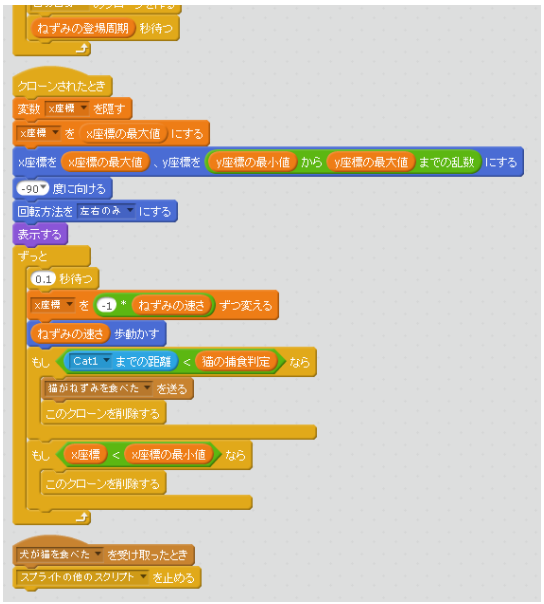
国立大学法人 東京学芸大学 理科教員高度支援センター

ASCeST

２０１９年度 理数系教員指導力向上研修

先生のためのＳｃｒａｔｃｈ入門講座

―プログラミング教育の必修化に向けて―



（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、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てプログラムを作成す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

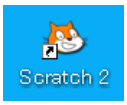
まず、
　https://scratch.mit.edu/scratch2download/
から Scratch 2 をダウンロードします。次に、
　https://get.adobe.com/jp/air/
から Adobe AIR をダウンロードします（上記の Scratch のダウンロードサイトからリンクあり）。

