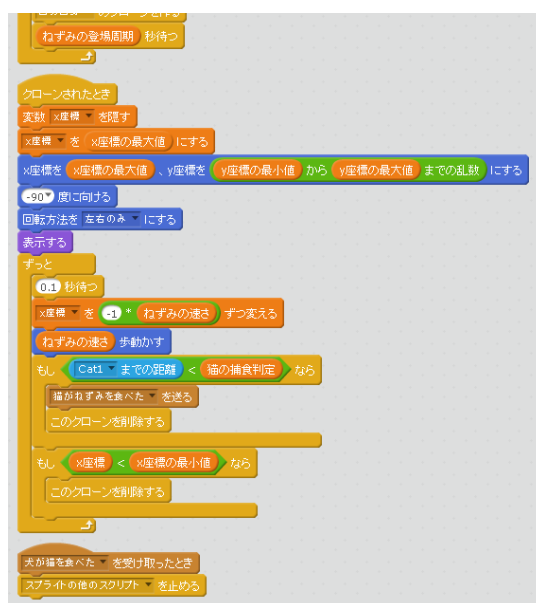


学芸カフェテリア講座

ゲーム作りで学ぶ、

先生のためのScratch入門講座 ～プログラミング必修化を楽しく先取りしよう！～



(画像はプログラミングの例です。この講座で作成するプログラムとは異なります)

東京学芸大学 気象学研究室
佐藤尚毅

はじめに

小学校で 2020 年度からの実施が見込まれる新しい教育課程においては、プログラミングを必修とする方向で議論が進められています。「小学校段階におけるプログラミング教育の在り方について（議論の取りまとめ）」（小学校段階における論理的思考力や創造性、問題解決能力等の育成とプログラミング教育に関する有識者会議）には以下のように述べられています。

小学校におけるプログラミング教育が目指すのは、前述のように、子供たちが、コンピュータに意図した処理を行うよう指示することができるということを体験しながら、身近な生活でコンピュータが活用されていることや、問題の解決には必要な手順があることに気付くこと、各教科等で育まれる思考力を基盤としながら基礎的な「プログラミング的思考」を身に付けること、コンピュータの働きを自分の生活に生かそうとする態度を身に付けることである。

プログラミング教育の実施に当たっては、コーディングを覚えることが目的ではないことを明確に共有していくことが不可欠である。また、「主体的・対話的で深い学び」の実現に資するプログラミング教育とすることが重要であり、一人で黙々とコンピュータに向かっているだけで授業が終わったり、子供自身の生活や体験と切り離された抽象的な内容に終始したりすることがないように、留意が必要である。楽しく学んでコンピュータに触れることが好きになることが重要であるが、一方で、楽しいだけで終わっては学校教育としての学習成果に結びついたとは言えず、子供たちの感性や学習意欲に働きかけるためにも不十分である。

（文部科学省のウェブサイトより。下線は著者が付した。）

簡潔にまとめれば、

- ・児童が自分で考えてプログラムを作る。
- ・ただしコードは書かない。

という 2 つの条件を満たしたプログラミング教育を行なってくださいということです。この講座で扱う Scratch は、コードをキーボードでタイプするのではなく、画面上でブロックを組み合わせることによってプログラムを作成する教材であり、まさに、上記の目的に合ったものです。Scratch 以外にも類似した機能を持つ教材はありますが、基本的な考え方は同じです。今後、小学校でのプログラミング教育では、ブロックを組み合わせるという方法が標準になっていくと思います。この講座で学んだ内容は、利用する具体的な教材を問わず、小学校でのプログラミング教育において、普遍的に役立つものです。この講座は、初心者者を前提にして、限られた時間の中で開講しているため、プログラムの例をあらかじめ示していますが、各自のアイデアで自分だけのプログラムを作成していくとさらに楽しく学べると思います。

第 1 部 : アプリケーションのインストール

アプリケーションのダウンロード

まず、

<https://scratch.mit.edu/scratch2download/>

から Scratch 2 をダウンロードします。次に、

<https://get.adobe.com/jp/air/>

から Adobe AIR をダウンロードします(上記の Scratch のダウンロードサイトからリンクあり)。

※この講座では、CD-ROM または USB メモリで配布します。

☞ 配布したデータの中の「アプリケーション」というフォルダを開いてください。Windows の場合は「Windows」というフォルダに、macOS の場合は「Mac」というフォルダに移動してください。

アプリケーションのインストール

まず、Adobe AIR をインストールします。これは Scratch を動作させるために必要なアプリケーションです。インストールするためには、ダウンロードした AdobeAIRInstaller.exe (AdobeAIR.dmg) をダブルクリックします。次に、Scratch 2 をインストールします。インストールするためには、ダウンロードした Scratch-448.exe (Scratch-448.dmg) をダブルクリックします。

