ということは、何が変っているのでしょうか？見た目が変わっていますか？音が出ていますか？

そうです。「動き」が変わっていますね。変わるのが「動き」だとすると、改造するのは青いブロックということになります。矢のスクリプトの中で青いブロックで、前に動かしていそうなブロックを探しましょう。

矢のスクリプトの中で、前に動くブロックは「20歩動かす」だけです。この動く歩数を変えれば矢が動くスピードが速くなります。数字を変えて、どう動きが変わったかゲームをスタートして試してみましょう。



Scratchのステージは、右端から左端までが480。上から下までが360しかありません。あまり大きな数字を入れるとすぐに画面から飛び出してしまいますので気をつけましょう。

改造のヒント②

「改造のヒント①」では組み立てられているスクリプトの数字を変える改造を試みましたが、次は新しいブロックを追加して改造してみましょう。

「的」のSpriteを選択してください。的は得点を重ねると小さくなっていきますが、小さくならないように改造してみたいと思います。

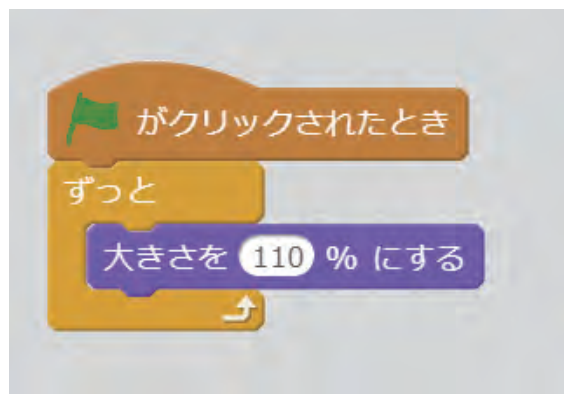
「『的』の表示が小さくならないようにする」ということは、どのカテゴリーのブロックを使えばいいのでしょうか？

そうです。「見た目」です。「見た目」のカテゴリーの中に「大きさを何々%にする」というブロックがあります。これで大きさを変えられます。



大きさを**変**えるブロックだけでは**思**ったとお**り**動きません。スクリプトの処**理**が行われるには、処**理**を**開**始する**条**件と、どの**く**らい処**理**を**続**けるかをコンピュータに**教**えて**あ**げる**必**要があるからです。

今回は「ゲームが**始**まったら**ず**っと**同**じ**大**きさで**い**て**ほ**しい」ので、「**的**」には**以**下の**よ**うに**プ**ログラミング**し**ま**し**ょう。



実はこのゲームは10点以上の**得**点を取**る**こともできます。パーフェクトが**取**れても、さら**に**いろいろな**改**造をして、**画**面から**星**があ**ふ**れる**く**らいの**得**点**を**自**指**して**み**て**く**ださい。



へーん。コーサクウ！
あんたも早くパーフェクトを取**っ**て**み**な**さ**いよ！



くっそー！ すごい**改**造して10000点**と**って**や**る**か**ら**見**て**ろ**よー！



HINT

「少し**変**えて少し**試**す」を**繰**り返**し**ま**し**ょう

ブロックの**数**字を**変**える**改**造と、**新**しく**ブ**ロックを**追**加する**改**造を**学**びました。矢**や**的**以**外にも、**エ**ルフが**矢**を**撃**つ**待**ち**時**間**や**制**限**時**間**の**数**字を**変**えたり、矢**が**的**を**追**い**か**け**る**よ**うに**ブ**ロックを**追**加することもできます。「少し**変**えて少し**試**す」を**繰**り返**し**て、**い**ろ**い**ろな**改**造を**し**て**み**ま**し**ょう。