※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 

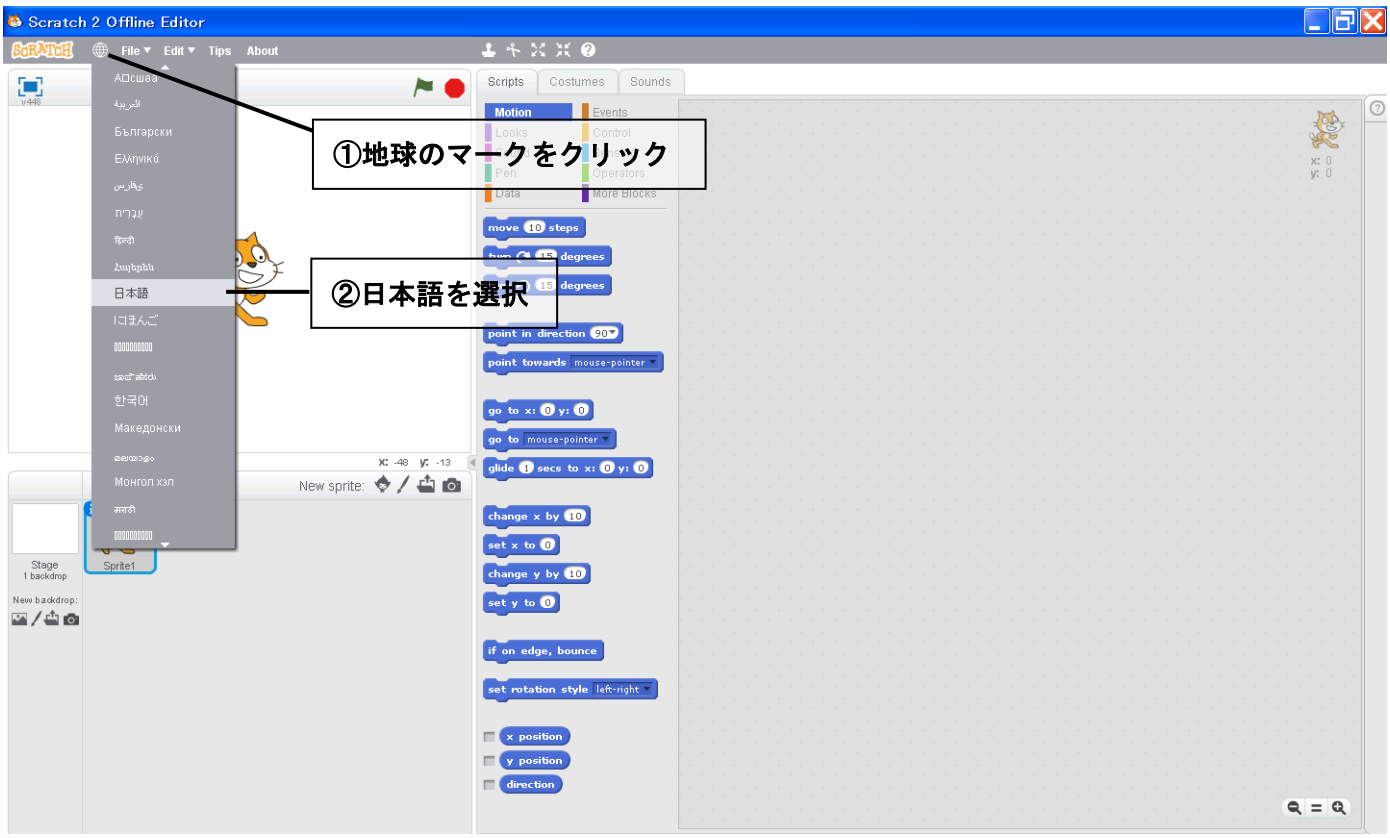
をダブルクリックします。

※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。

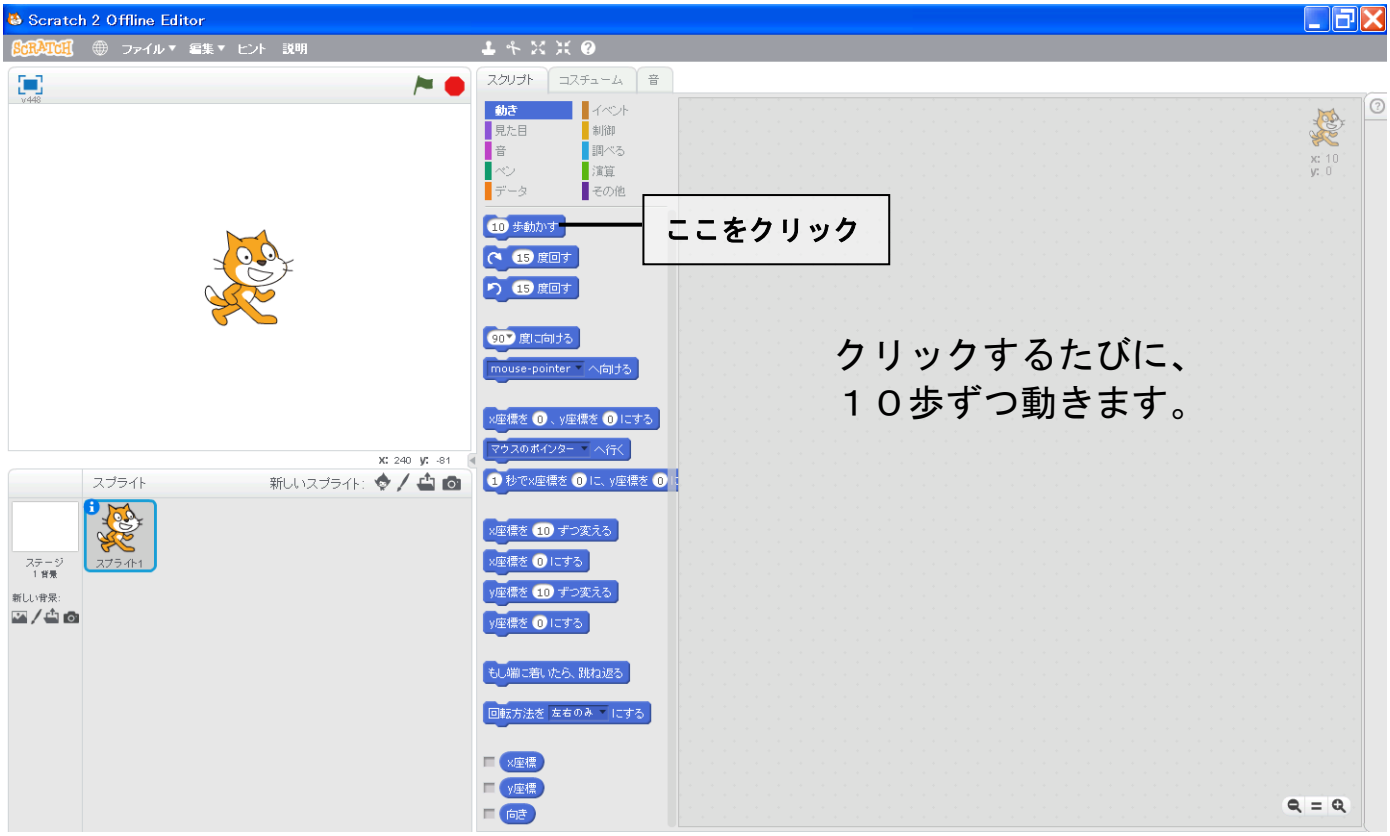


※低学年の児童の場合、「にほんご」を選ぶと、簡単な日本語表現にすることができます。

第２部：プログラミング

まずは、命令を試してみよう

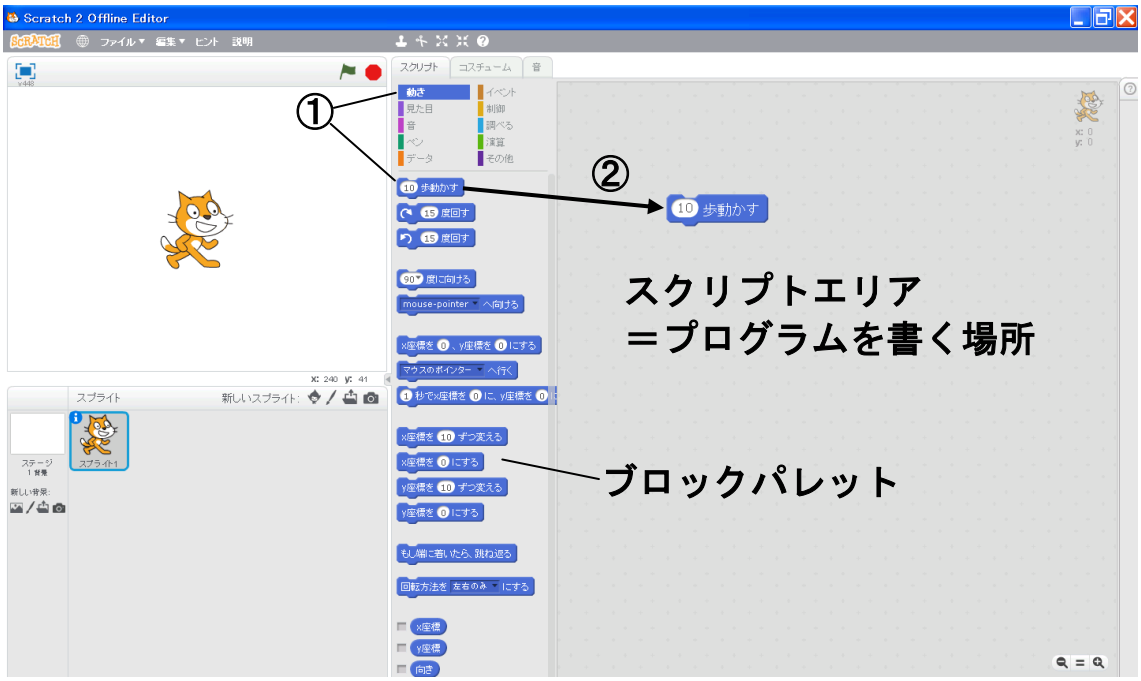
命令の内容：ネコを１０歩動かす



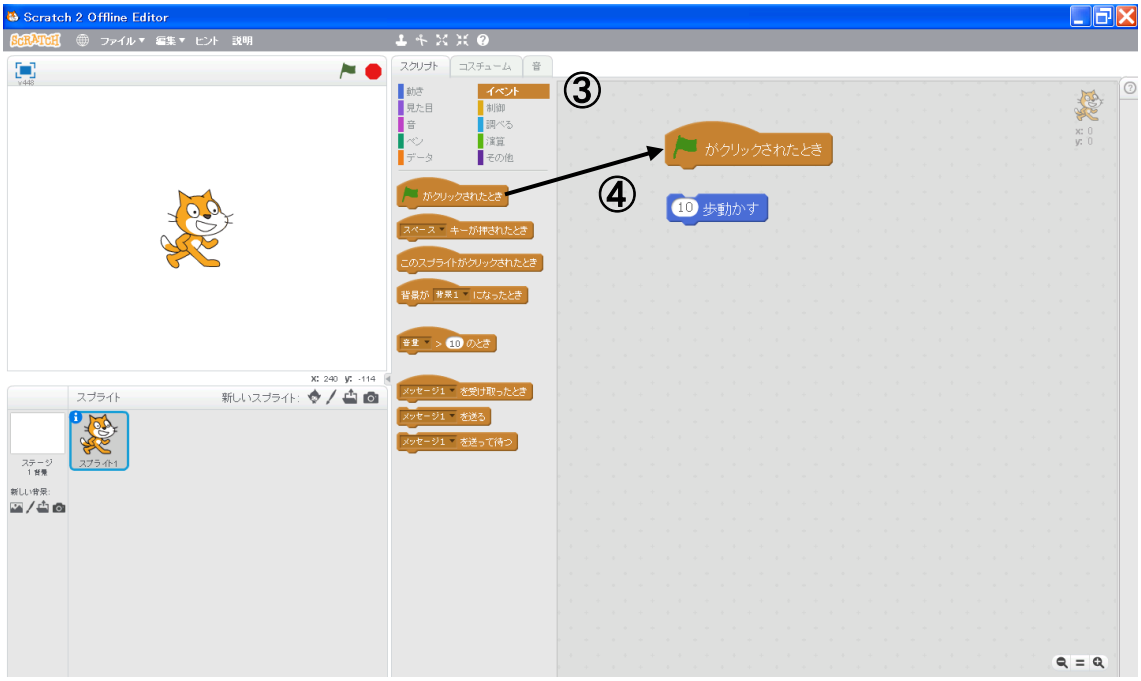
次に、簡単なプログラムを書いてみよう

プログラムの内容：ネコを１０歩動かす

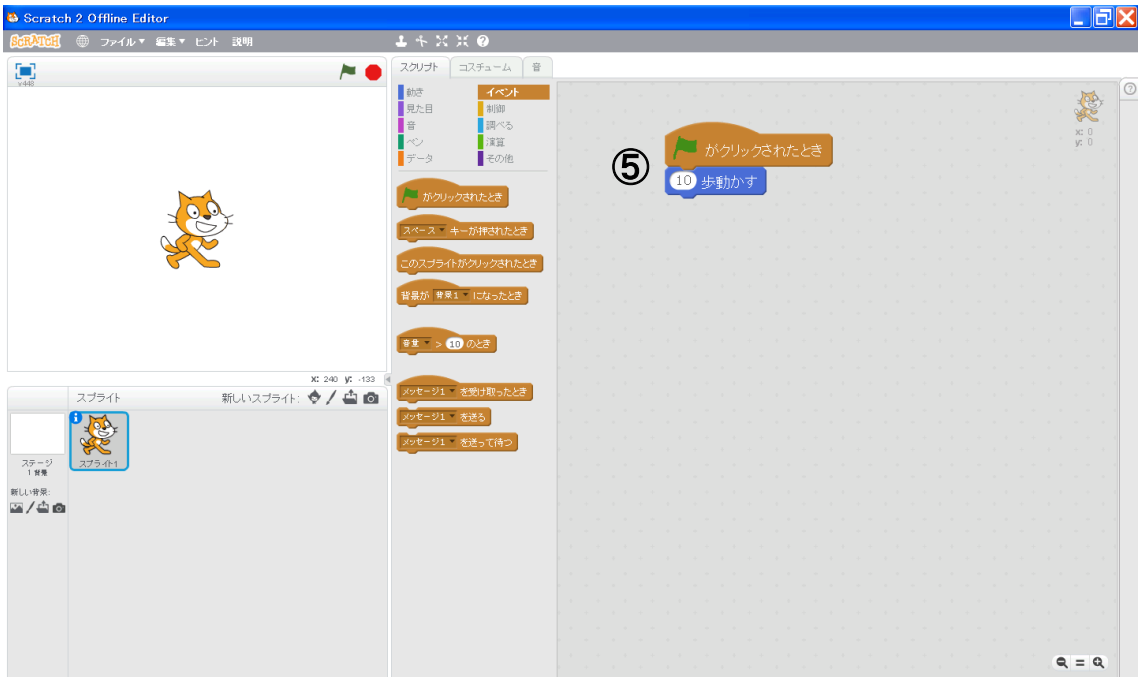
①中央のブロックパレットの中の**動き**から**１０歩動かす**を選んで、
②右側のスクリプトエリアにドラッグする（クリックをしたまま引きずる）。



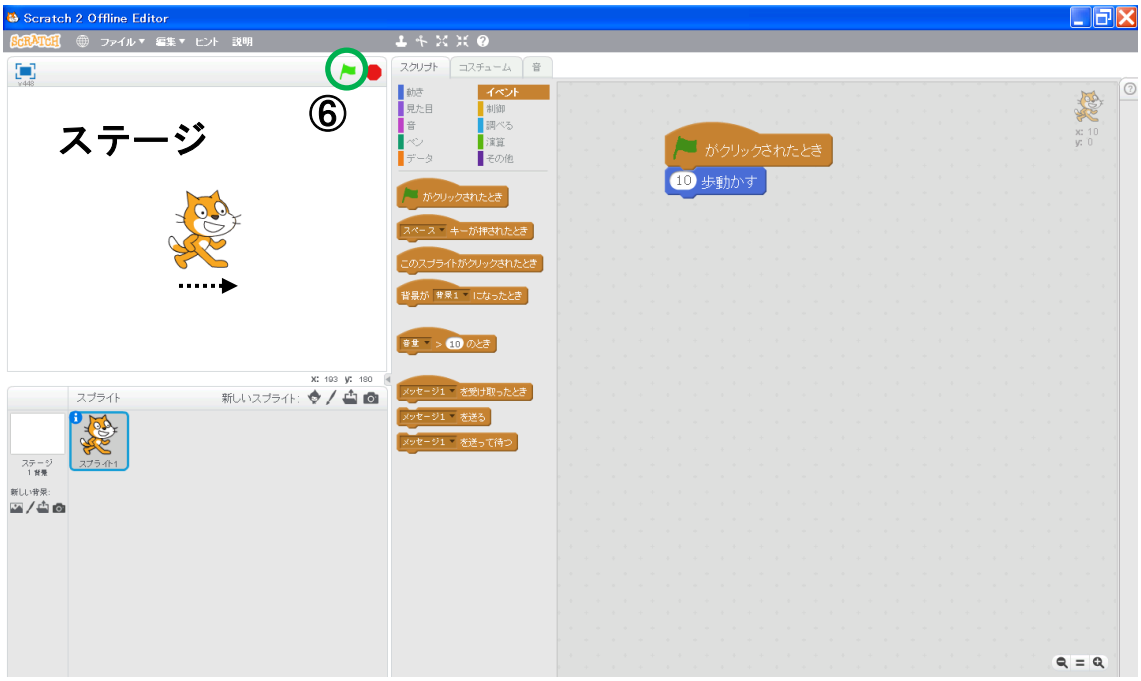
③中央のブロックパレットの中の**イベント**から**緑の旗がクリックされたとき**を選んで、
④右側のスクリプトエリアにドラッグする。