起動してみよう

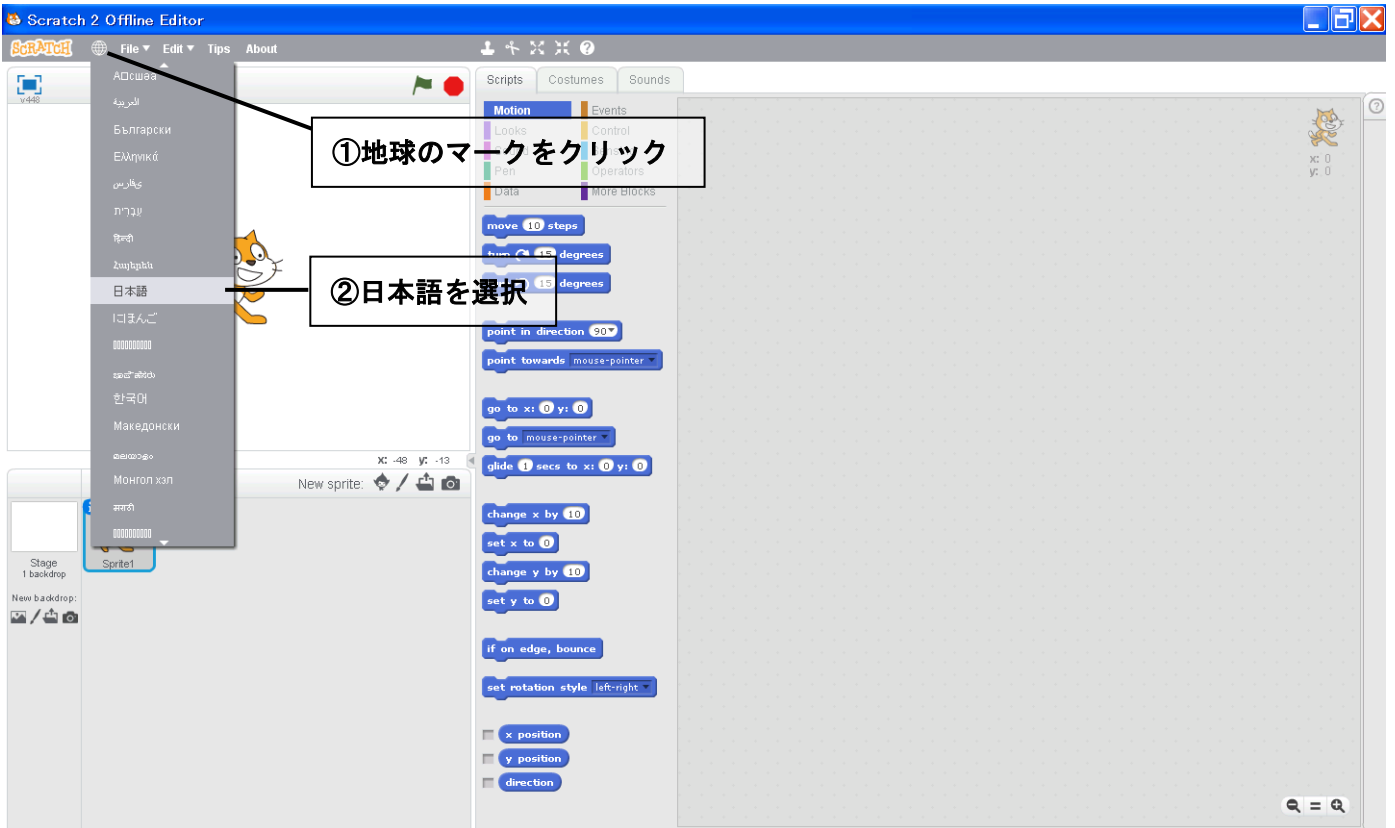
☞ デスクトップ上のアイコン  をダブルクリックします。

※終了するときは、メニューバーの「ファイル」をクリックして「終了」を選びます。

日本語の設定

もし日本語が表示されなかったら…

☞メニューバーにある地球のマークをクリックして「日本語」を選びます。

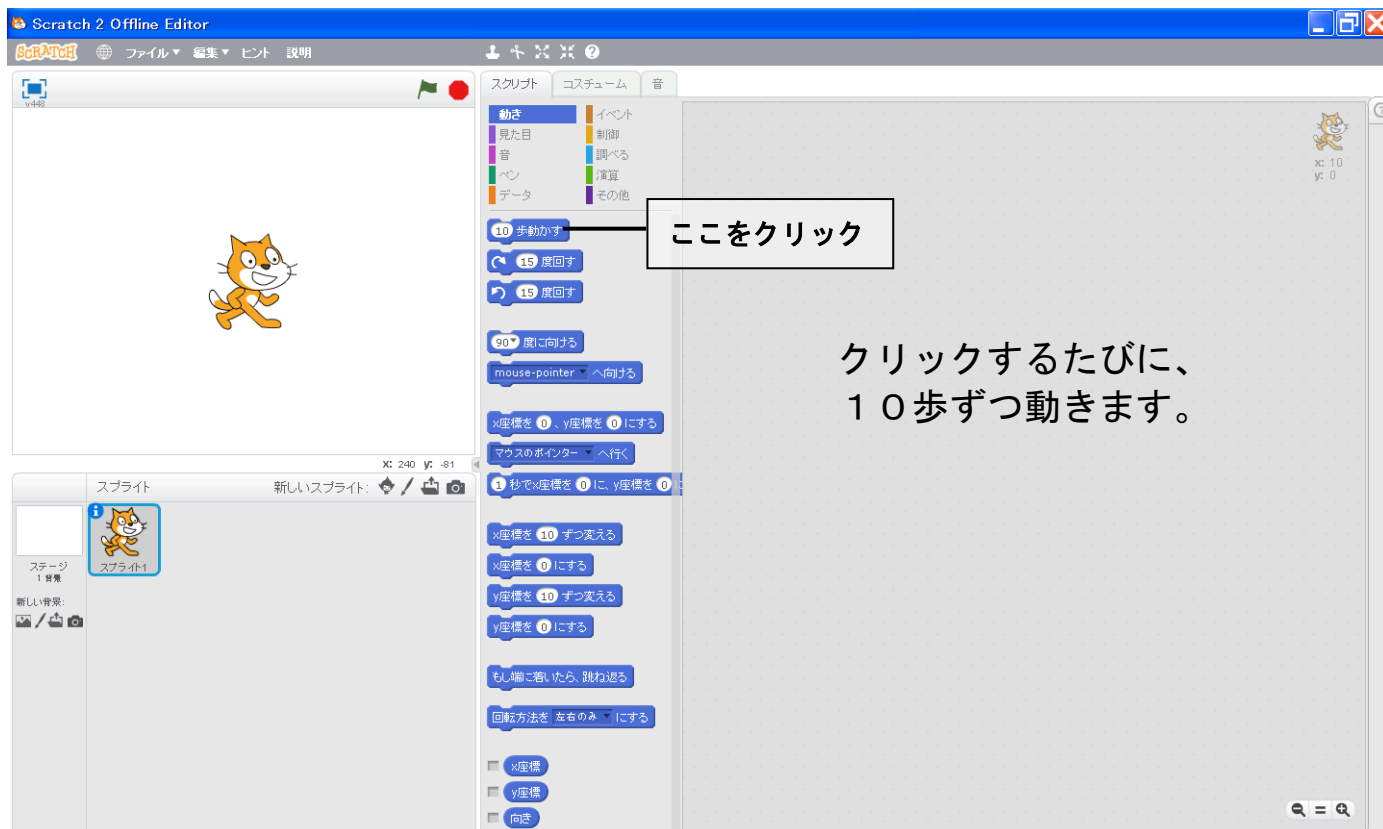


※低学年の児童の場合、「にほんご」を選ぶと、簡単な日本語表現にすることができます。

第2部：プログラミング

まずは、命令を試してみよう

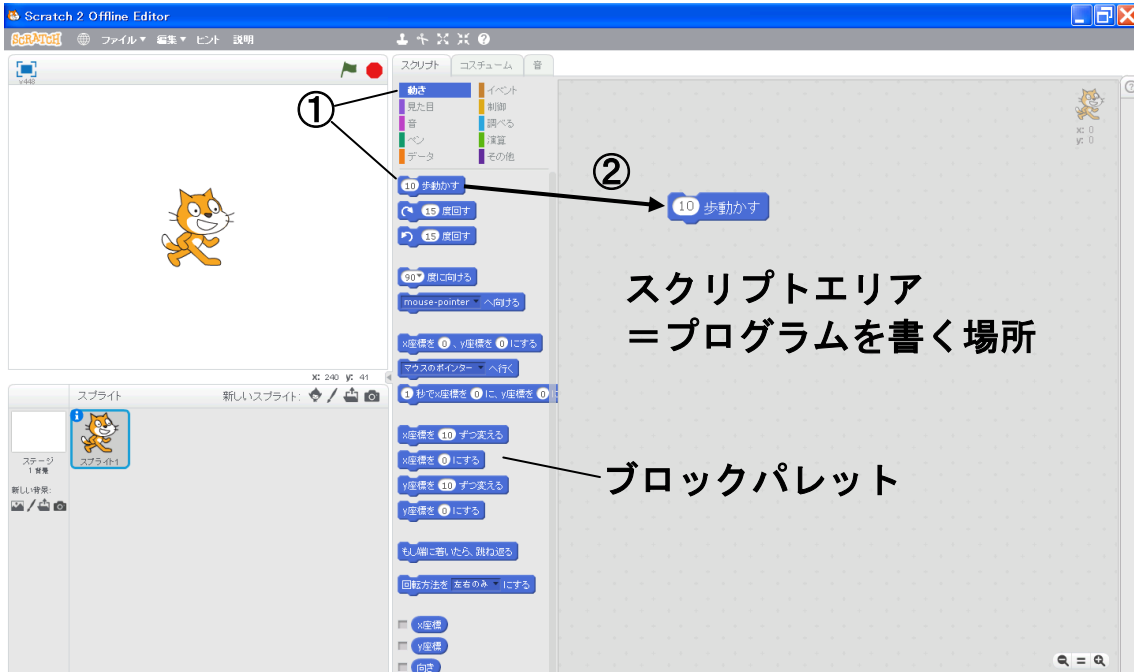
命令の内容：ネコを10歩動かす



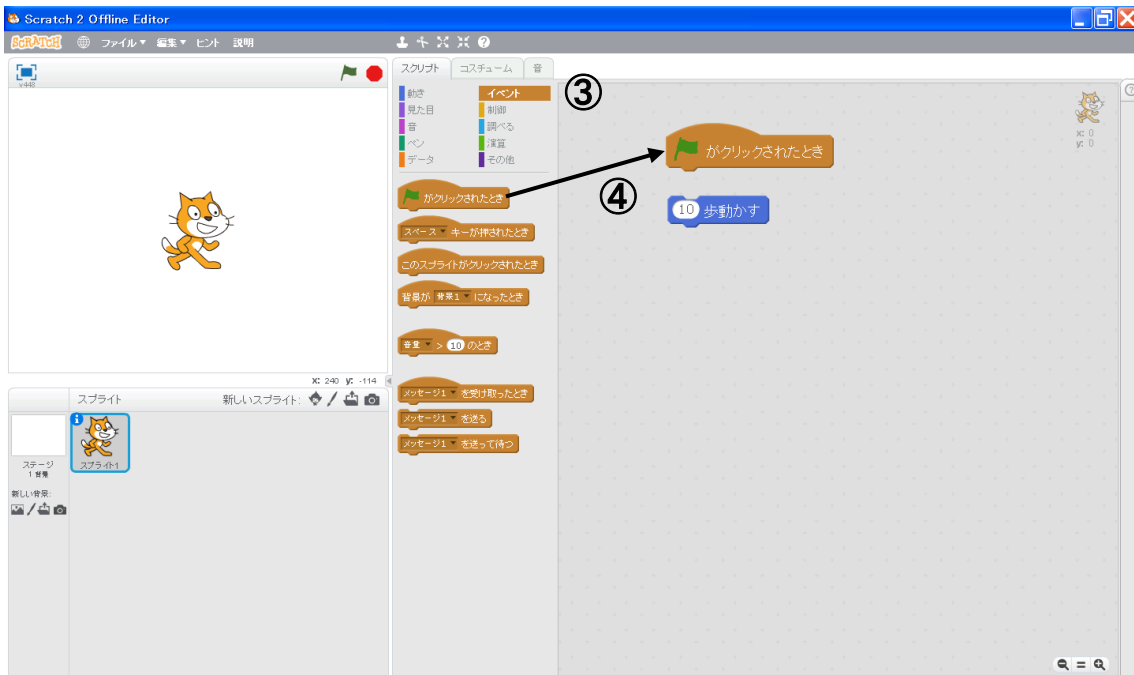
次に、簡単なプログラムを書いてみよう

プログラムの内容：ネコを10歩動かす

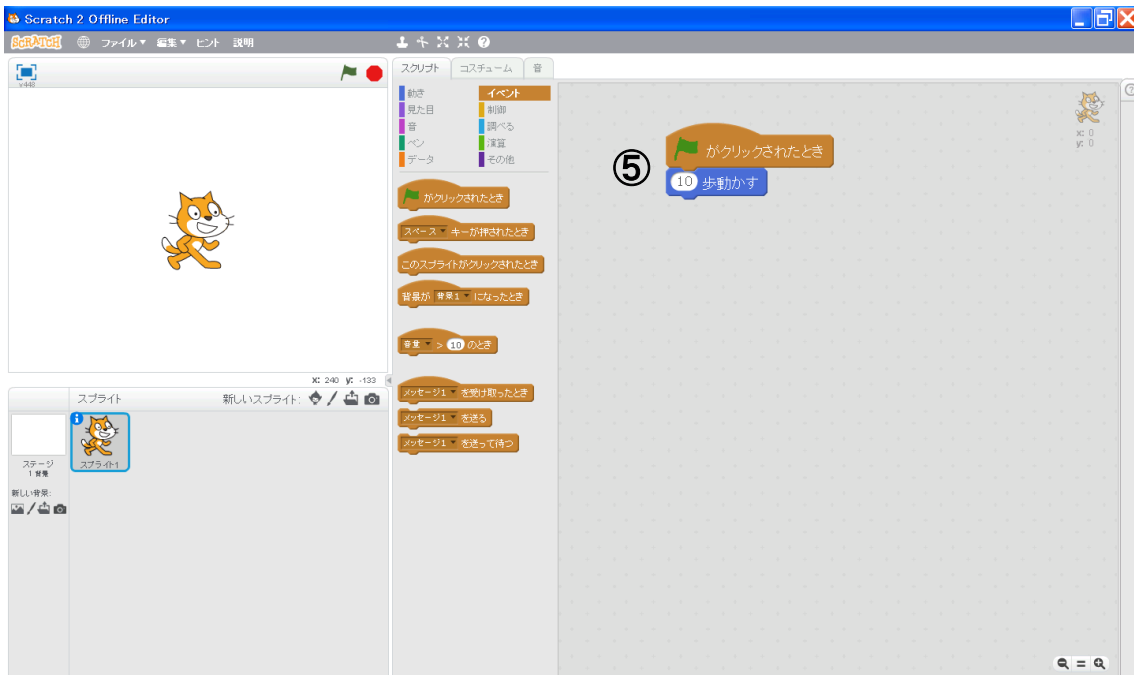
- ①中央のブロックパレットの中の動きから10歩動かすを選んで、
- ②右側のスクリプトエリアにドラッグする（クリックをしたまま引きずる）。



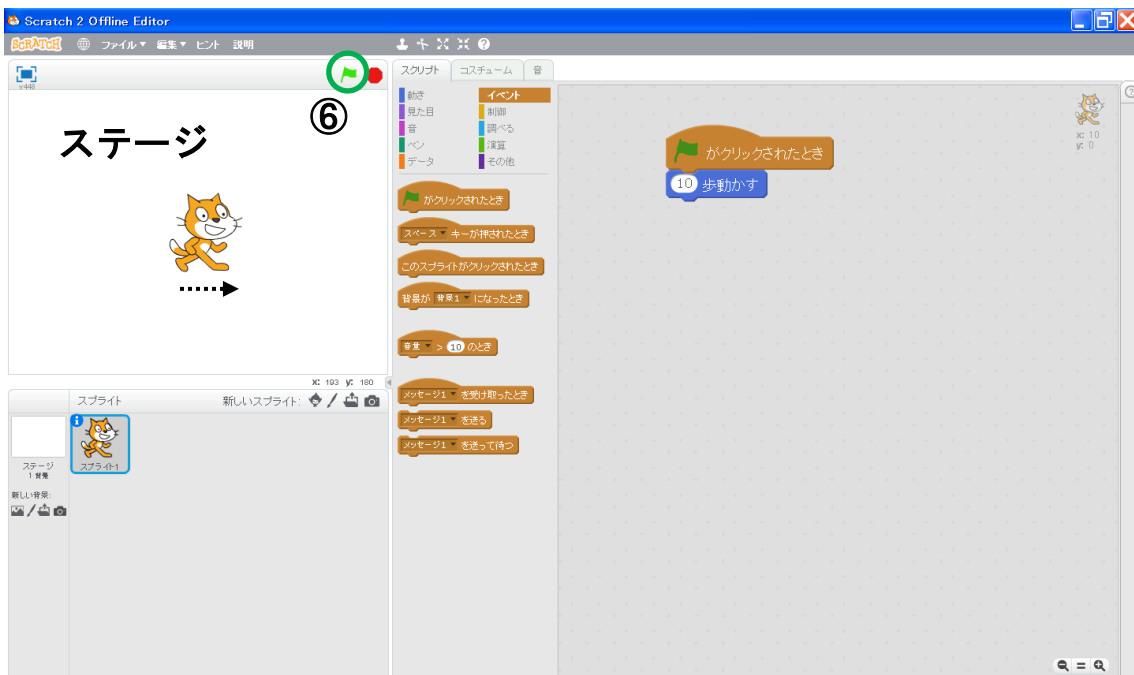
- ③中央のブロックパレットの中のイベントから緑の旗がクリックされたときを選んで、
- ④右側のスクリプトエリアにドラッグする。




⑤右側のスクリプトエリアで、10歩動かすと緑の旗がクリックされたときをくっつける。



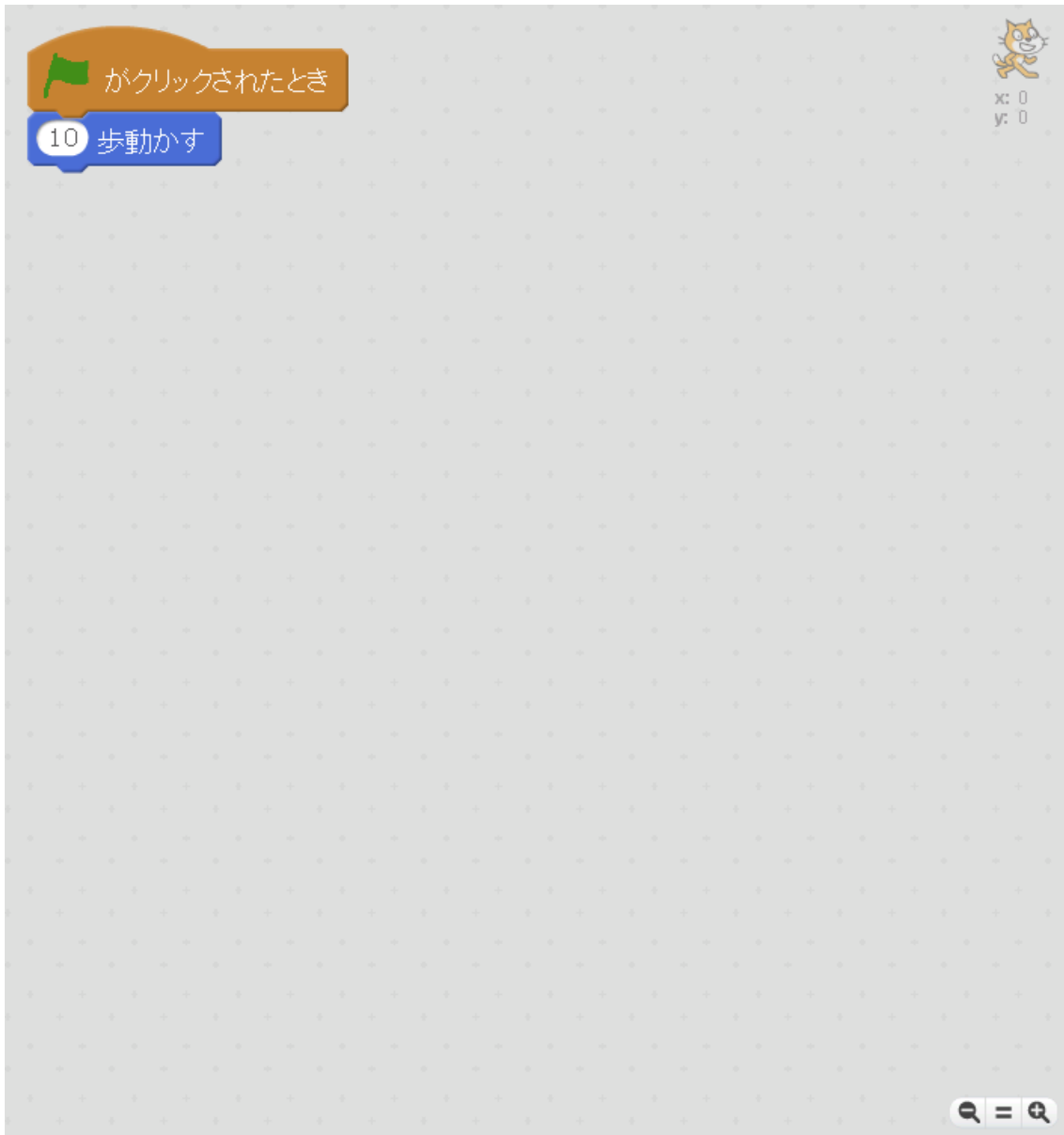
⑥左側のステージで緑の旗をクリックする。



※ステージ左上のアイコン  をクリックすると全画面表示にすることができます。

①ネコを動かそう

☞ブロックパレットからブロックを選んで、スクリプトエリアでプログラムを書こう。

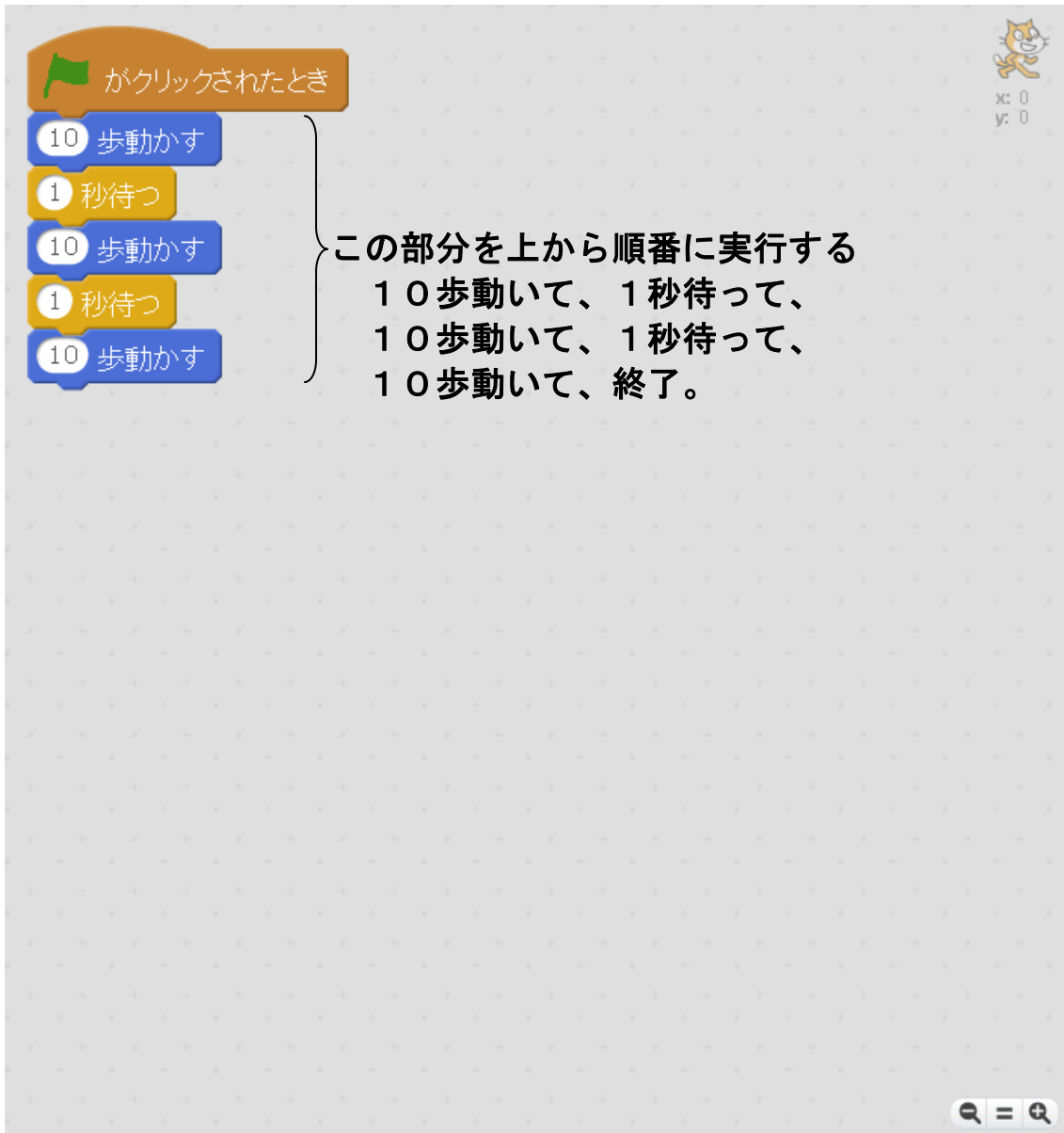


☞書き終わったら、ステージで**緑の旗**をクリックして実行してみよう。

☞実行し終わったら、メニューバーの「ファイル」から「名前をつけて保存」を選んで、プログラムを保存しよう。

②ネコを3回連続して動かそう

👉ブロックを追加してプログラムを完成しよう。



がクリックされたとき

10 歩動かす

1 秒待つ

10 歩動かす

1 秒待つ

10 歩動かす

この部分を上から順番に実行する
10歩動いて、1秒待つて、
10歩動いて、1秒待つて、
10歩動いて、終了。

x: 0
y: 0

👉書き終わったら、ステージで**緑の旗**をクリックして実行してみよう。

👉実行し終わったら、メニューバーの「ファイル」から「名前をつけて保存」を選んで、プログラムを保存しよう。①とは別の名前で保存するとよいでしょう。

次に、10回連続して動かしてみましょう。「10歩動かす」「1秒待つ」を10回書いてもよいのですが、少し大変そうですね。。。 →次のページへ続く

③ネコを10回連続して動かそう

☞ 「〇回繰り返す」を追加して同じ処理を決められた回数だけ反復します。



※〇の中の数字は変更できます。数字の部分をクリックして、キーボードから入力してください。

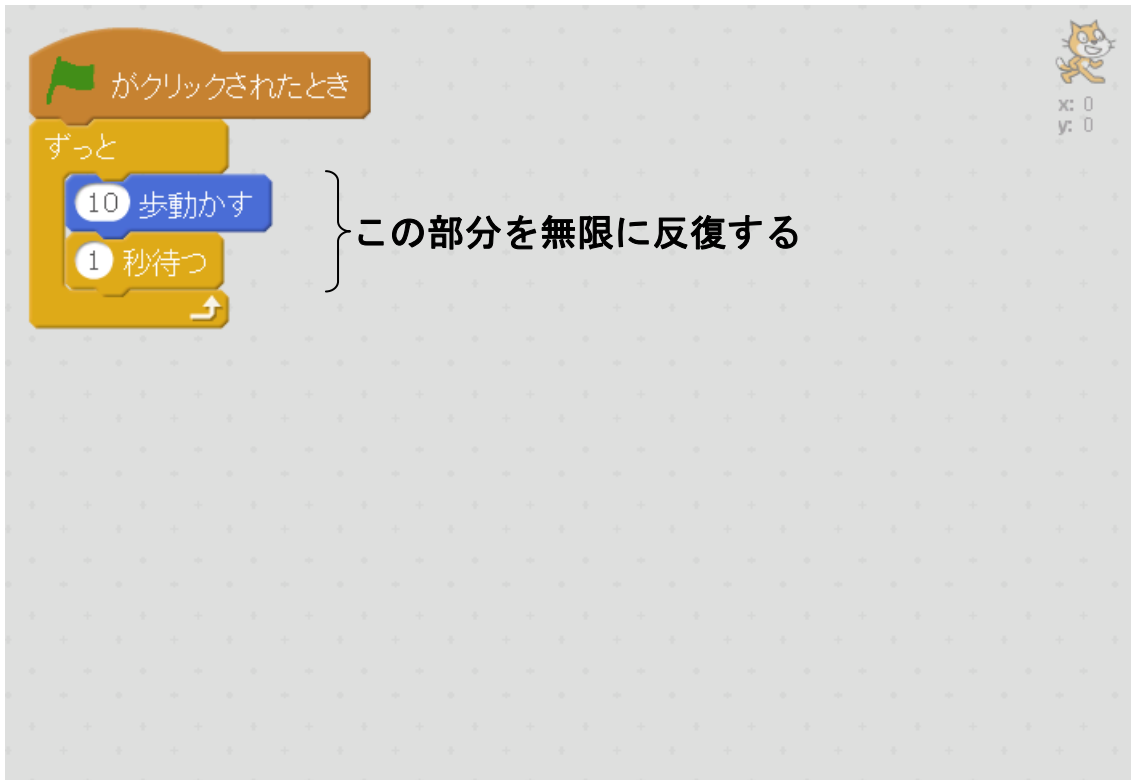
プログラミング言語では、たとえば、次のように書きます：

```
for (i=1; i<=10; i++) {  
    動かす (10. 0);  
    待つ (1. 0);  
}
```