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。



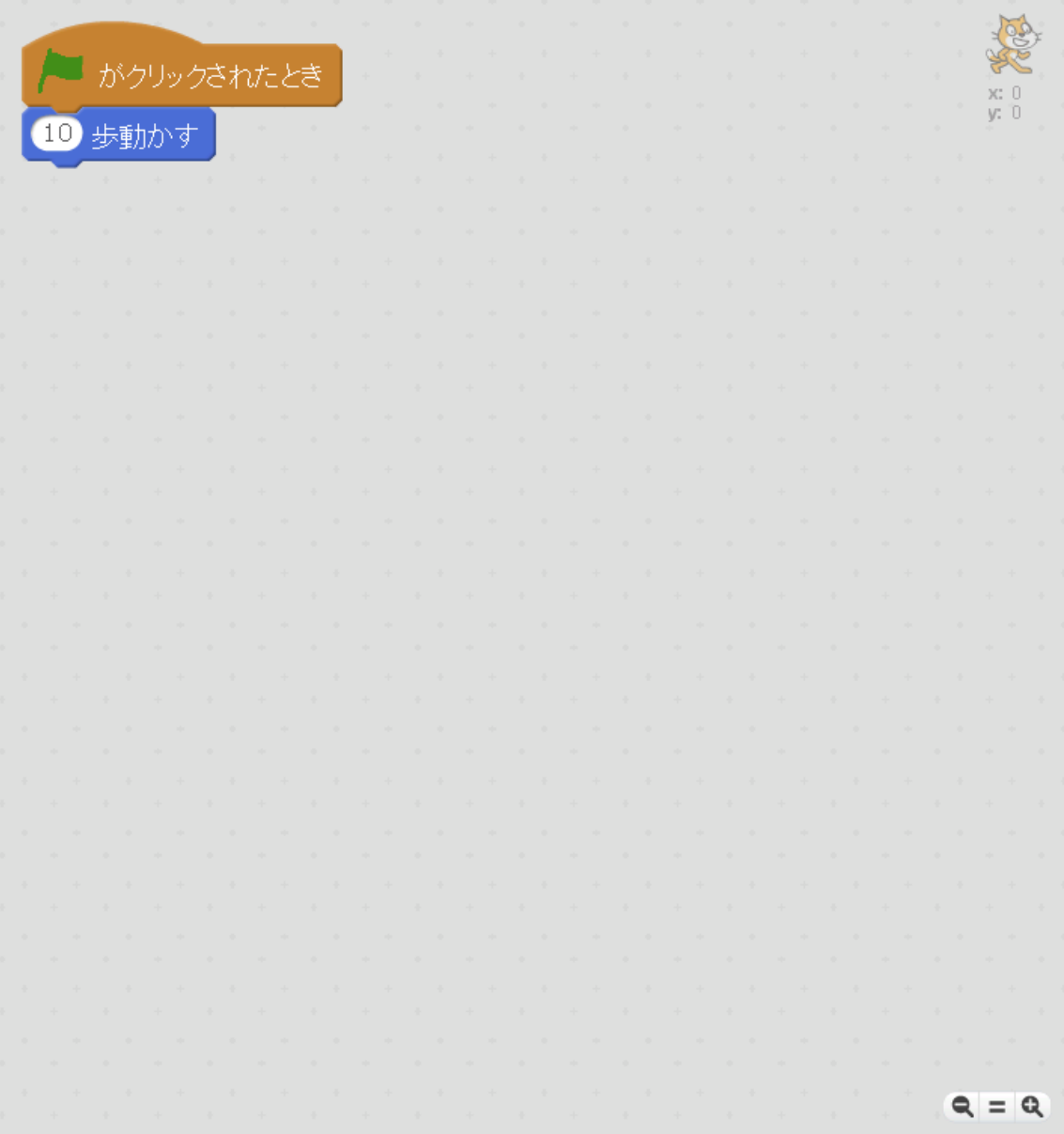
⑥左側のステージで緑の旗をクリックする。



※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。

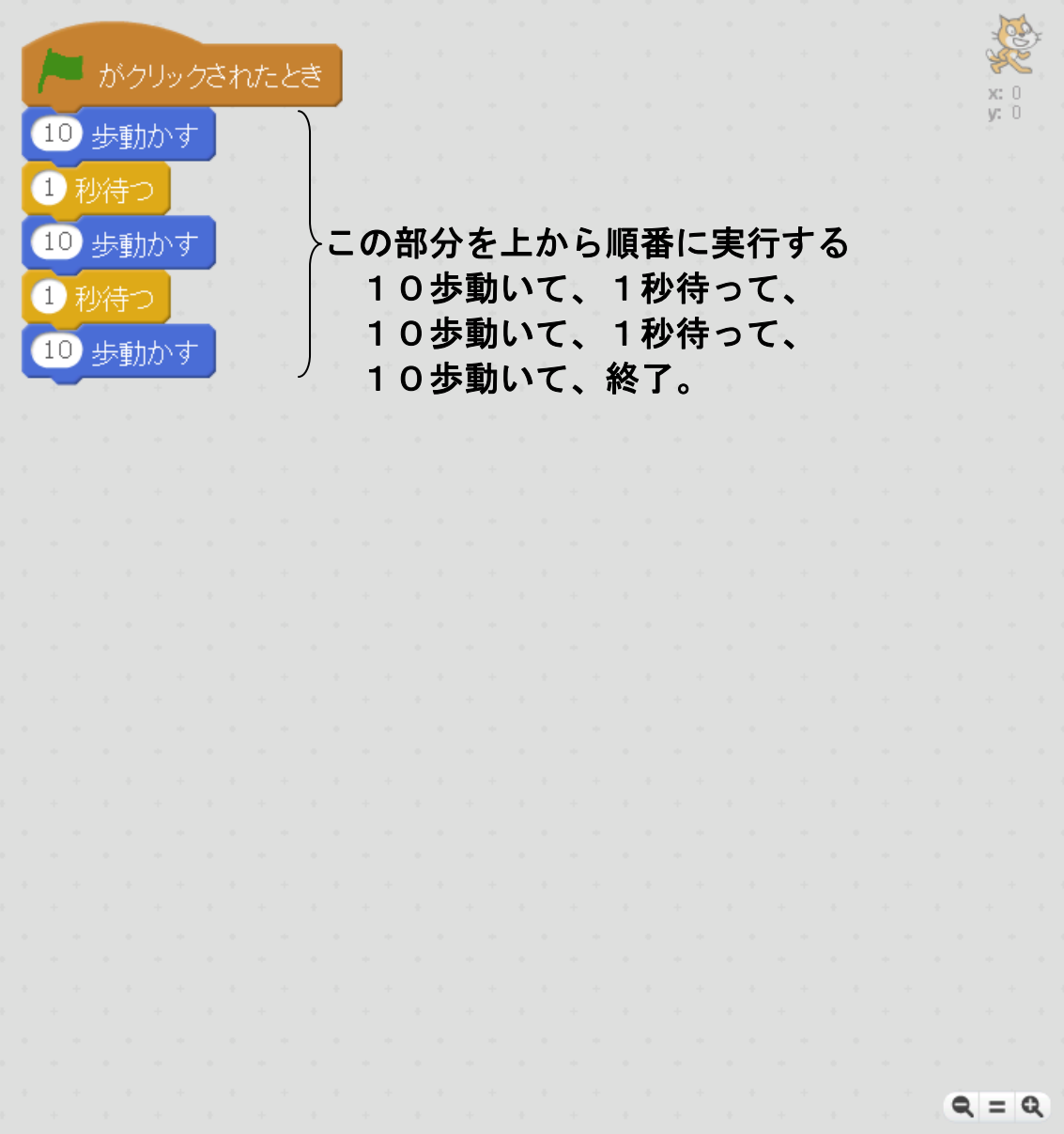


☞書き終わったら、ステージで**緑の旗**をクリックして実行してみよう。

☞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。



☞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。

不要になったブロックは、切り離した後で、右クリックして「削除」を選べば消去できます。

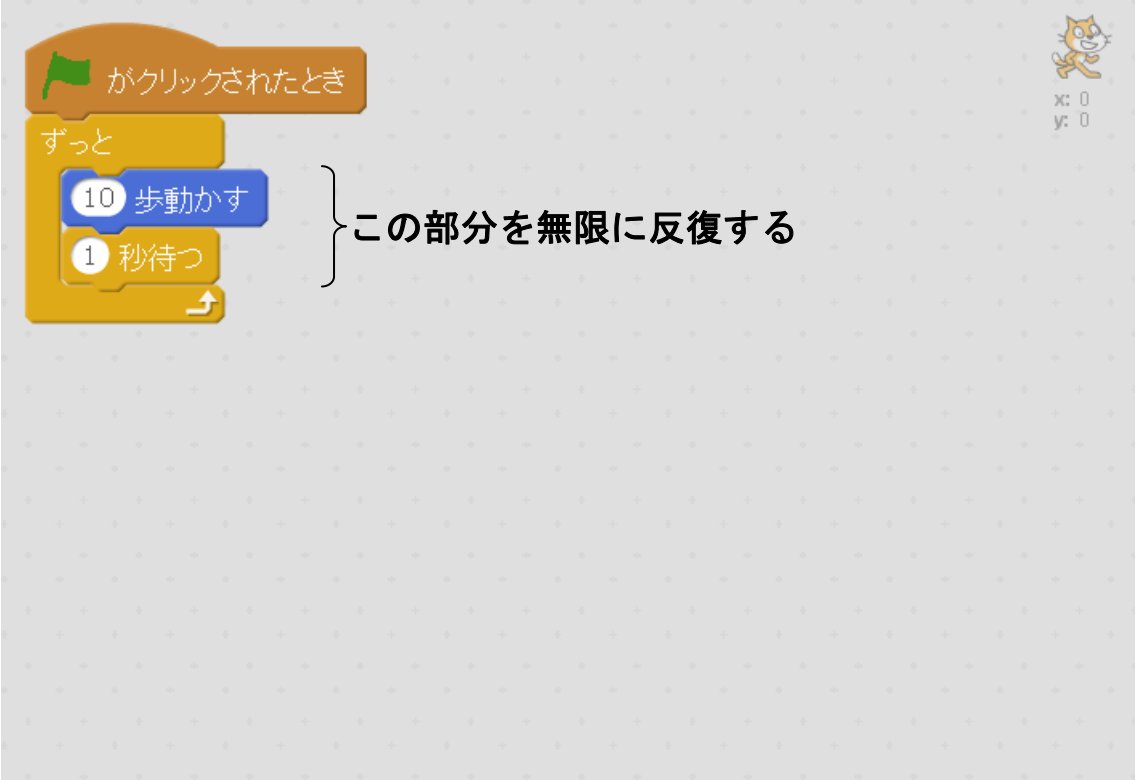
プログラミング言語では、たとえば、次のように書きます：

```
for (i=1; i<=10; i++) {
  動かす (10.0);
  待つ (1.0);
}
```

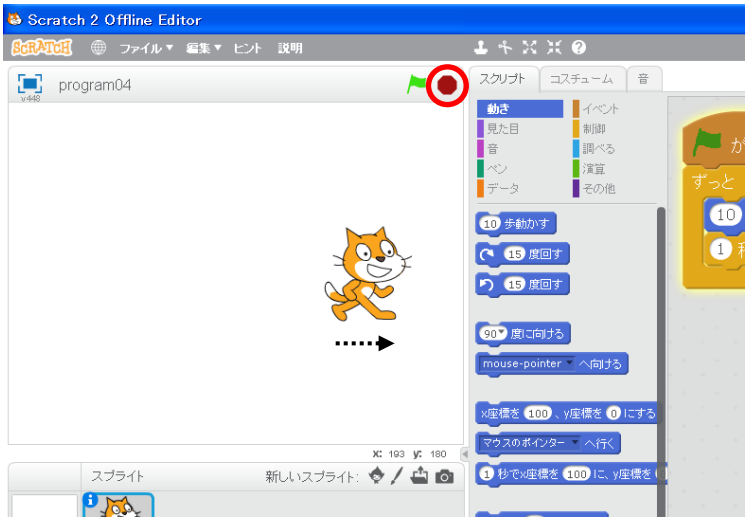
※代表的なプログラミング言語であるＣを例にして、有限回の反復処理の書き方を概念的に示しています。日本語の部分は架空の文法です。

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。



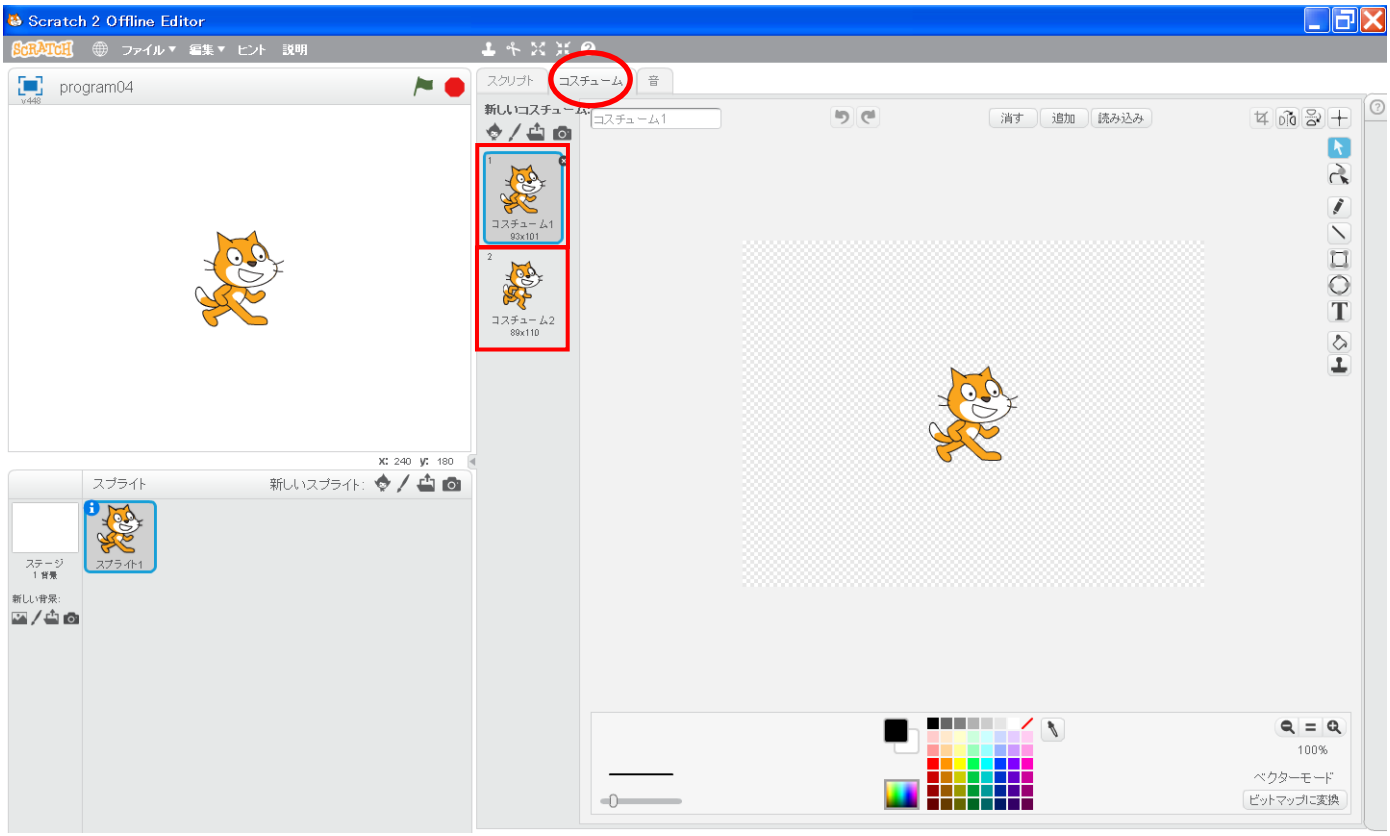
プログラミング言語では、たとえば、次のように書きます：

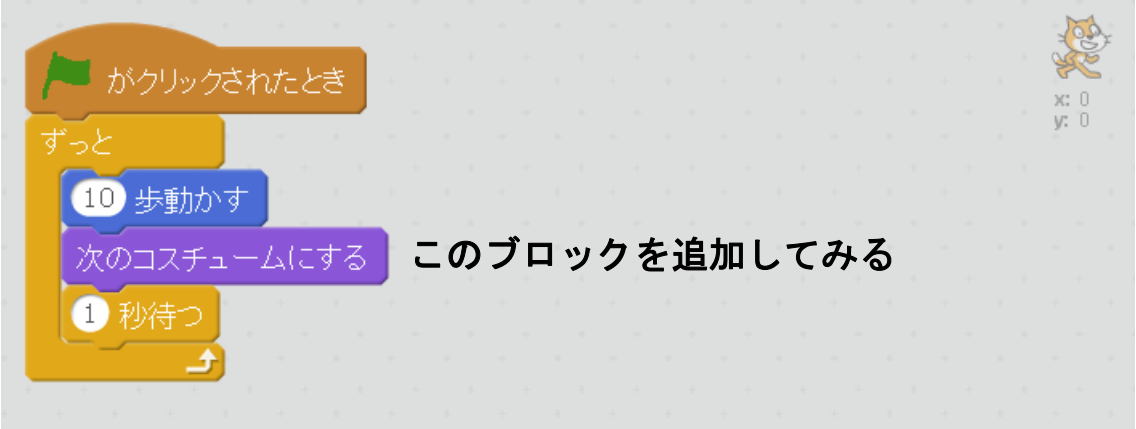
```
while (1){
  動かす (10.0);
  待つ (1.0);
}
```

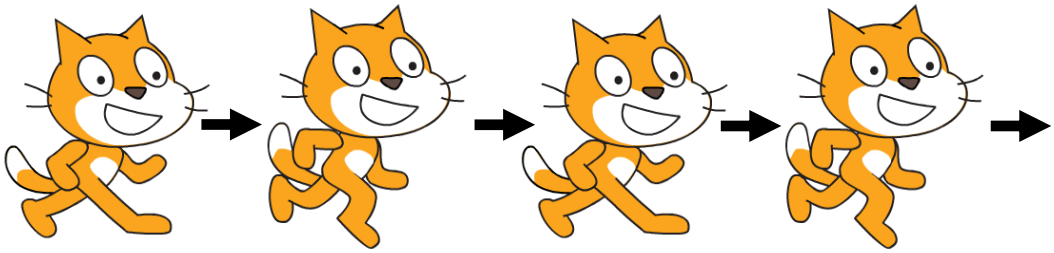
※代表的なプログラミング言語であるＣを例にして、無限回の反復処理の書き方を概念的に示しています。日本語の部分は架空の文法です。