※代表的なプログラミング言語であるCを例にして、有限回の反復処理の書き方を概念的に示しています。日本語の部分は架空の文法です。

④ネコをずっと連続して動かそう

☞「ずっと」を追加して同じ処理を無限に（終了ボタンが押されるまで）反復します。



☞終了するときには、ステージの右上の**赤いボタン**（終了ボタン）をクリックする。



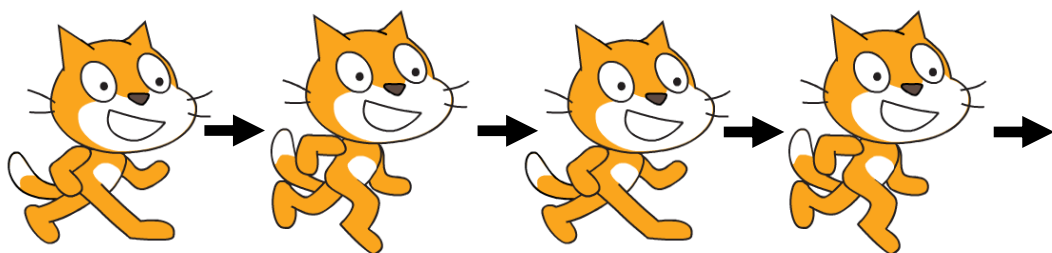
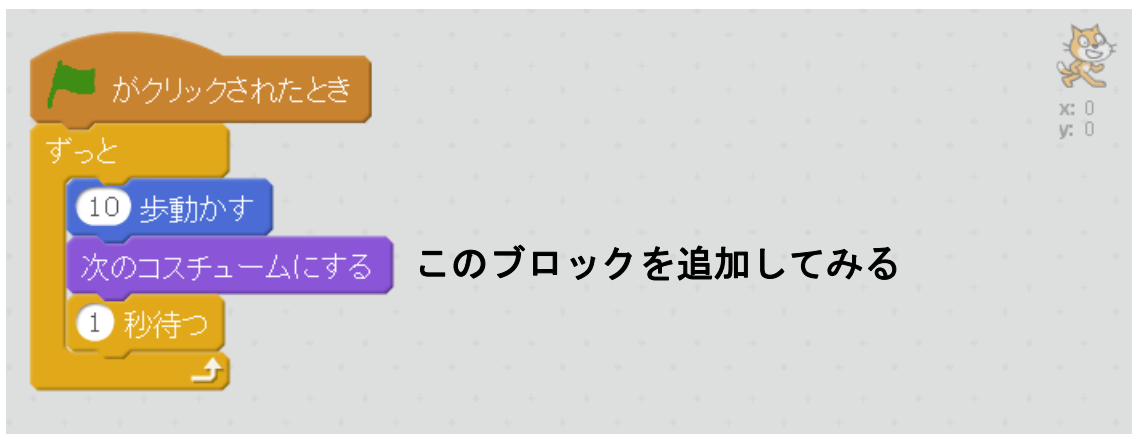
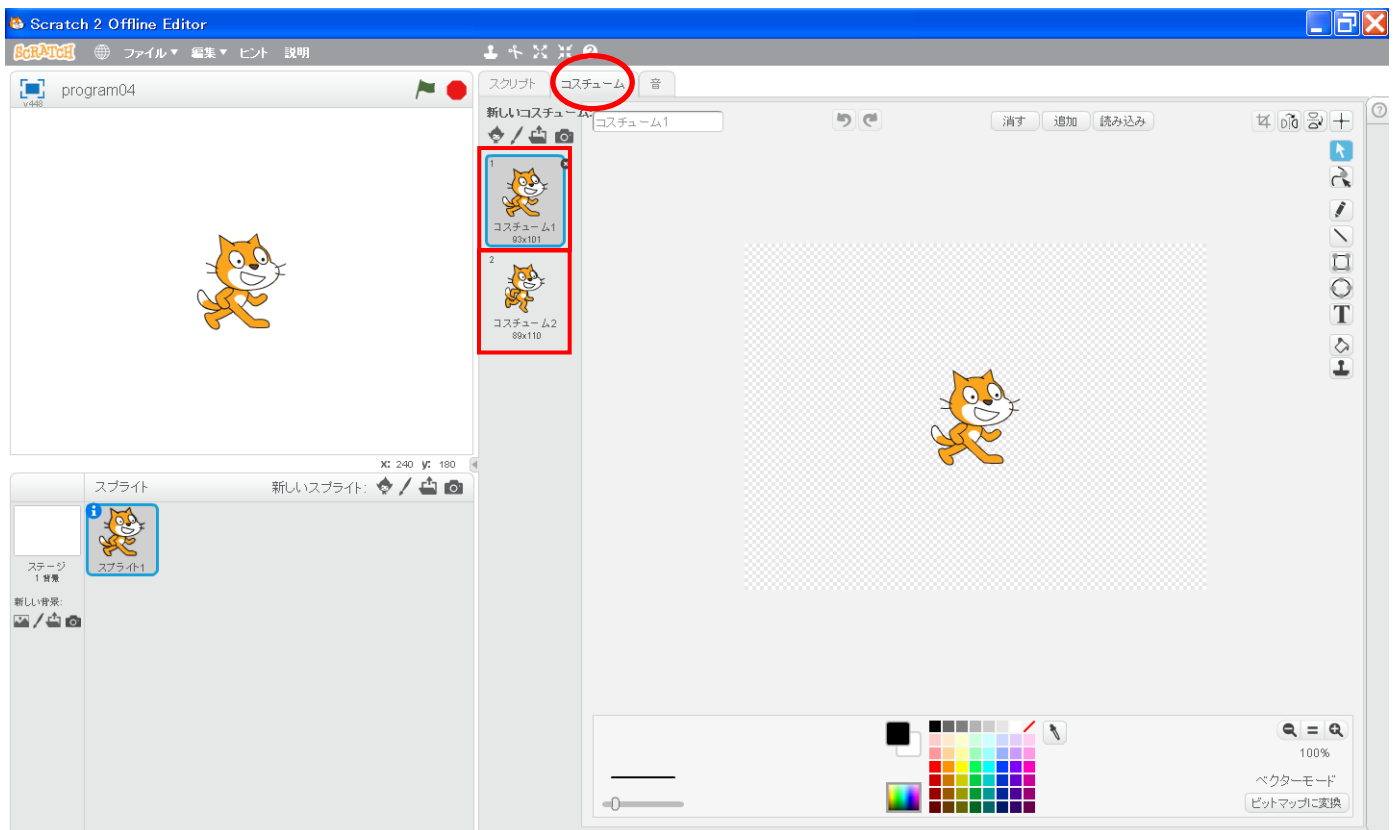
プログラミング言語では、たとえば、次のように書きます：

```
while (1) {  
    動かす (10. 0);  
    待つ (1. 0);  
}
```

※代表的なプログラミング言語であるCを例にして、無限回の反復処理の書き方を概念的に示しています。日本語の部分は架空の文法です。

寄り道：歩いているように見せる方法

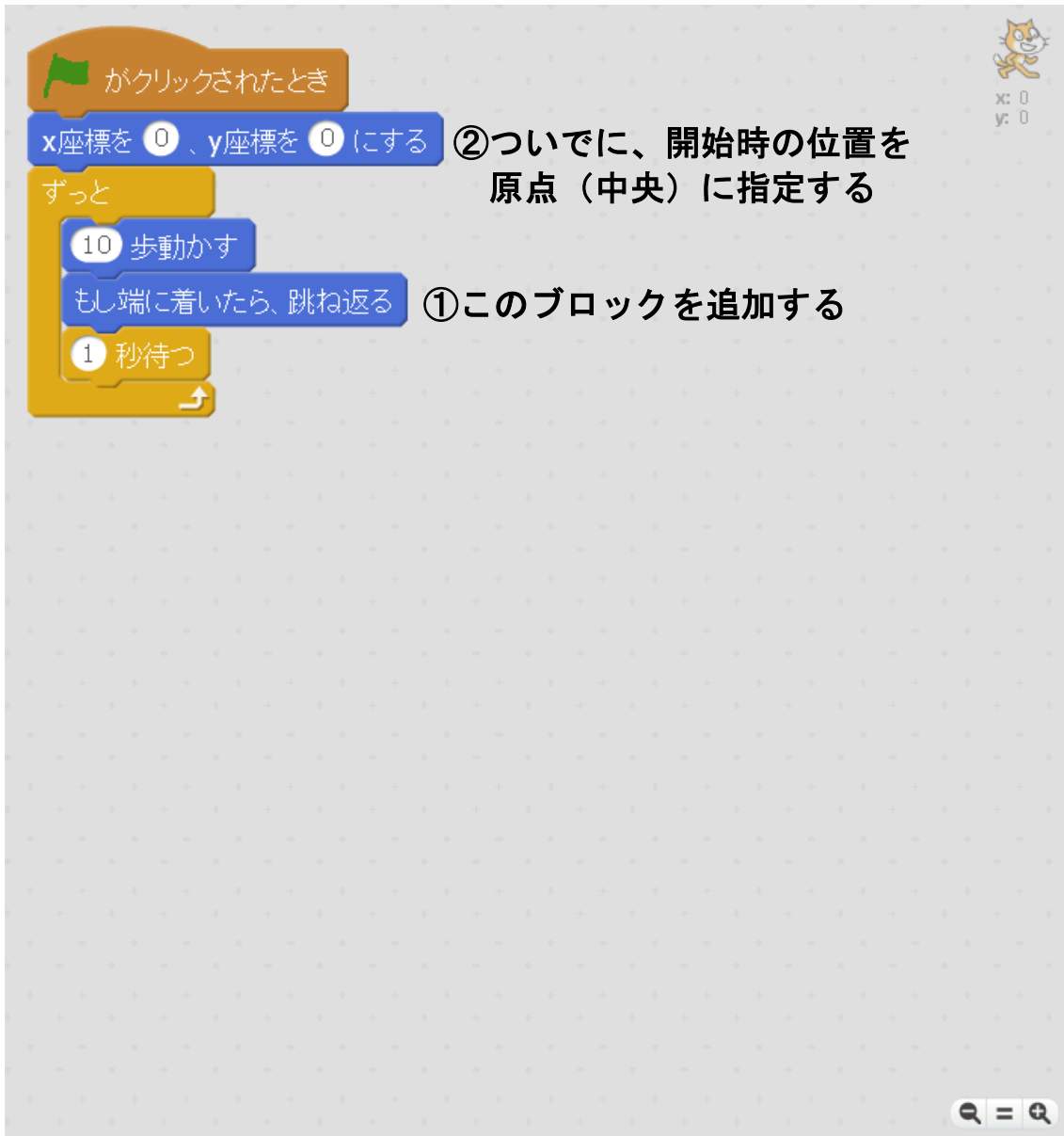
ネコ（スプライト1）には2つのコスチュームが用意されています。コスチューム1とコスチューム2を交互に表示することで歩いているように見せることができます。



昔のテレビゲームは実際にこんな感じで動きを表現していたようです。

⑤端に着いたら、戻ってこよう

☞ステージの端に着いたら、向きを変えて戻ってくるようにします。



がクリックされたとき

x座標を 0、y座標を 0 にする ②ついでに、開始時の位置を原点（中央）に指定する

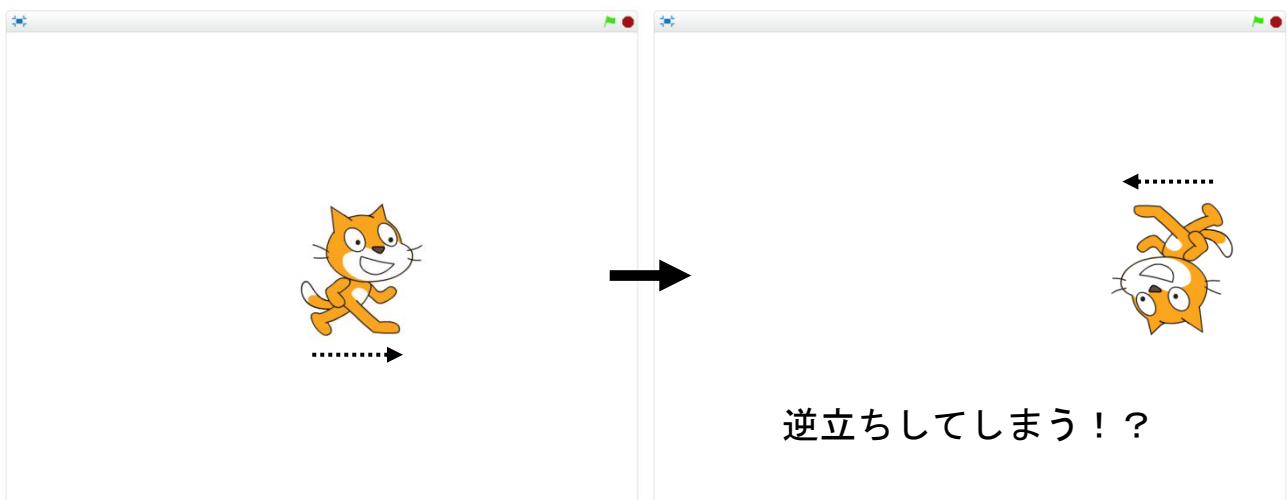
ずっと

10 歩動かす

もし端に着いたら、跳ね返る ①このブロックを追加する

1 秒待つ

実行例

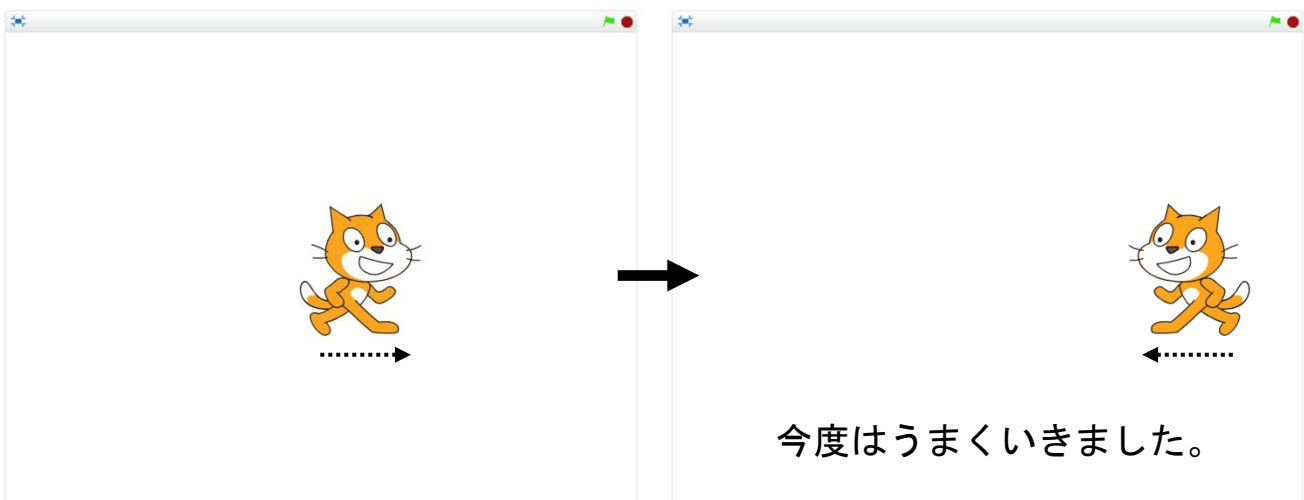


⑥逆立ちしないようにしよう

☞向きを変えたときに逆立ちしてしまわないように、開始時に設定を変更します。

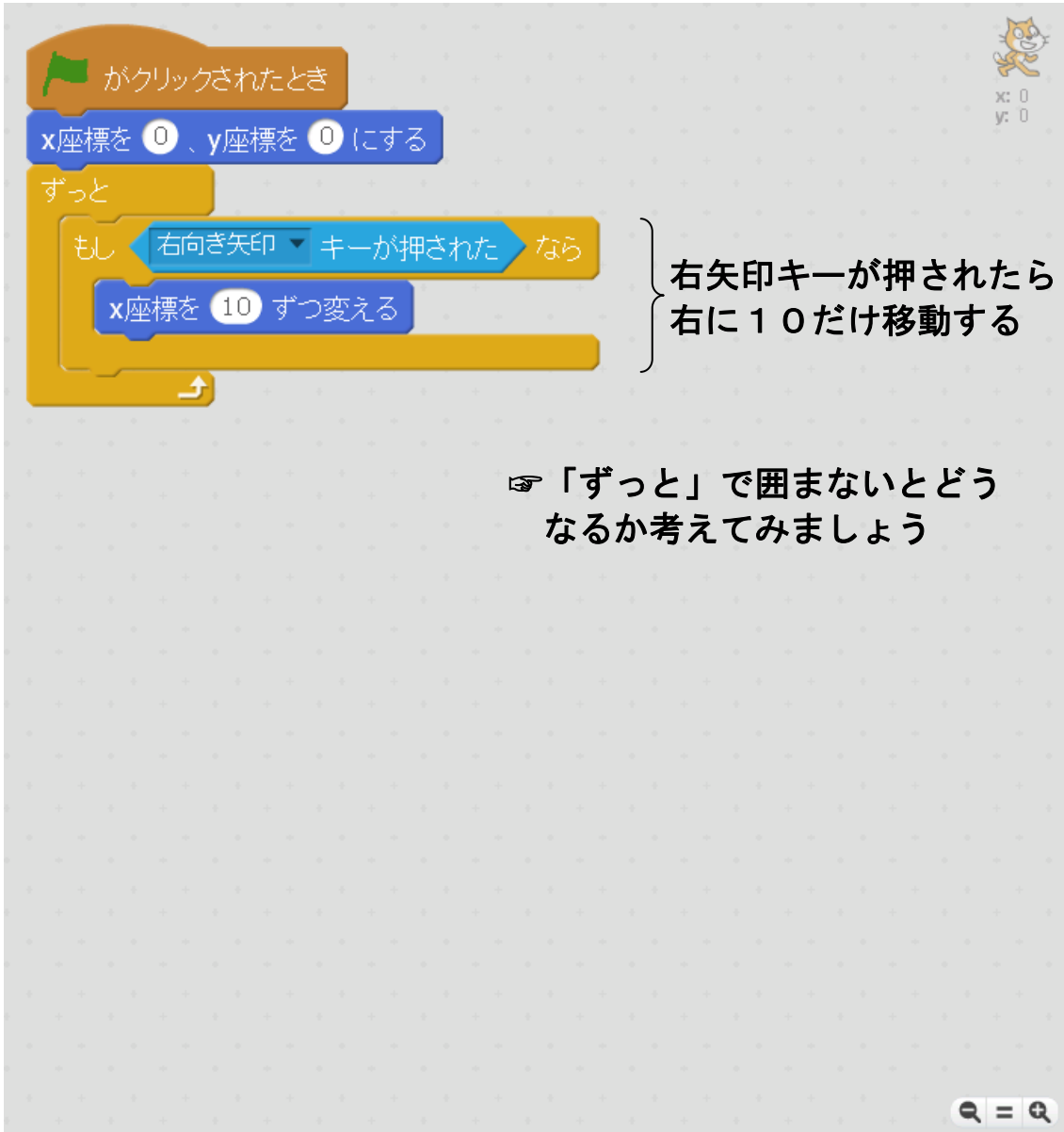


実行例



⑦キーボードで右に動かそう

☞ネコが自動で動くのではなく、キーを押したときだけ動くようにします。



The image shows a Scratch script on a stage background. The script starts with a green flag icon and the text "がクリックされたとき". This is followed by a blue block "x座標を 0、y座標を 0 にする". Below that is a yellow "ずっと" (forever) loop block. Inside the loop is a blue "もし" (if) block with the condition "右向き矢印 キーが押された なら". Inside the "もし" block is another blue block "x座標を 10 ずつ変える". To the right of the code, a bracket groups the "もし" and "x座標を 10 ずつ変える" blocks with the text "右矢印キーが押されたら 右に10だけ移動する". Below the code, there is a text prompt: "☞「ずっと」で囲まないとうなるか考えてみましょう". In the top right corner of the stage, there is a cat icon and the coordinates "x: 0" and "y: 0". In the bottom right corner, there are search and zoom icons.

プログラミング言語では、たとえば、次のように書きます：

```
x = 0.0;
y = 0.0;
while (1) {
    if (右矢印キーが押された) {
        x = x + 10.0;
    }
}
```

※代表的なプログラミング言語であるCを例にして、反復処理や条件分岐の書き方を概念的に示しています。日本語の部分は架空の文法です。

⑧キーボードで4方向に動かそう

☞ 4方向に動けるようにします。



The image shows a Scratch script for moving a character in four directions using a keyboard. The script starts with a green flag icon and a 'when clicked' event. It then sets the x and y coordinates to 0. A 'forever' loop follows, containing four 'if' blocks: 'if right arrow key is pressed, increase x by 10', 'if left arrow key is pressed, decrease x by 10', 'if up arrow key is pressed, increase y by 10', and 'if down arrow key is pressed, decrease y by 10'. A bracket on the right side of the script points to these four 'if' blocks with the text 'この部分を追加する' (Add this part).

```
Scratch Script:
- Event: Green flag clicked
- Action: Set x coordinate to 0, y coordinate to 0
- Loop: Forever
  - If: Right arrow key is pressed, then: Increase x coordinate by 10
  - If: Left arrow key is pressed, then: Decrease x coordinate by 10
  - If: Up arrow key is pressed, then: Increase y coordinate by 10
  - If: Down arrow key is pressed, then: Decrease y coordinate by 10
```


⑨ネズミを登場させよう

ネズミを登場させます。登場人物（スプライト）を追加し、ネズミのスク립トを書きます。
ネコ（スプライト1）



The image shows a Scratch script for a cat sprite. The script starts with a 'when clicked' event block. This is followed by a 'set x coordinate to 0, y coordinate to 0' block. Then, there is a 'forever' loop containing four 'if key pressed' blocks. The first 'if' block checks for the right arrow key; if pressed, it changes the x coordinate by 10. The second 'if' block checks for the left arrow key; if pressed, it changes the x coordinate by -10. The third 'if' block checks for the up arrow key; if pressed, it changes the y coordinate by 10. The fourth 'if' block checks for the down arrow key; if pressed, it changes the y coordinate by -10. The text 'ネコのスク립トに変更はありません' (No changes to the cat script) is displayed below the script. In the top right corner of the script area, there is a cat icon and the coordinates 'x: 0' and 'y: 0'. In the bottom right corner, there are search and zoom icons.