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。



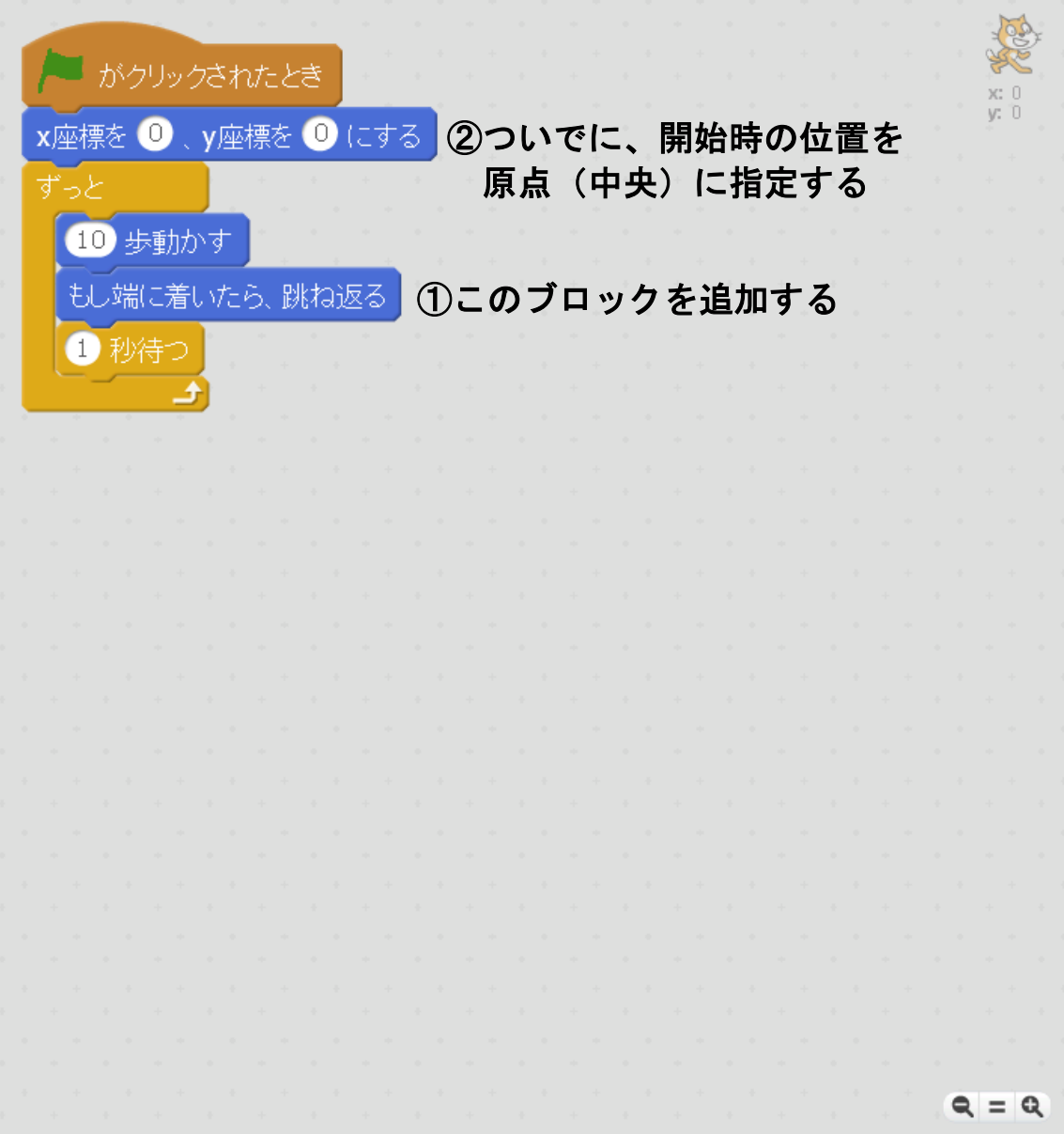




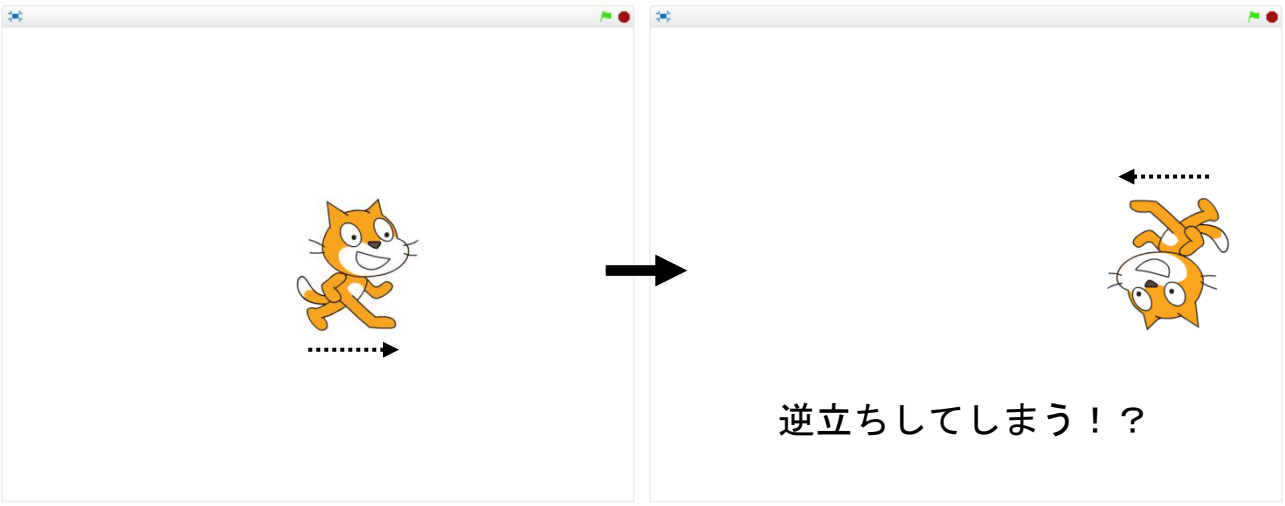
昔のテレビゲームは実際にこんな感じで動きを表現していたようです。

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。



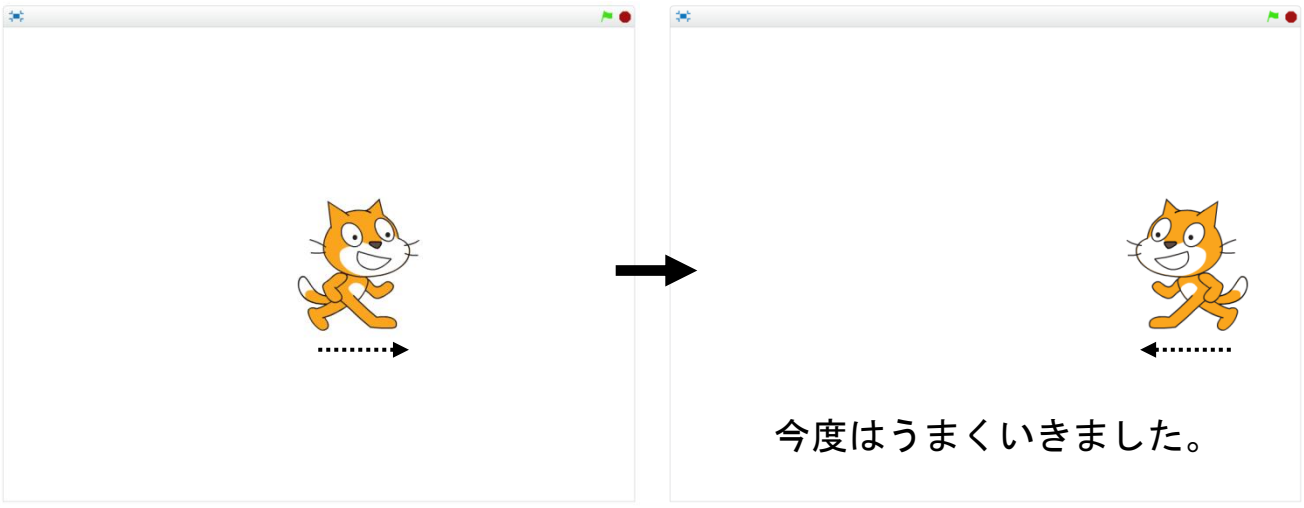
実行例



⑥逆立ちしないようにしよう

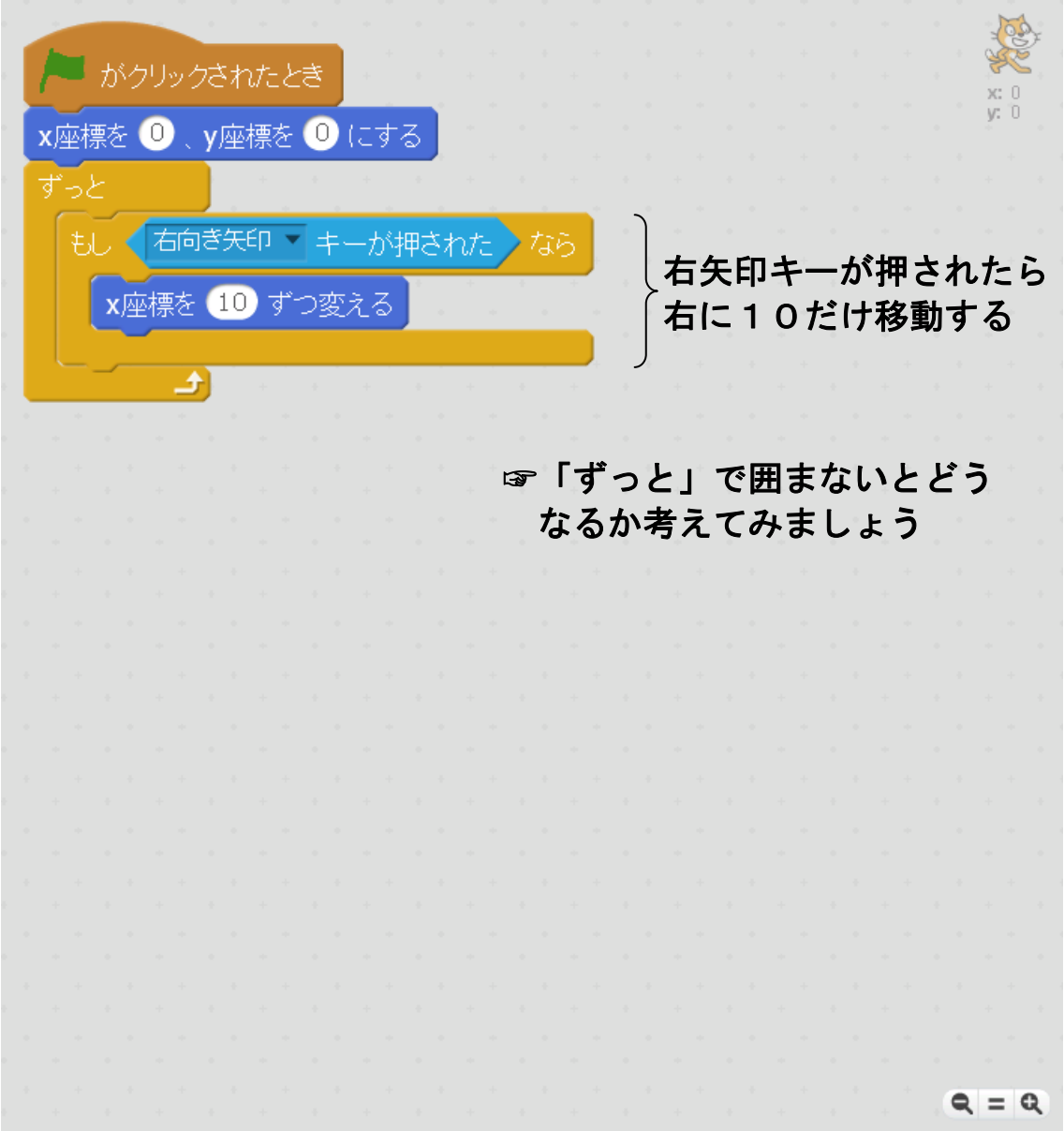
☞向きを変えたときに逆立ちしてしまわないように、開始時に設定を変更します。

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。



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。

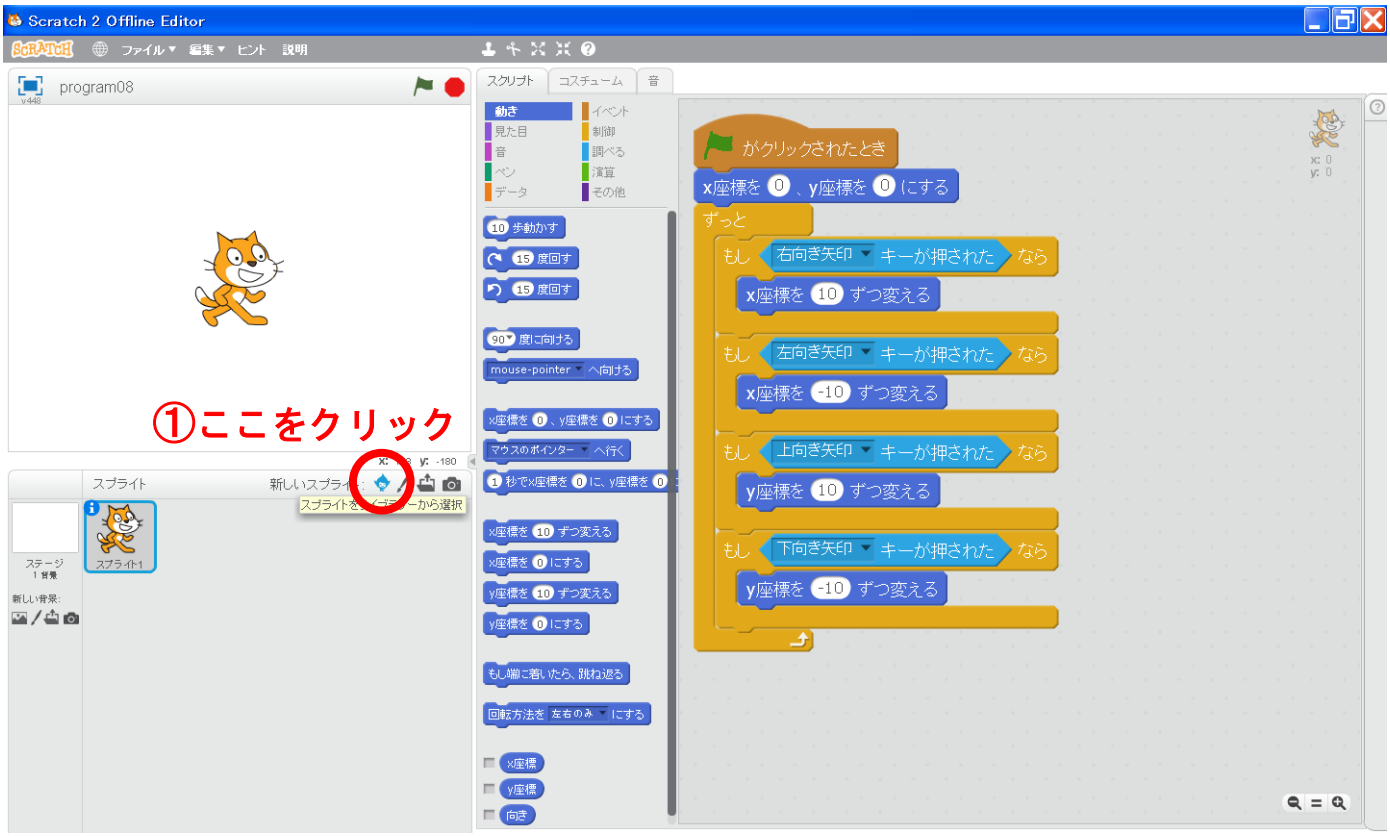
⑨ネズミを登場させよう

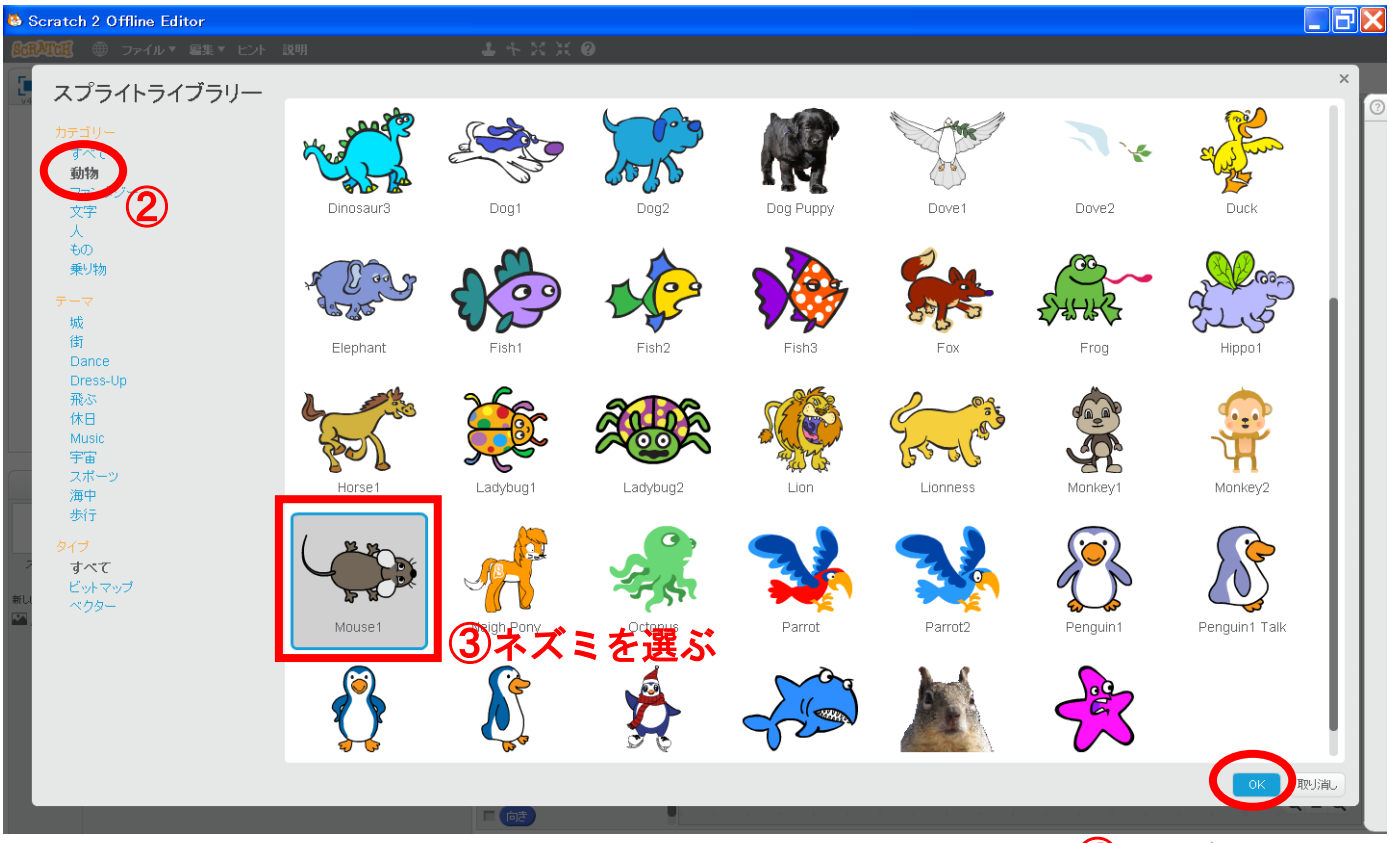