がクリックされたとき

x座標を 0、y座標を 0 にする

ずっと

もし 右向き矢印 ▼ キーが押された なら

x座標を 10 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 -10 ずつ変える

もし 上向き矢印 ▼ キーが押された なら

y座標を 10 ずつ変える

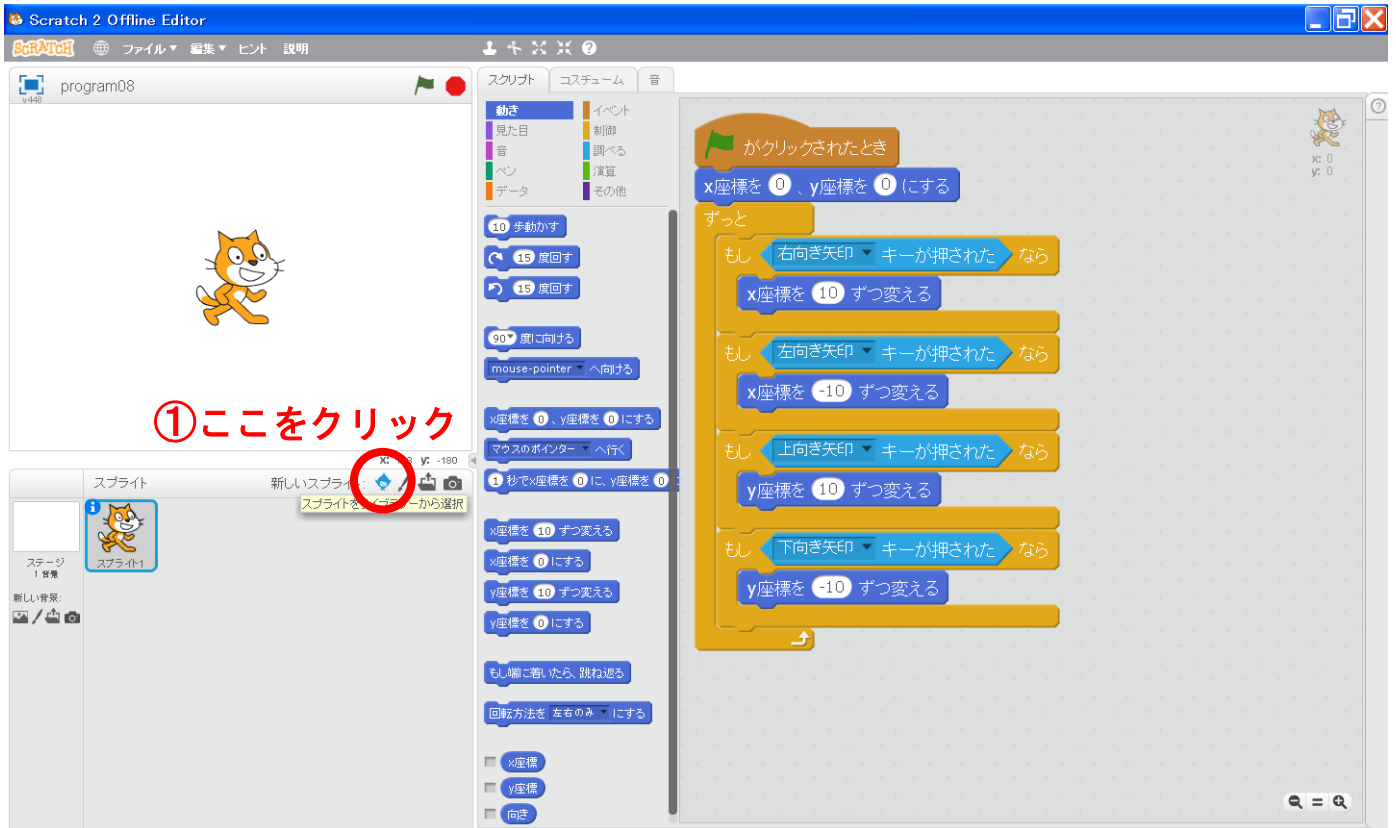
もし 下向き矢印 ▼ キーが押された なら

y座標を -10 ずつ変える

ネコのスク립トに変更はありません

x: 0
y: 0

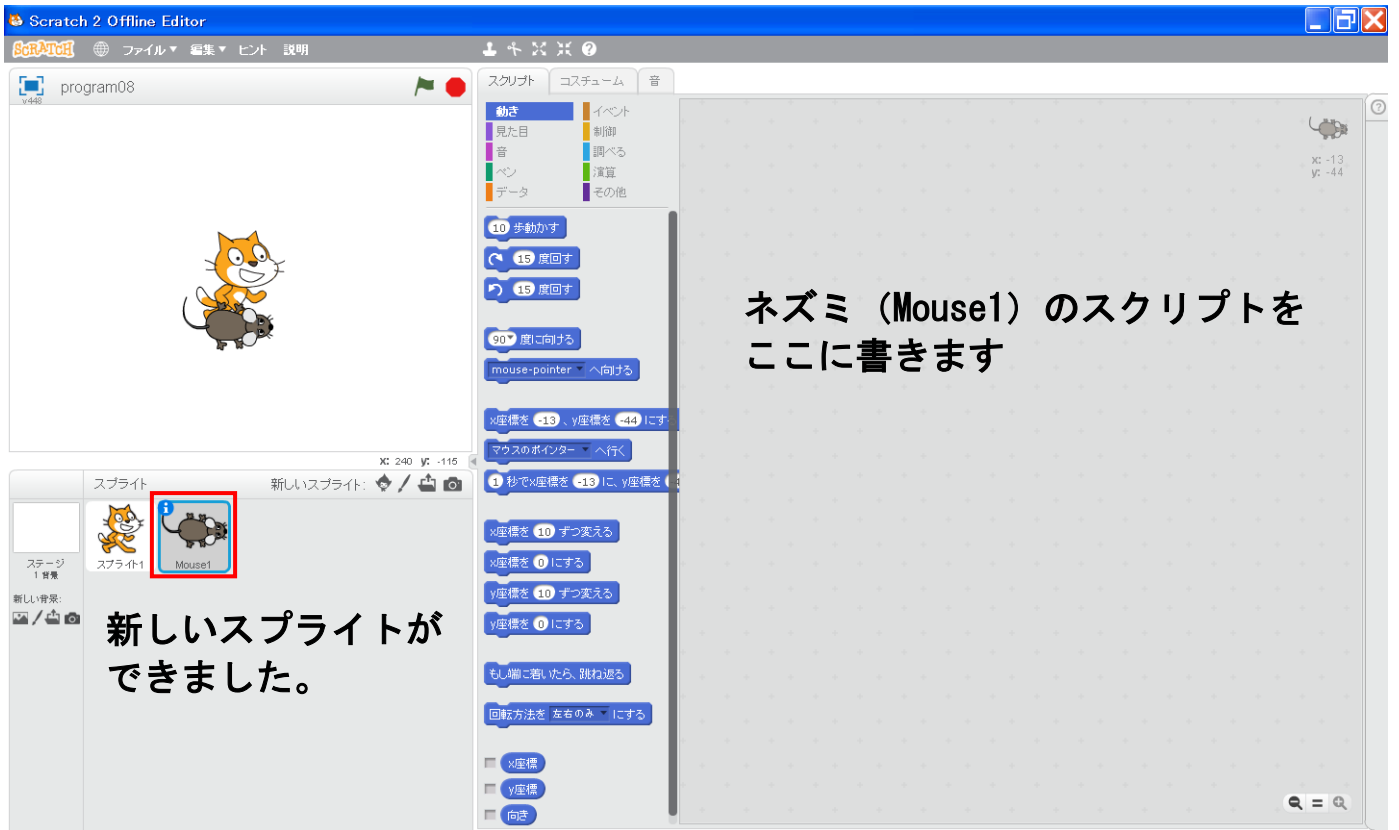
登場人物（スプライト）を追加する



①ここをクリック

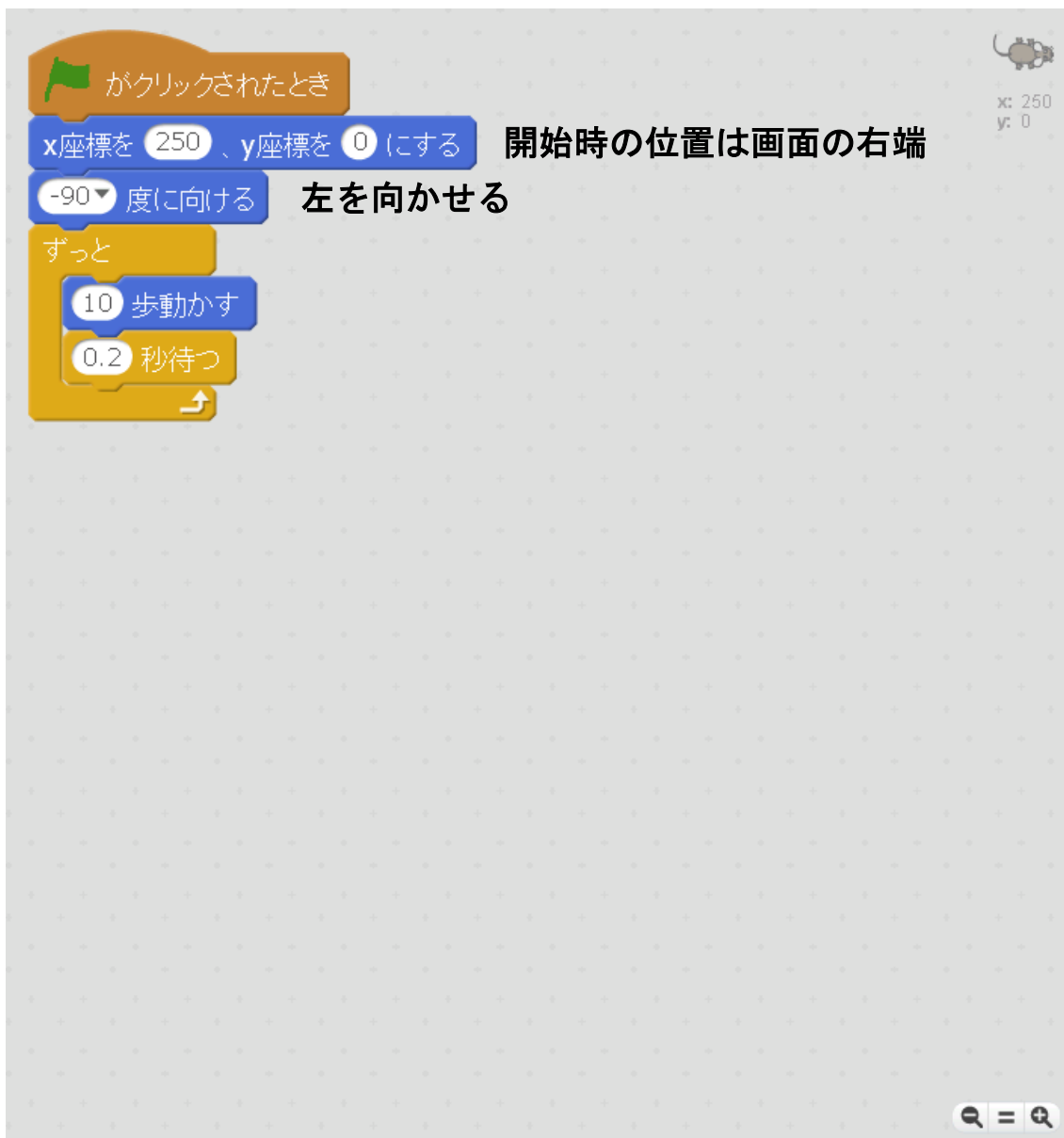


④OKをクリック



ネズミ (Mouse 1)

ネズミのスクリプトを新しく書きます。

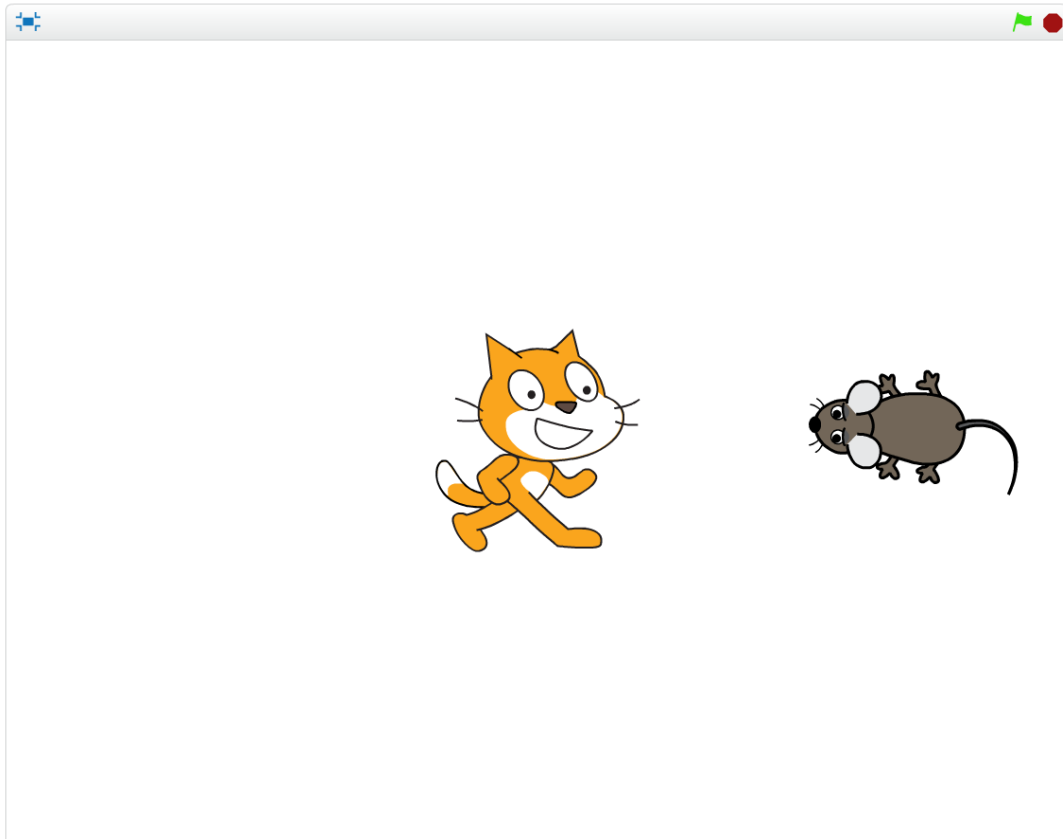


The image shows a Scratch script for a mouse character. The script is as follows:

- がクリックされたとき** (When clicked)
- x座標を 250、y座標を 0 にする** (Set x coordinate to 250, y coordinate to 0) **開始時の位置は画面の右端** (Starting position is the right edge of the screen)
- 90 度に向ける** (Turn 90 degrees) **左を向かせる** (Turn left)
- ずっと** (Forever loop)
 - 10 歩動かす** (Move 10 steps)
 - 0.2 秒待つ** (Wait 0.2 seconds)

The script is set to start at x: 250, y: 0. The mouse character is positioned at the right edge of the screen. The script is written in Japanese.

実行例



⑩ネズミを食べよう

ネコがネズミをつかまえて食べるようにします。

ネコ (スプライト1)

変更はありません。

ネズミ (Mouse1)

がクリックされたとき

表示する

x座標を 250 、y座標を 0 にする

-90 度に向ける

ずっと

10 歩動かす

0.2 秒待つ

がクリックされたとき

ずっと

もし スプライト1 までの距離 < 50 なら

隠す

すべて を止める

この部分を追加する

もしネコまでの距離が50未満なら、
隠す=ネズミを消す
すべてを止める=ゲーム終了

⑪ネズミを食べたら、ニャーと鳴こう

👉ゲームを楽しくするために、音も出るようにします。

「ニャー」と鳴くのは、ネズミではなくネコです。そこで次のような手順でネコを鳴かせます。

1. ネズミの側で「猫に食べられた」と判定したら、
2. ネズミからネコに「猫に食べられた」というメッセージを送る。
3. ネコの側で「猫に食べられた」というメッセージを受け取ったら、
4. ネコが「ニャー」と鳴く。

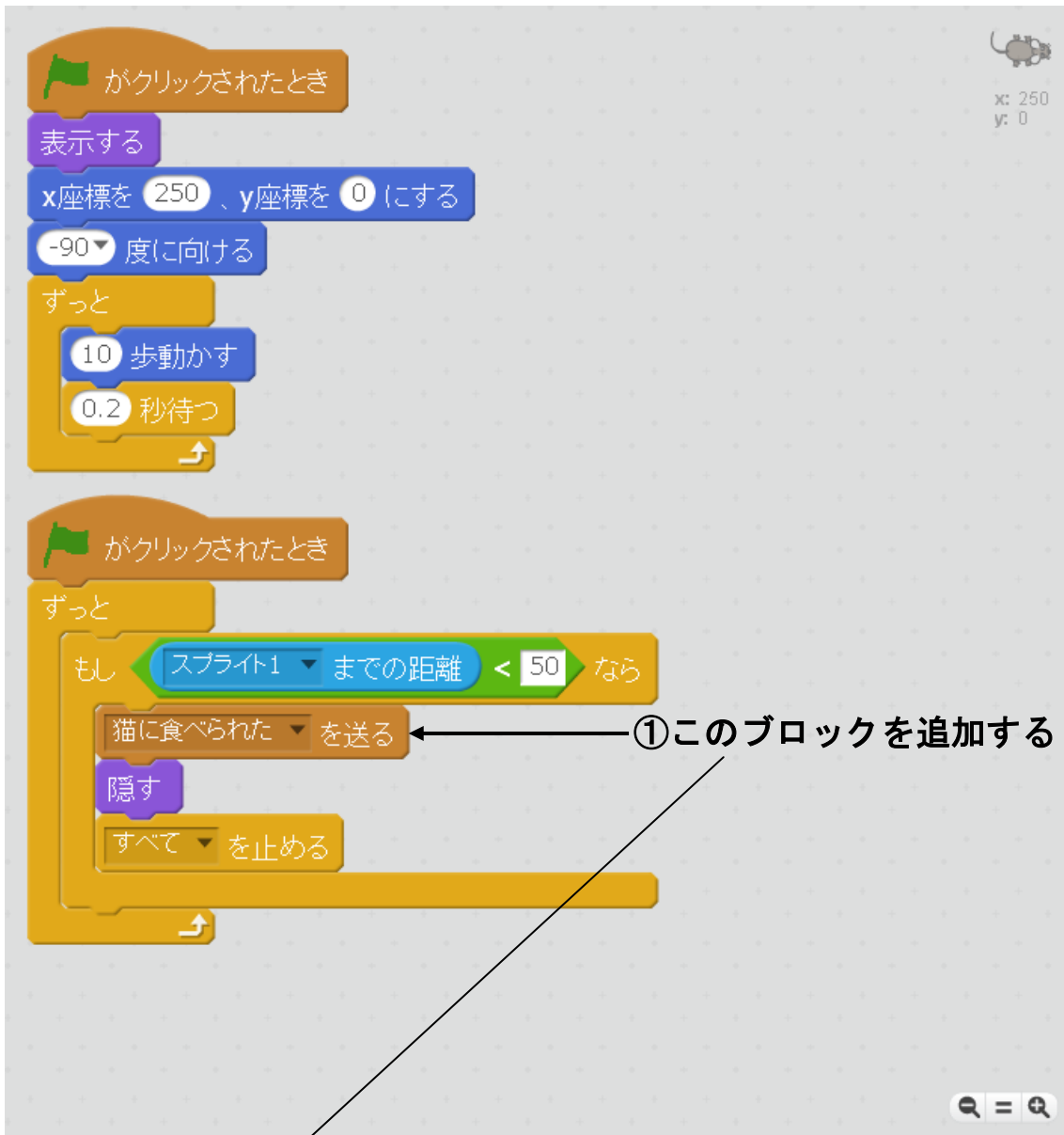
ネコ（スプライト1）



The image shows a Scratch script for a cat sprite. The script is divided into two main sections. The first section, triggered by a green flag click, sets the x and y coordinates to 0 and enters a 'forever' loop. Inside the loop, there are four 'if' blocks: 'if right arrow key pressed, then x coordinate +10', 'if left arrow key pressed, then x coordinate -10', 'if up arrow key pressed, then y coordinate +10', and 'if down arrow key pressed, then y coordinate -10'. The second section is triggered when the message 'cat was eaten' is received, and it plays the 'meow' sound. A bracket on the right side of the second section points to it with the text '②この部分を追加する' (Add this part).

```
Scratch Script for Cat (Sprite 1):  
- Event: Green flag clicked  
  - Action: Set x coordinate to 0, y coordinate to 0  
  - Loop: Forever  
    - If: Right arrow key pressed, then x coordinate +10  
    - If: Left arrow key pressed, then x coordinate -10  
    - If: Up arrow key pressed, then y coordinate +10  
    - If: Down arrow key pressed, then y coordinate -10  
- Message Received: Cat was eaten  
  - Action: Play sound: meow
```

ネズミ (Mouse 1)



※メッセージを定義する方法



⑫ネズミをランダムな位置から登場させよう

いつも同じ場所ではなく、ランダムな場所からネズミが出てくるようにします。
ネコ (スプライト1)

変更はありません。

ネズミ (Mouse 1)

この部分に**乱数**のブロックを入れる
乱数=適当な数がランダムに決まる

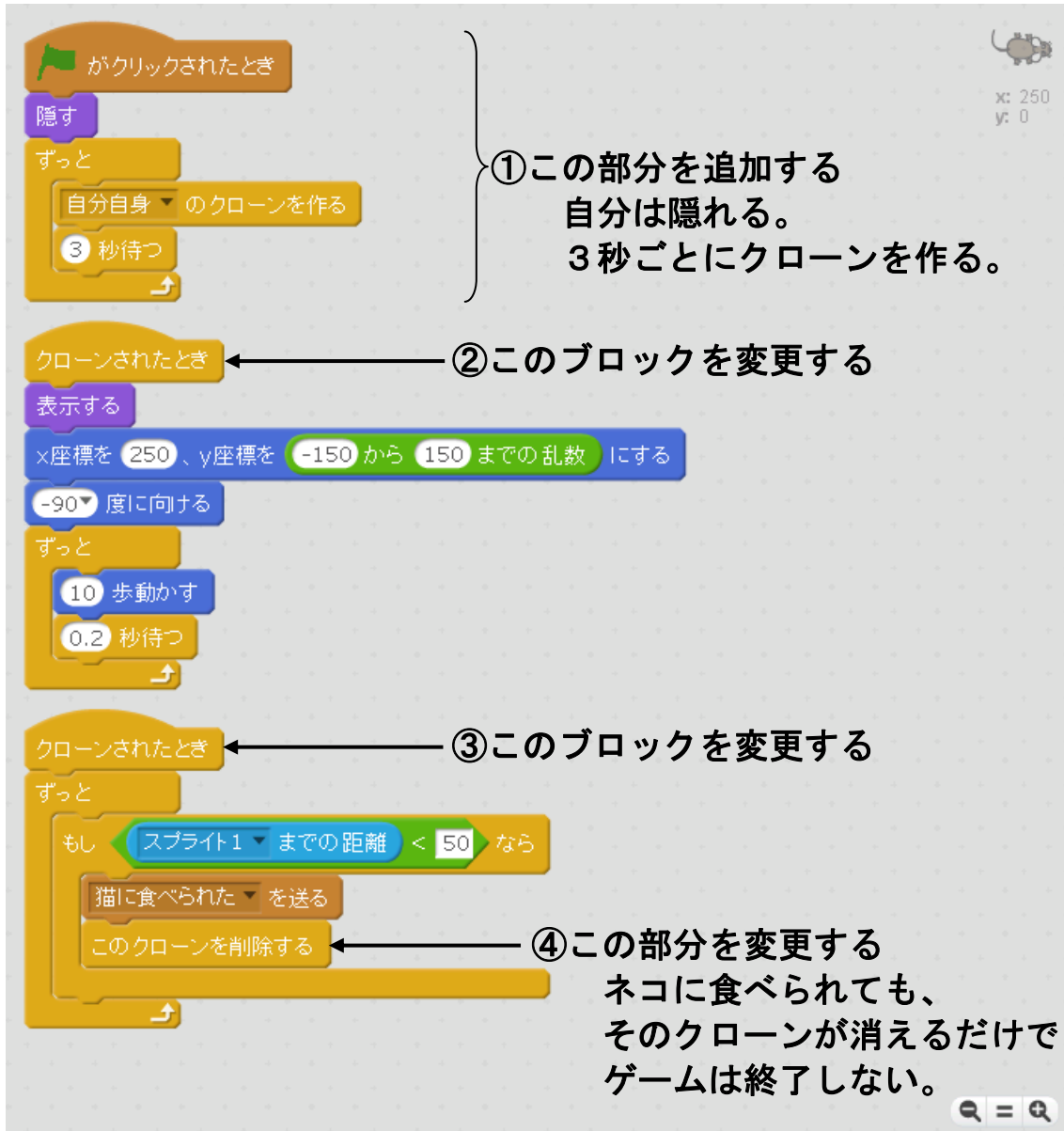
⑬ネズミをたくさん登場させよう

ネズミが1匹だけでは物足りないので、たくさん出てくるようにします。

ネコ (スプライト1)

変更はありません。

ネズミ (Mouse 1)



The image shows a Scratch script for a mouse character. The script is divided into three main sections, each starting with a 'when clicked' event. The first section includes 'hide', 'repeat forever' with 'create clone of self', and 'wait 3 seconds'. The second section starts with 'when cloned', followed by 'show', 'set x and y coordinates to random values between -150 and 150', 'set direction to -90 degrees', and a 'repeat forever' loop containing 'move 10 steps' and 'wait 0.2 seconds'. The third section starts with 'when cloned', followed by a 'while' loop: 'while distance to sprite 1 is less than 50', containing 'send message to cat when eaten' and 'delete this clone'. Annotations with arrows point to these sections, explaining the changes to be made.

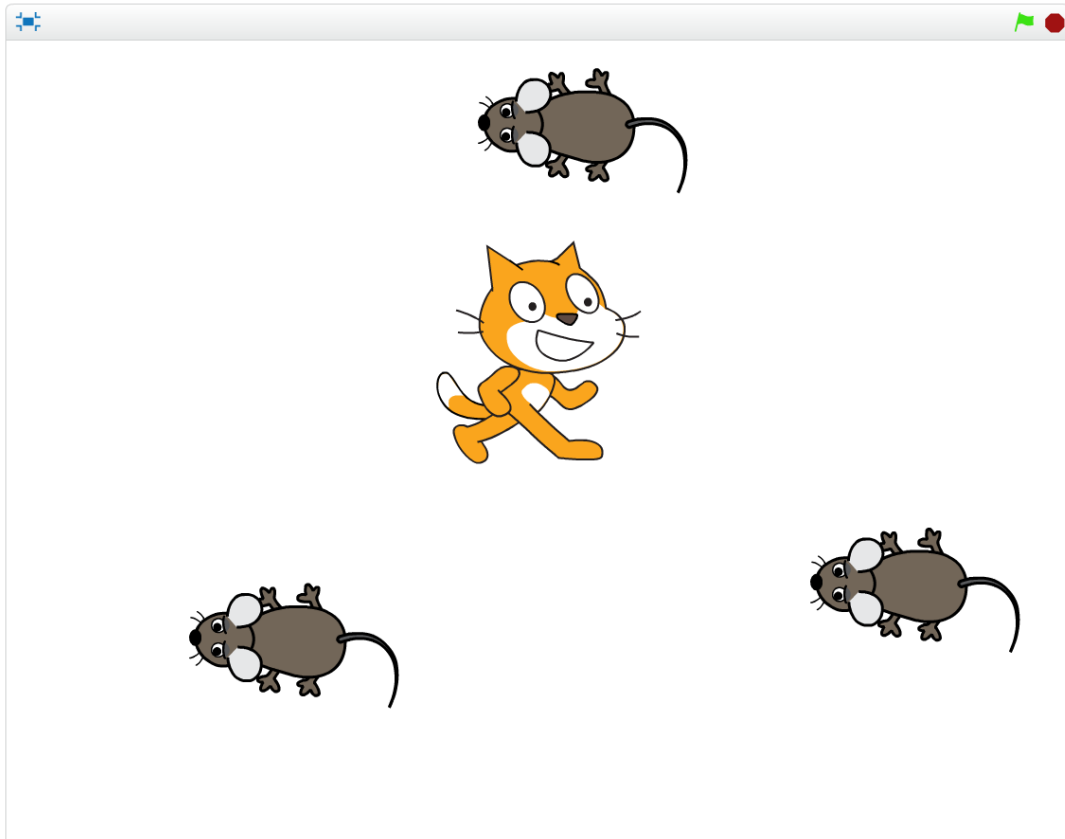
①この部分を追加する
自分は隠れる。
3秒ごとにクローンを作る。

②このブロックを変更する

③このブロックを変更する

④この部分を変更する
ネコに食べられても、
そのクローンが消えるだけで
ゲームは終了しない。

実行例



⑭ゲームの進行を速くしよう

☞設定を変えて、スリルのあるゲームにします。

ネコ (スプライト1)

変更はありません。

ネズミ (Mouse 1)

The image shows the Scratch code editor for the mouse character. The code is organized into three event-driven sections:

- がクリックされたとき (When clicked):** Includes a '隠す' (Hide) block, followed by a 'ずっと' (Forever) loop containing '自分自身 のクローンを作る' (Create a clone of myself) and a '0.5 秒待つ' (Wait 0.5 seconds) block. A red arrow points to the '0.5' value.
- クローンされたとき (When cloned):** Includes a '表示する' (Show) block, followed by 'x座標を 250、y座標を -150 から 150 までの乱数 にする' (Set x coordinate to 250, y coordinate to a random number between -150 and 150), and '-90 度に向ける' (Turn 90 degrees counter-clockwise). This is followed by another 'ずっと' (Forever) loop containing '10 歩動かす' (Move 10 steps) and a '0.05 秒待つ' (Wait 0.05 seconds) block. A red arrow points to the '0.05' value.
- クローンされたとき (When cloned again):** Includes a 'ずっと' (Forever) loop with a 'もし スプライト1 までの距離 < 50 なら' (If distance to sprite 1 is less than 50) condition. Inside the loop are '猫に食べられた を送る' (Send 'eaten by cat') and 'このクローンを削除する' (Delete this clone) blocks.

At the top right, the mouse's current position is shown as x: 250, y: 0. At the bottom right, there are search, zoom, and refresh icons.