☞ネズミを登場させます。登場人物（スプライト）を追加し、ネズミのスクリプトを書きます。

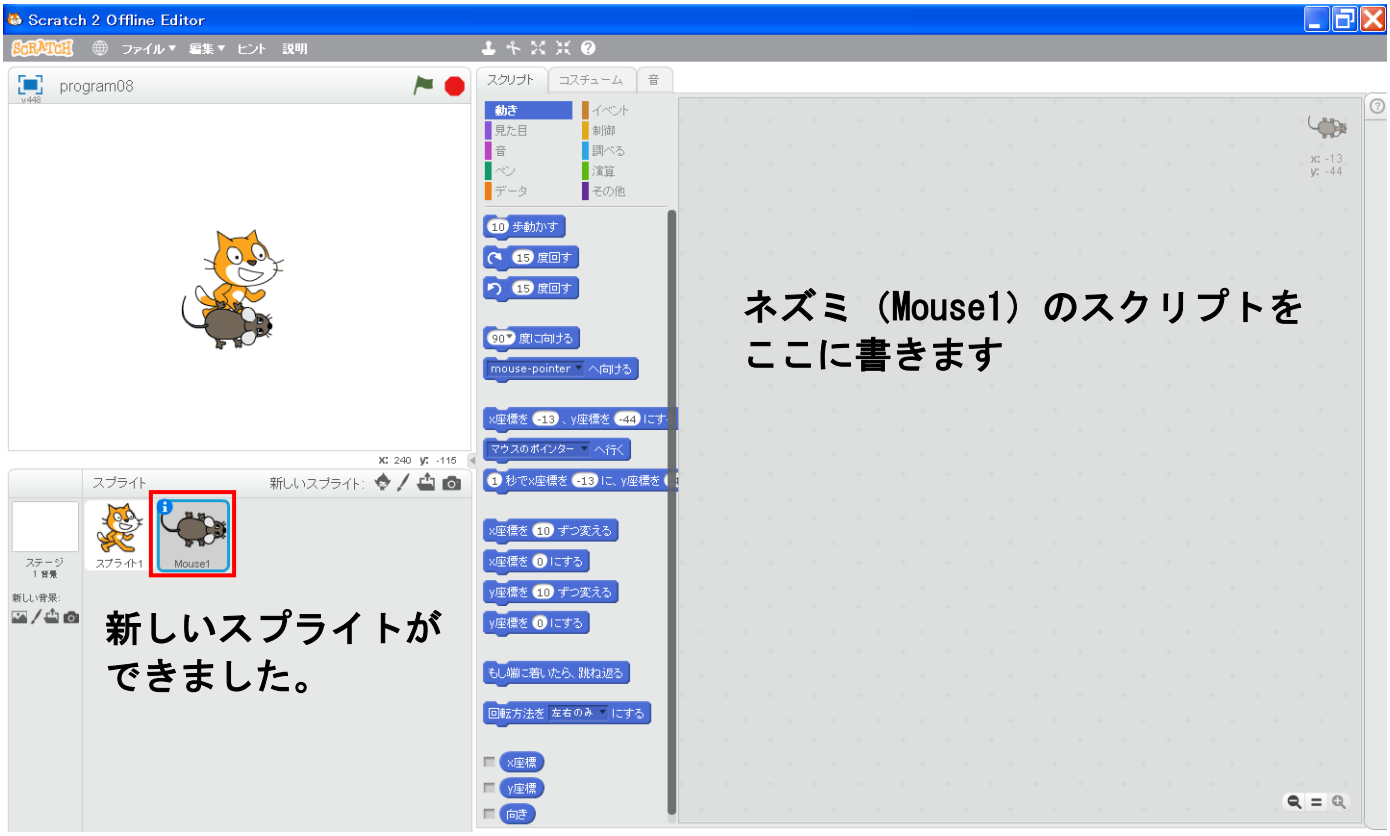
ネコ（スプライト１）

ネコのスクリプトに変更はありません

登場人物（スプライト）を追加する



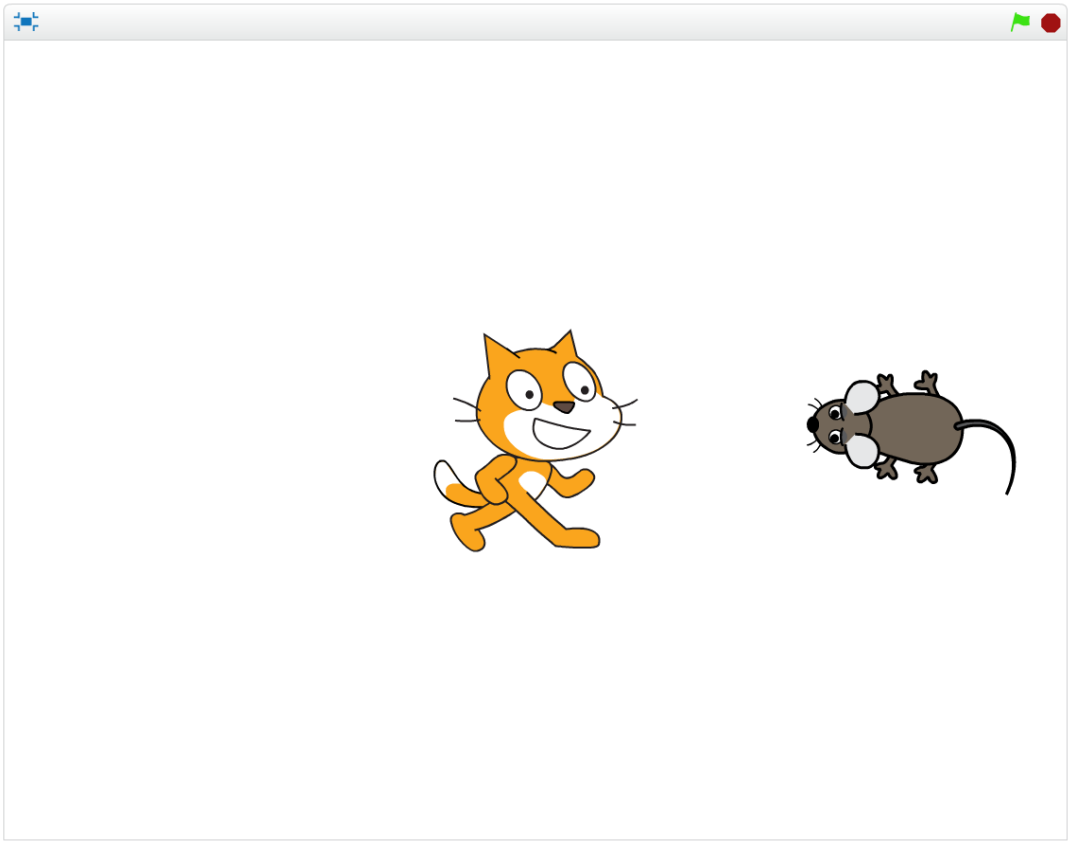



Scratch 2 Offline Editor
ファイル
編集
ヒント
説明
program08
スクリプト
コスチューム
音
動き
イベント
見た目
制御
音
調べる
ペン
演算
データ
その他
10 歩動かす
15 度回す
-15 度回す
90 度に向ける
mouse-pointer へ向ける
x座標を -13 、y座標を -44 にする
マウスのポインター へ行く
1 秒でx座標を -13 に、y座標を
x座標を 10 ずつ変える
x座標を 0 にする
y座標を 10 ずつ変える
y座標を 0 にする
もし端に着いたら、跳ね返る
回転方法を 左右のみ にする
x座標
y座標
向き
x: -13
y: -44
x: 240 y: -115
スプライト
新しいスプライト:
ステージ
1 背景
新しい背景:
スプライト1
Mouse1
新しいスプライトができました。
ネズミ（Mouse1）のスクリプトをここに書きます

ネズミ（Ｍｏｕｓｅ１）

ネズミのスクリプトを新しく書きます。

実行例



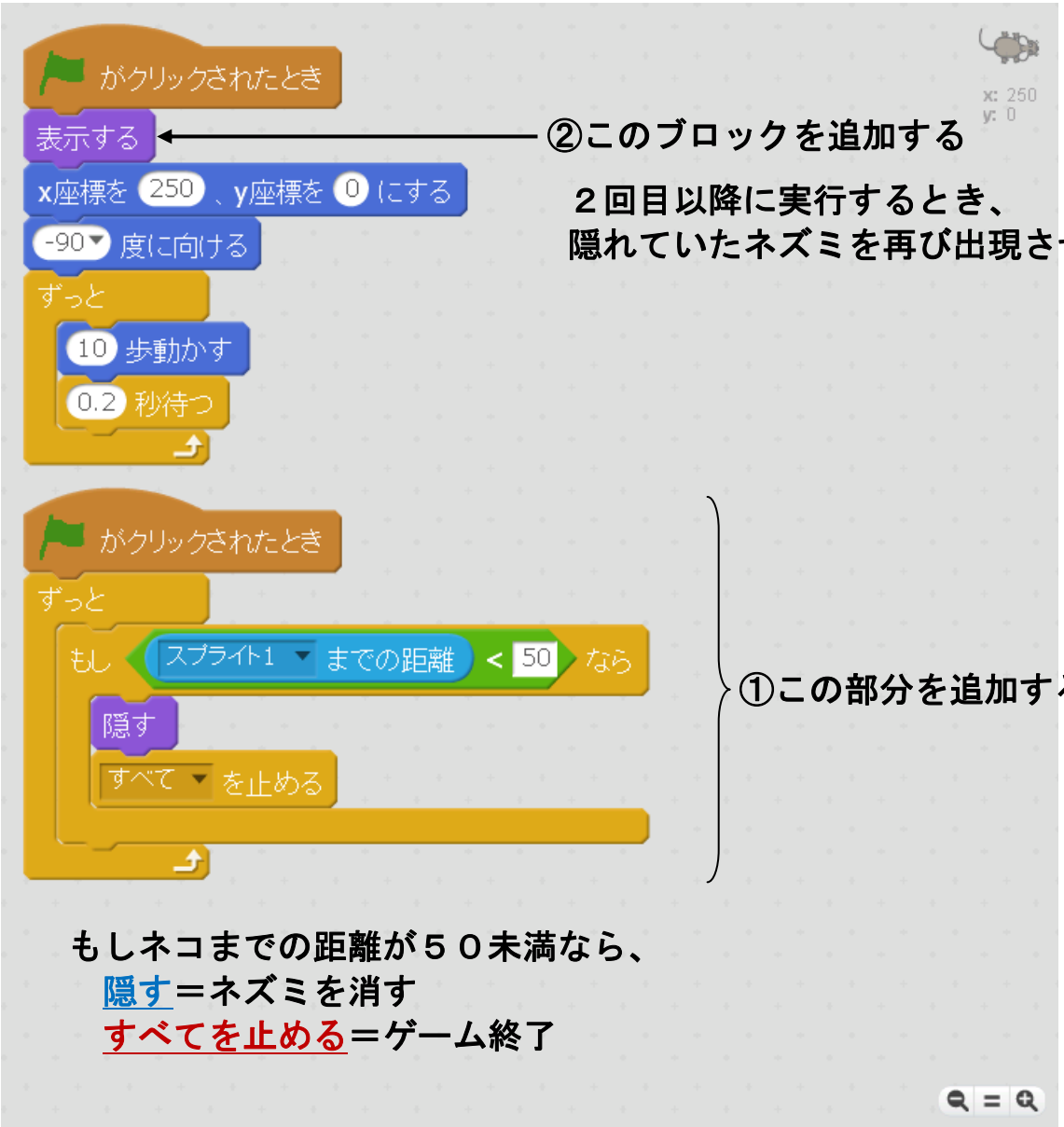
⑩ネズミを食べよう

☞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）



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。

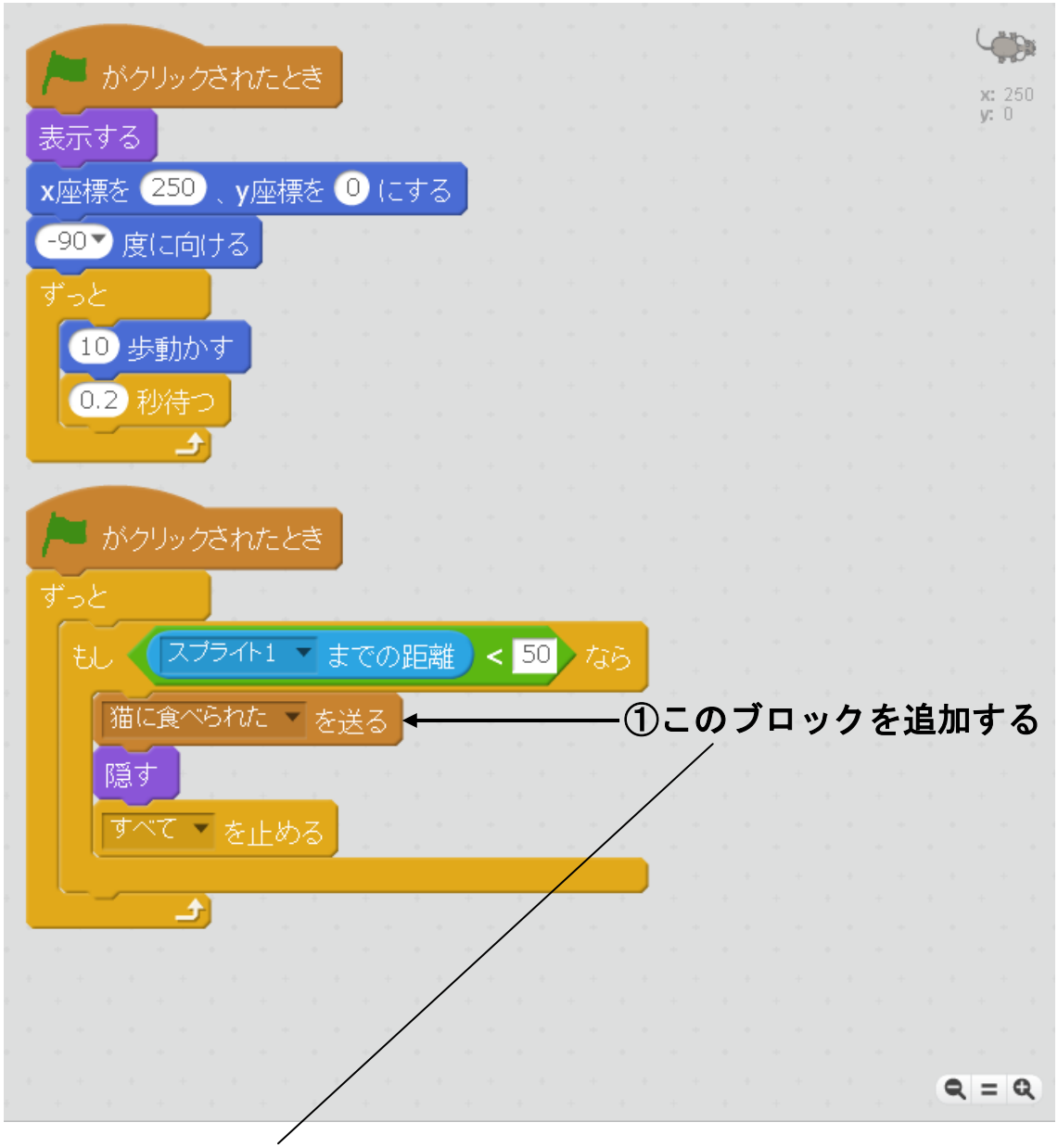
「ニャー」と鳴くのは、ネズミではなくネコです。そこで次のような手順でネコを鳴かせます。