⑮食べたネズミの数を数えよう

☞変数を定義して、ネコに食べられたネズミの数を数えます。

変数を定義する

Scratch 2 Offline Editor

変数を定義する

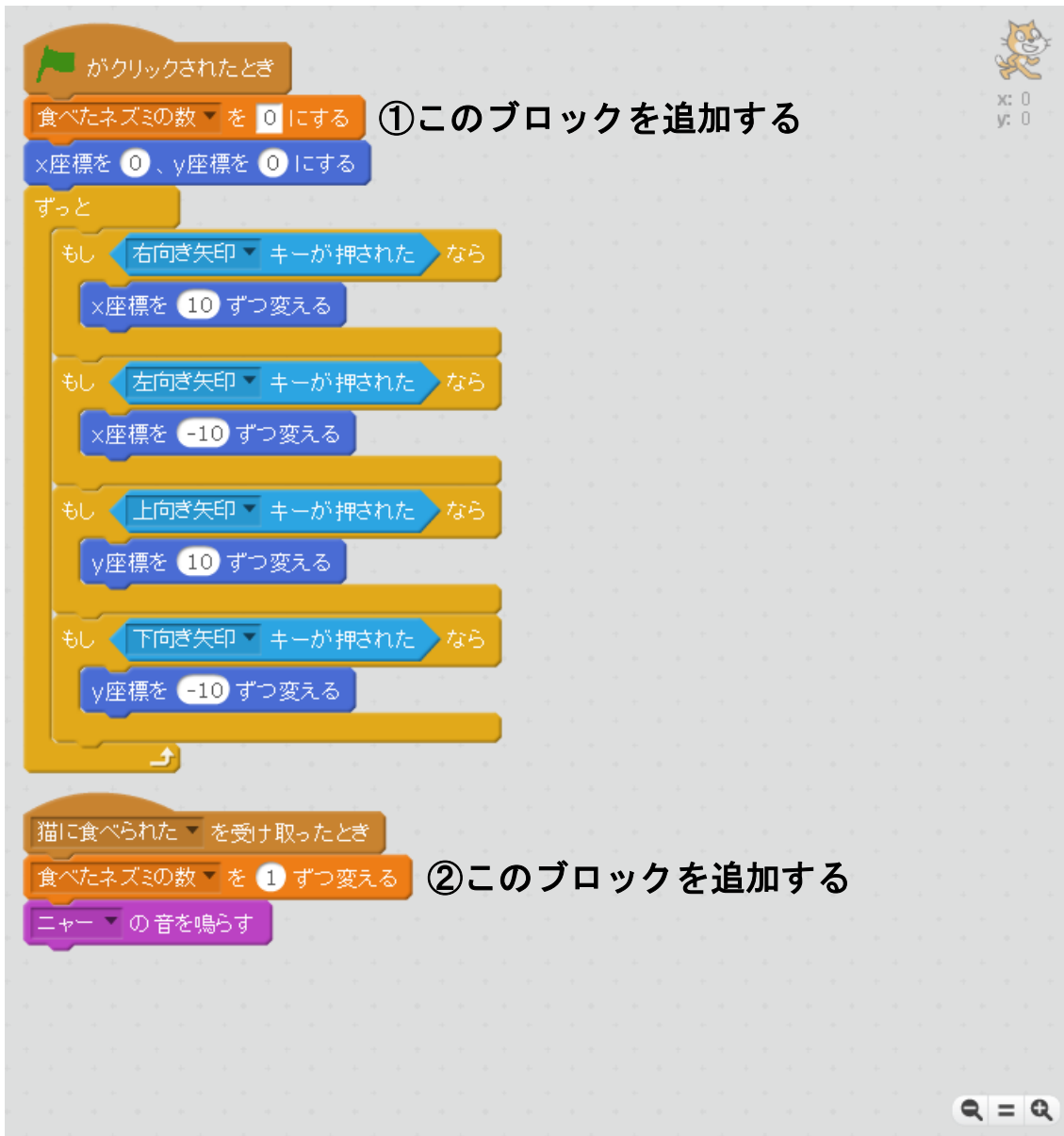
- ① データをクリック
- ② 「変数を作る」をクリック
- ③ 変数名を入力
- ④ OKをクリック

Scratch 2 Offline Editor

変数が表示されます。

変数が作られました。

ネコ (スプライト1)



がクリックされたとき

食べたネズミの数 を 0 にする ①このブロックを追加する

x座標を 0、y座標を 0 にする

ずっと

もし 右向き矢印 キーが押された なら

x座標を 10 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 -10 ずつ変える

もし 上向き矢印 キーが押された なら

y座標を 10 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 -10 ずつ変える

猫に食べられた を受け取ったとき

食べたネズミの数 を 1 ずつ変える ②このブロックを追加する

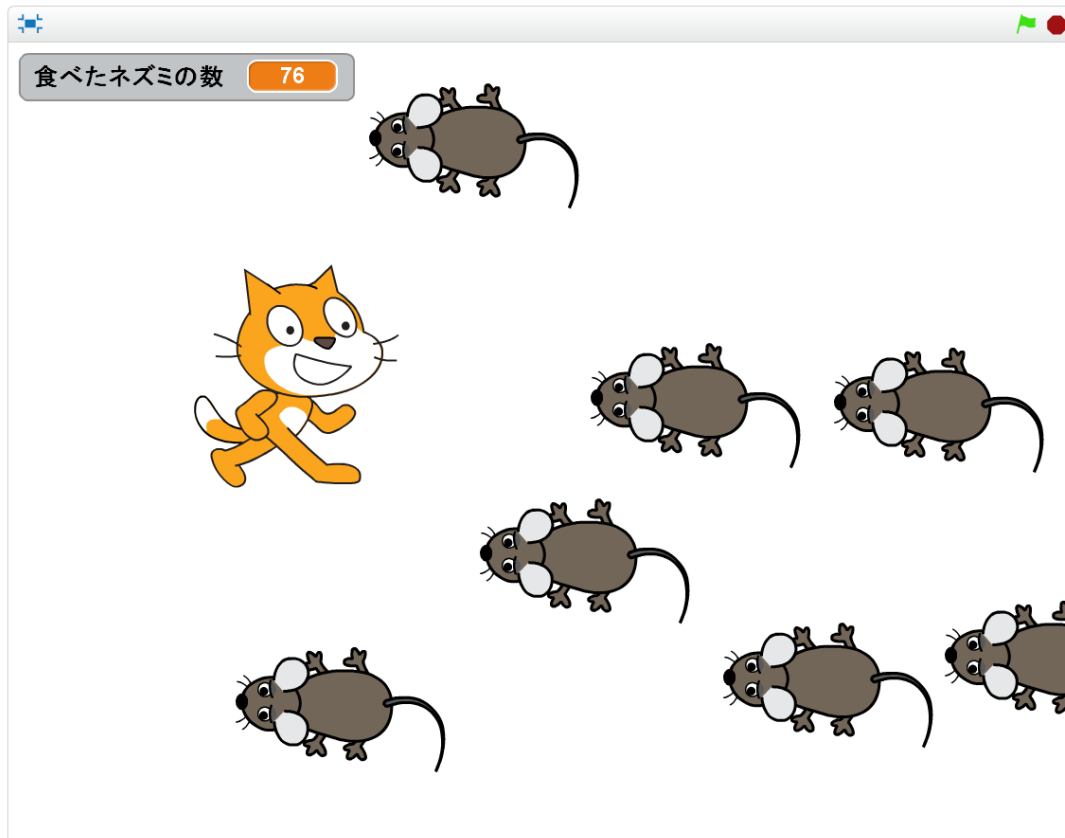
ニャー の音を鳴らす

x: 0
y: 0

ネズミ (Mouse1)

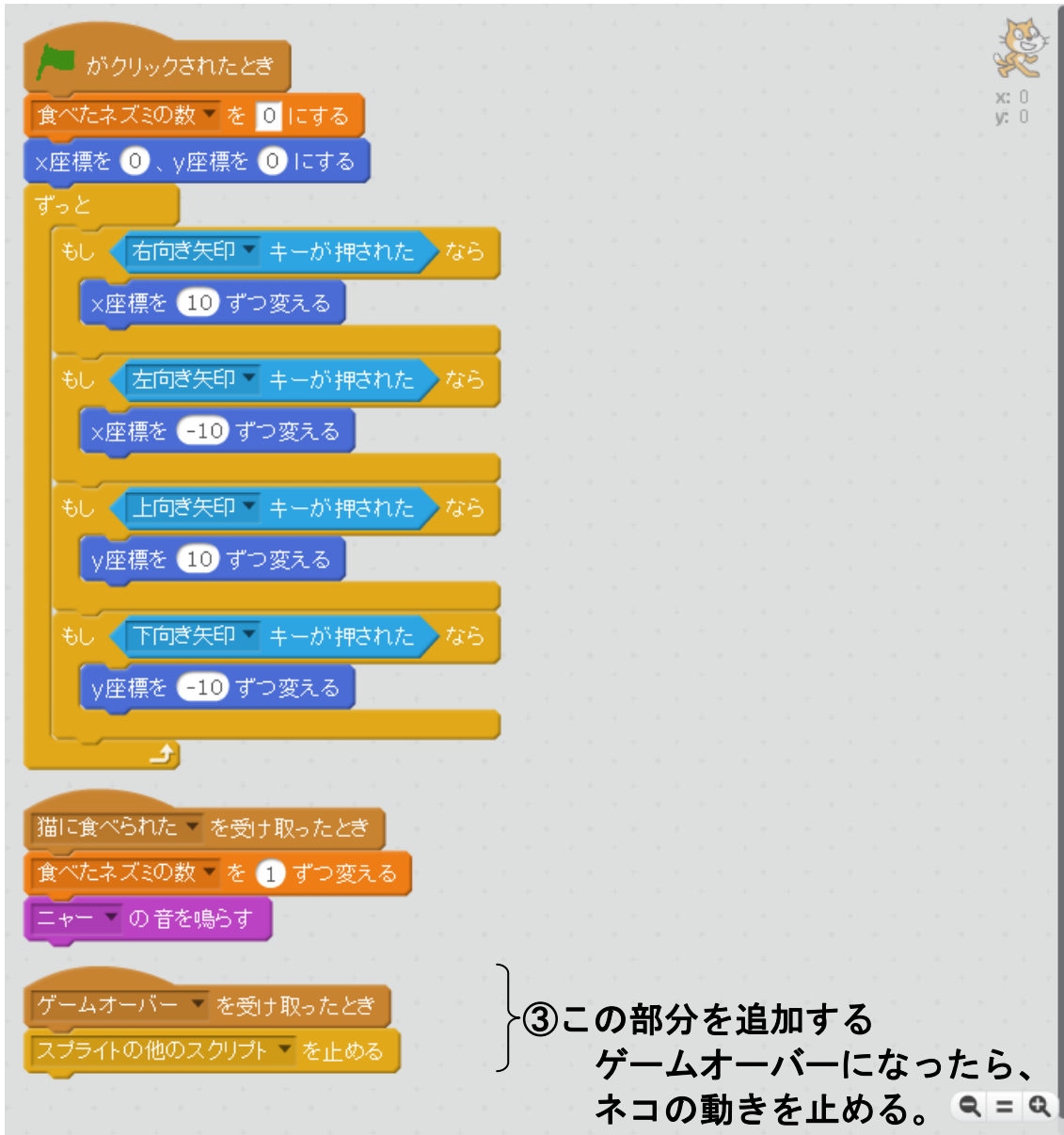
変更はありません。

実行例



⑩ネズミを取り逃がしたらゲームオーバーにしよう

いよいよ完成です。ネズミが画面の左端に来たら、ゲームを止めるようにします。
ネコ（スプライト1）



The image shows a Scratch script for a cat sprite. The script is organized into three main sections:

- When clicked:** Resets the 'Number of mice eaten' to 0 and the x and y coordinates to 0.
- Forever loop:** Checks for arrow key presses and moves the cat accordingly:
 - Right arrow: x coordinate +10
 - Left arrow: x coordinate -10
 - Up arrow: y coordinate +10
 - Down arrow: y coordinate -10
- When eaten by cat:** Increments the 'Number of mice eaten' by 1 and plays a meow sound.
- When game over:** Stops all other scripts for the sprite.

③この部分を追加する
ゲームオーバーになったら、
ネコの動きを止める。

ネズミ (Mouse 1)

0.5 秒待つ

クローンされたとき

表示する

x座標を 250、y座標を -150 から 150 までの乱数 にする

-90 度に向ける

ずっと

10 歩動かす

0.05 秒待つ

クローンされたとき

ずっと

もし スプライト1 までの距離 < 50 なら

猫に食べられた を送る

このクローンを削除する

もし x座標 < -250 なら

ゲームオーバー を送る

ゲームオーバー を受け取ったとき

スプライトの他のスクリプト を止める

①この部分を追加する
左端まで来たら、
ゲームオーバーを宣言する。

②この部分を追加する
ゲームオーバーになったら、
ネズミの動きを止める。

1匹でも取り逃がすとゲームオーバーです。簡単なゲームですが、なかなかスリルがありますね。自分で手を加えてさらにおもしろくしてもよいし、新しいゲームを考えてもよいでしょう。

参考までに、この講座で作ってもらったものとは少し違ったゲームの作成例も配布しておきます。ゲーム作りやプログラミングを楽しんでください。