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。

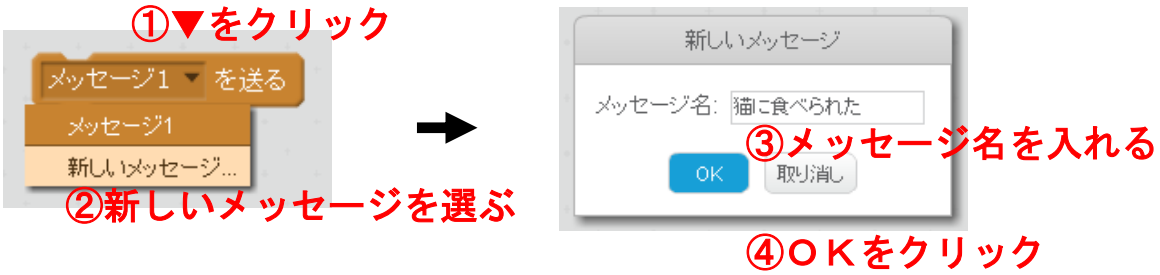
ネコ（スプライト１）　　<次のページを先にやってからこのページに戻ります>

②この部分を追加する

ネズミ（Ｍｏｕｓｅ１）



※メッセージを定義する方法



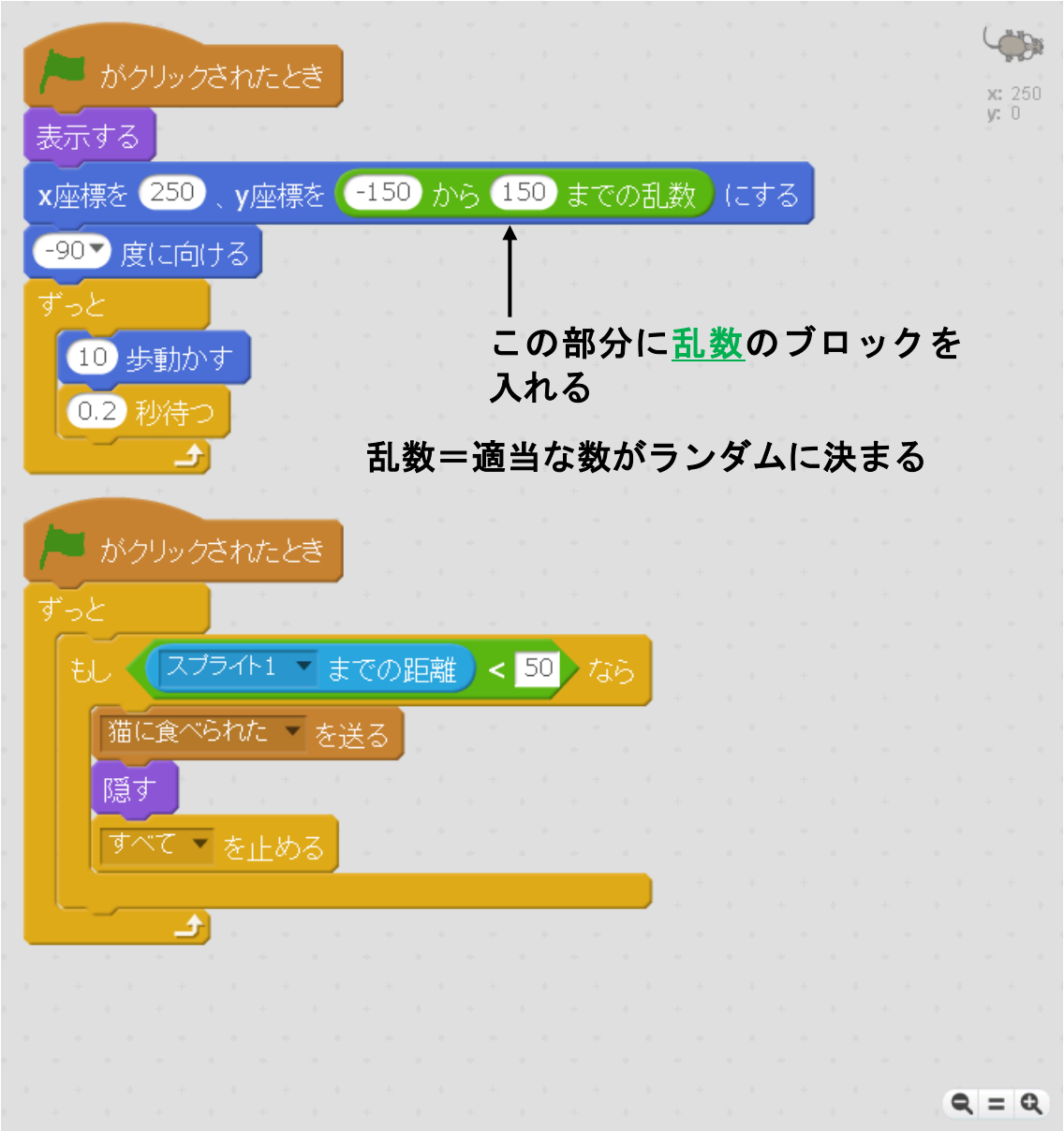
⑫ネズミをランダムな位置から登場させよう

☞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（Ｍｏｕｓｅ１）



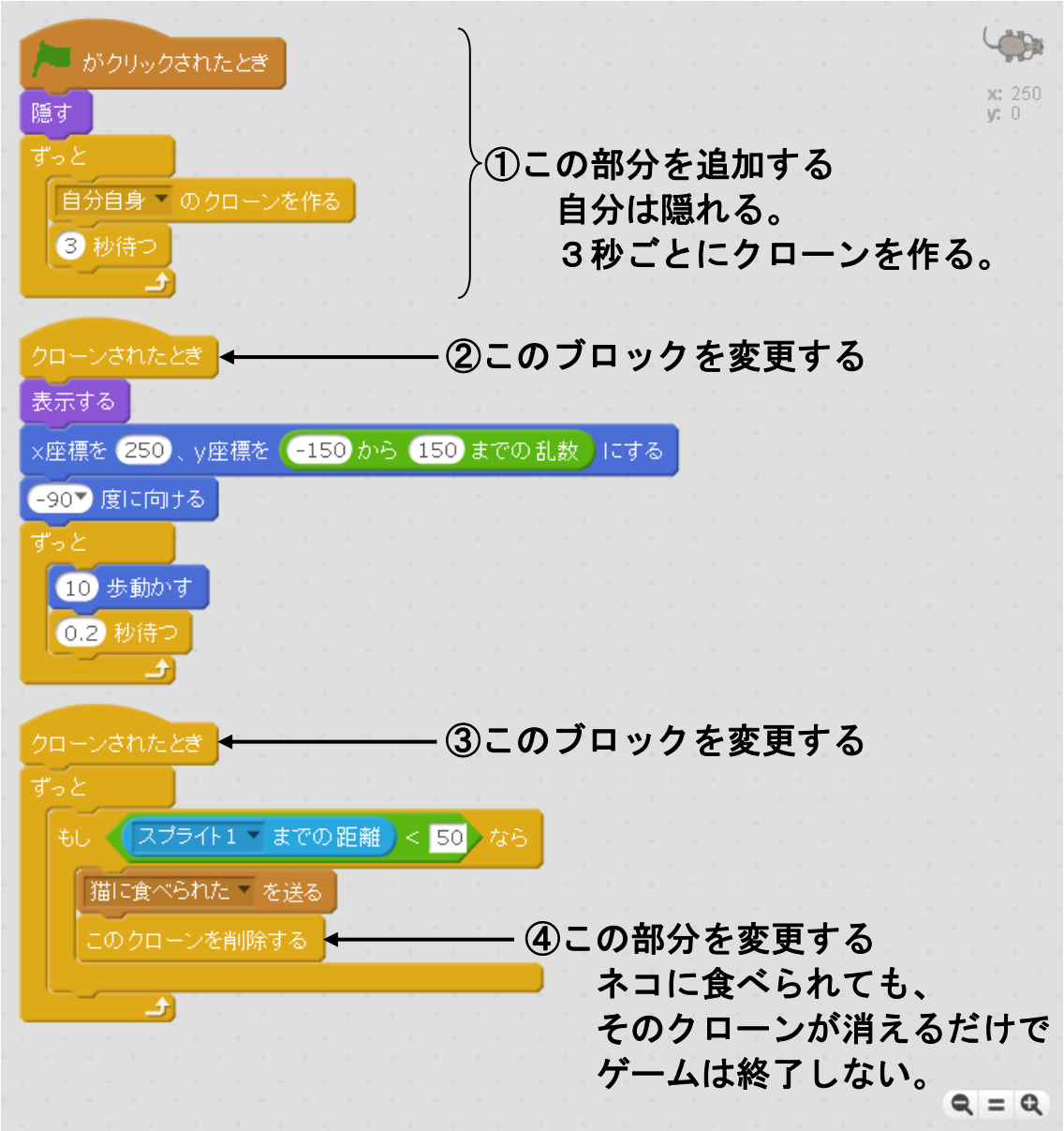
⑬ネズミをたくさん登場させよう

☞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。

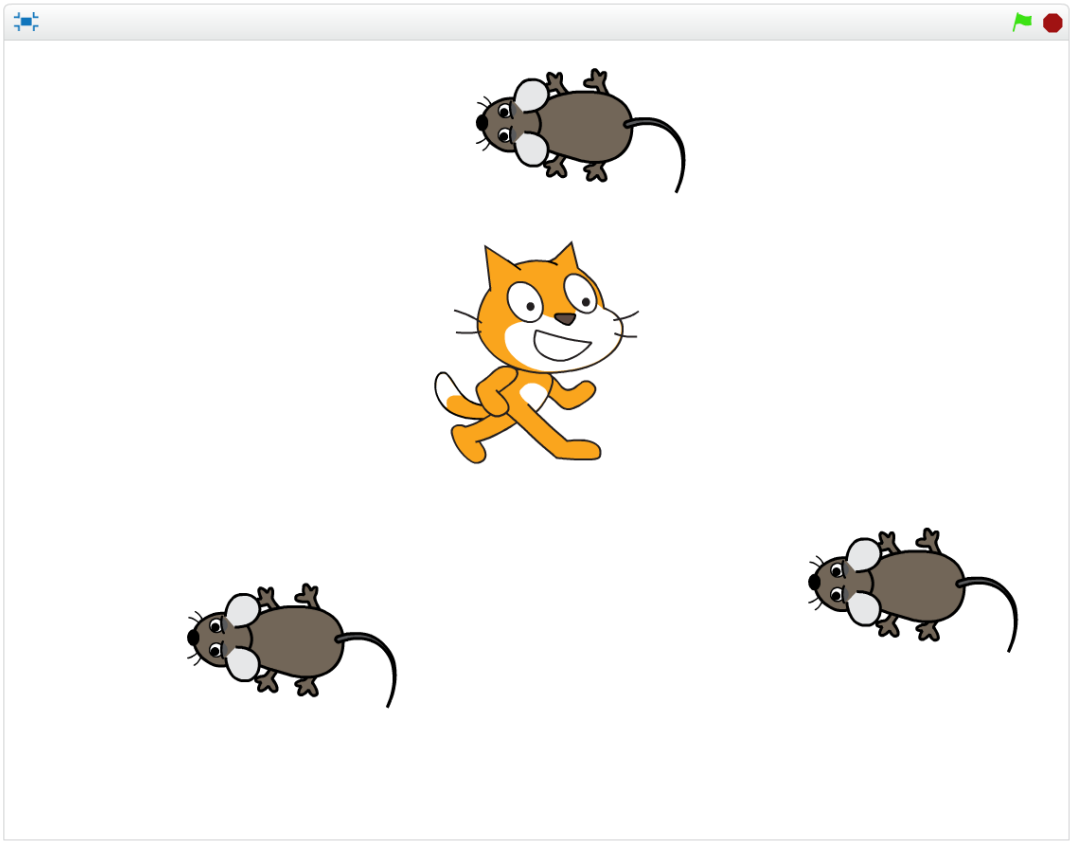
ネコ（スプライト１）

変更はありません。

ネズミ（Ｍｏｕｓｅ１）



実行例



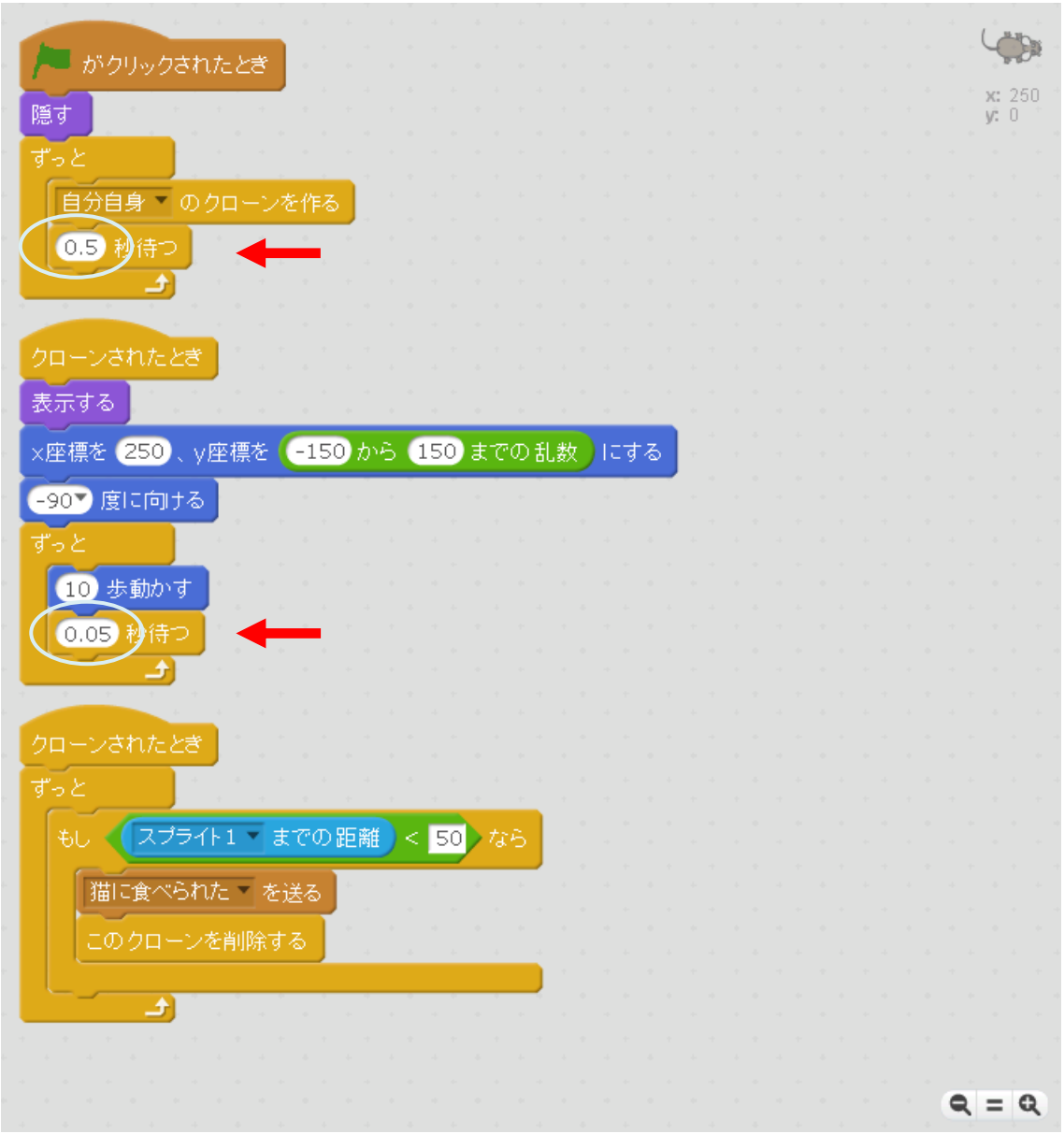
⑭ゲームの進行を速くしよう

☞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。

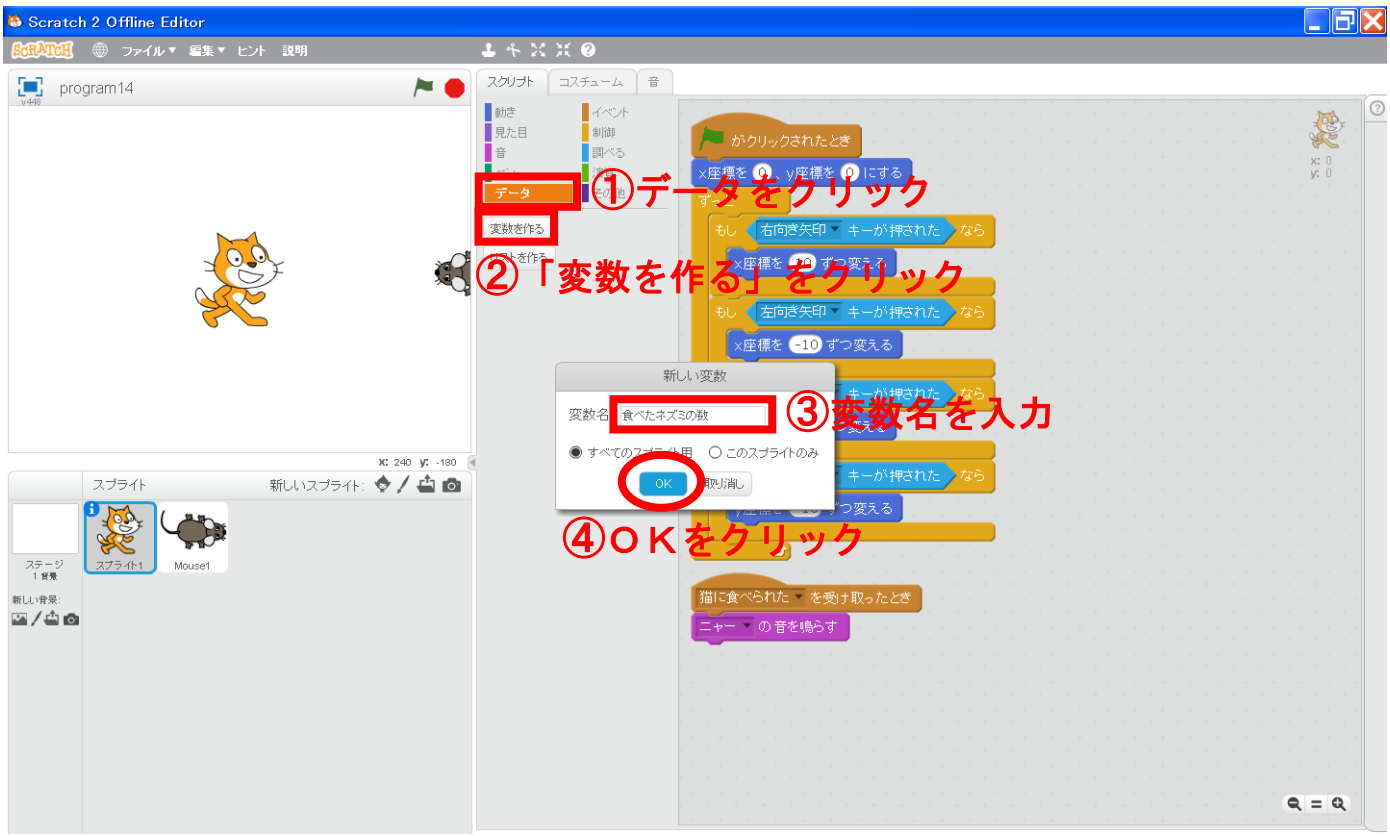
ネズミ（Ｍｏｕｓｅ１）



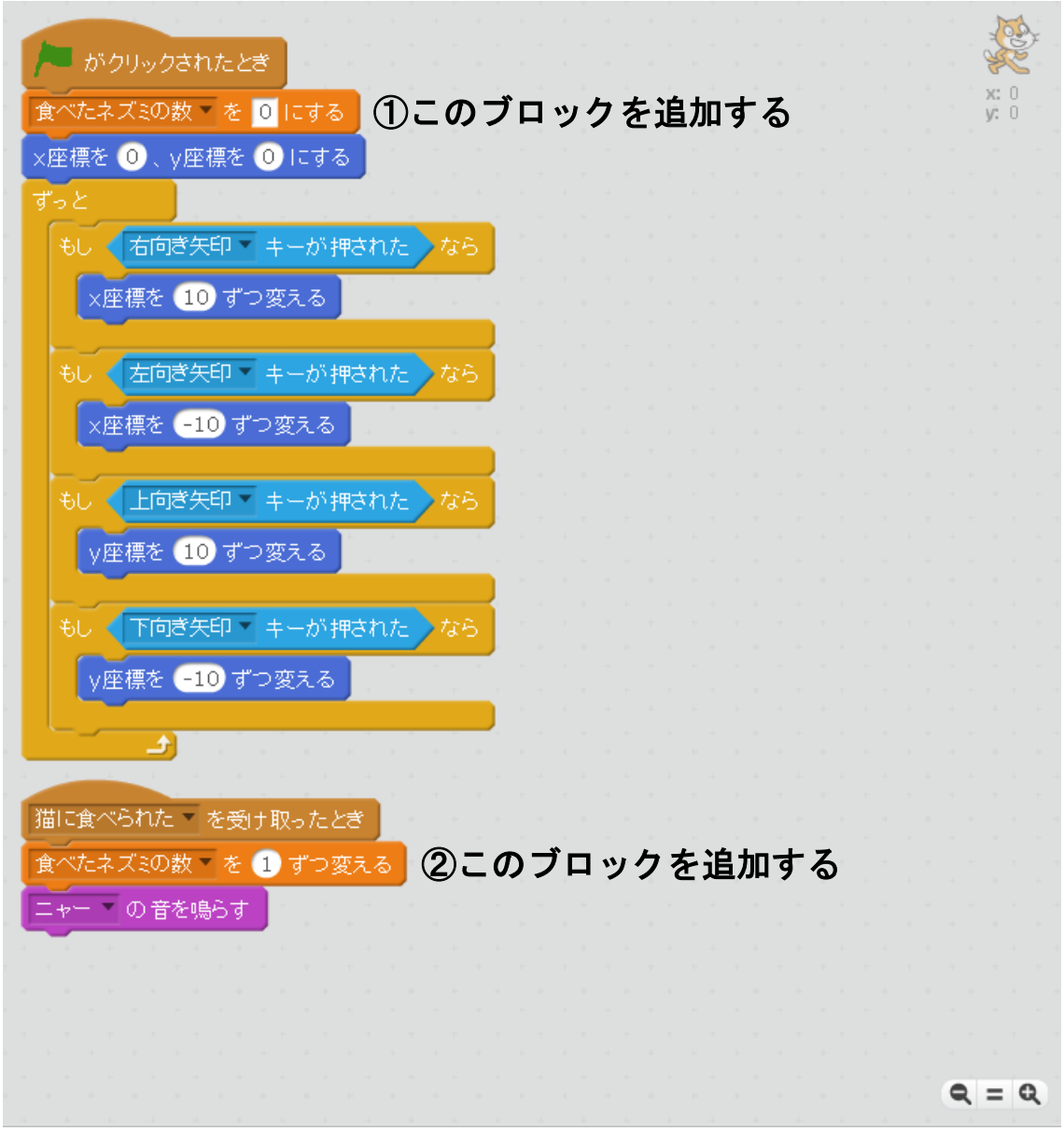
⑮食べたネズミの数を数えよう

☞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



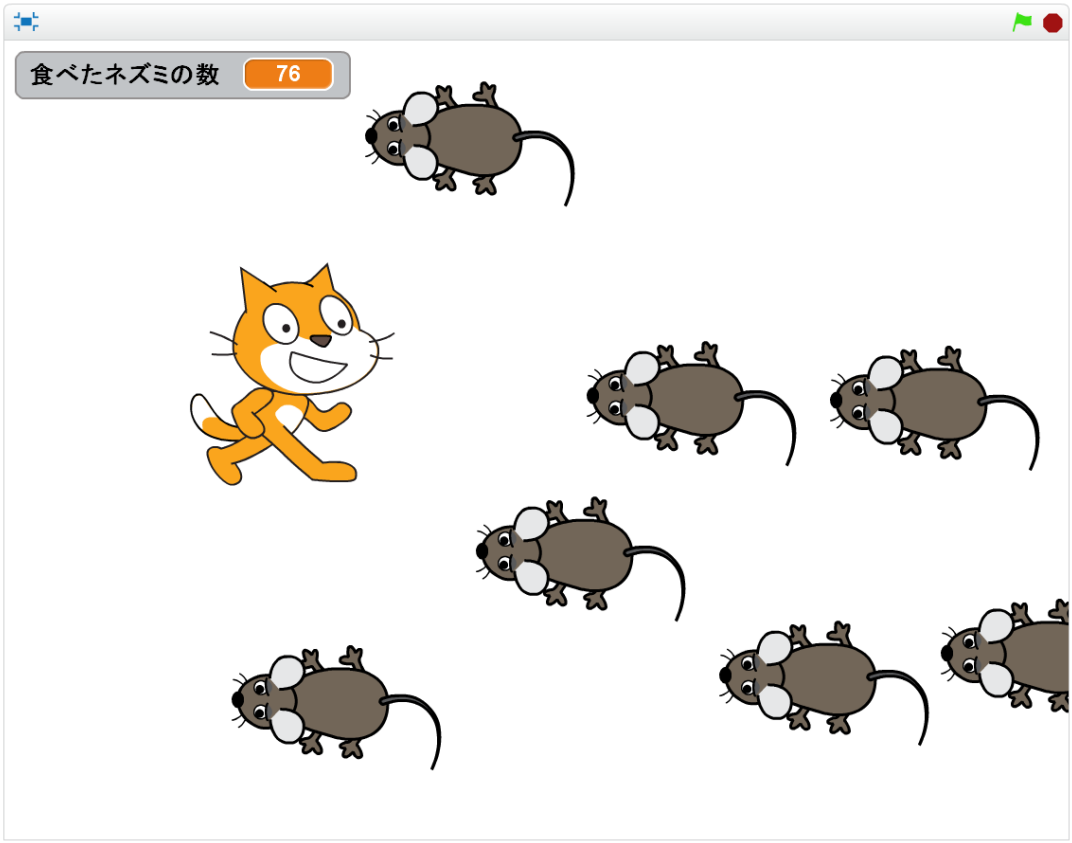
ネコ（スプライト１）



ネズミ（Ｍｏｕｓｅ１）

変更はありません。

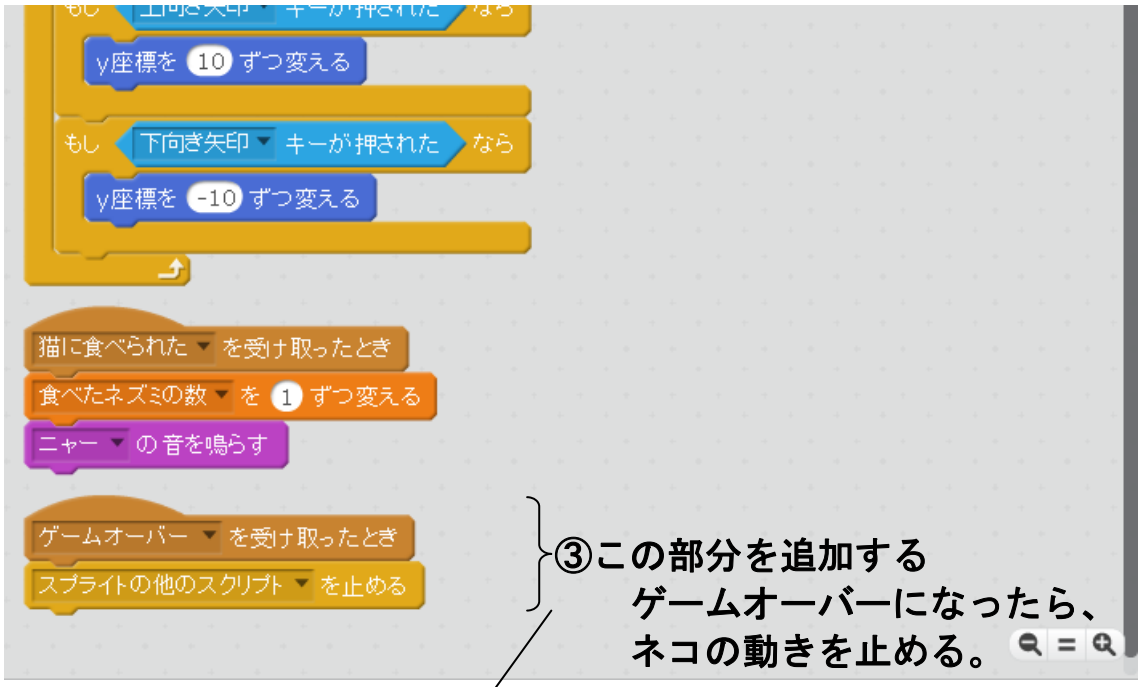
実行例



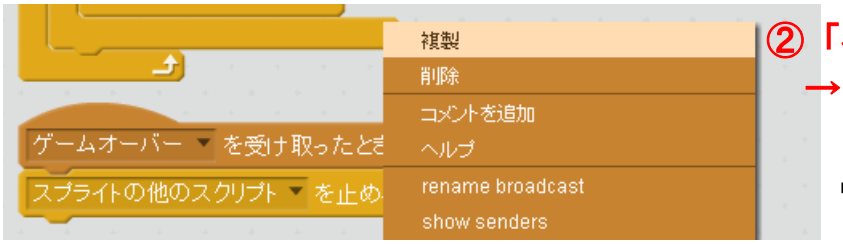
⑯ネズミを取り逃がしたらゲームオーバーにしよう

☞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）　　<次のページを先にやってからこのページに戻ります>



※③は、ネズミのほうで作った②をコピーしてくると便利です。



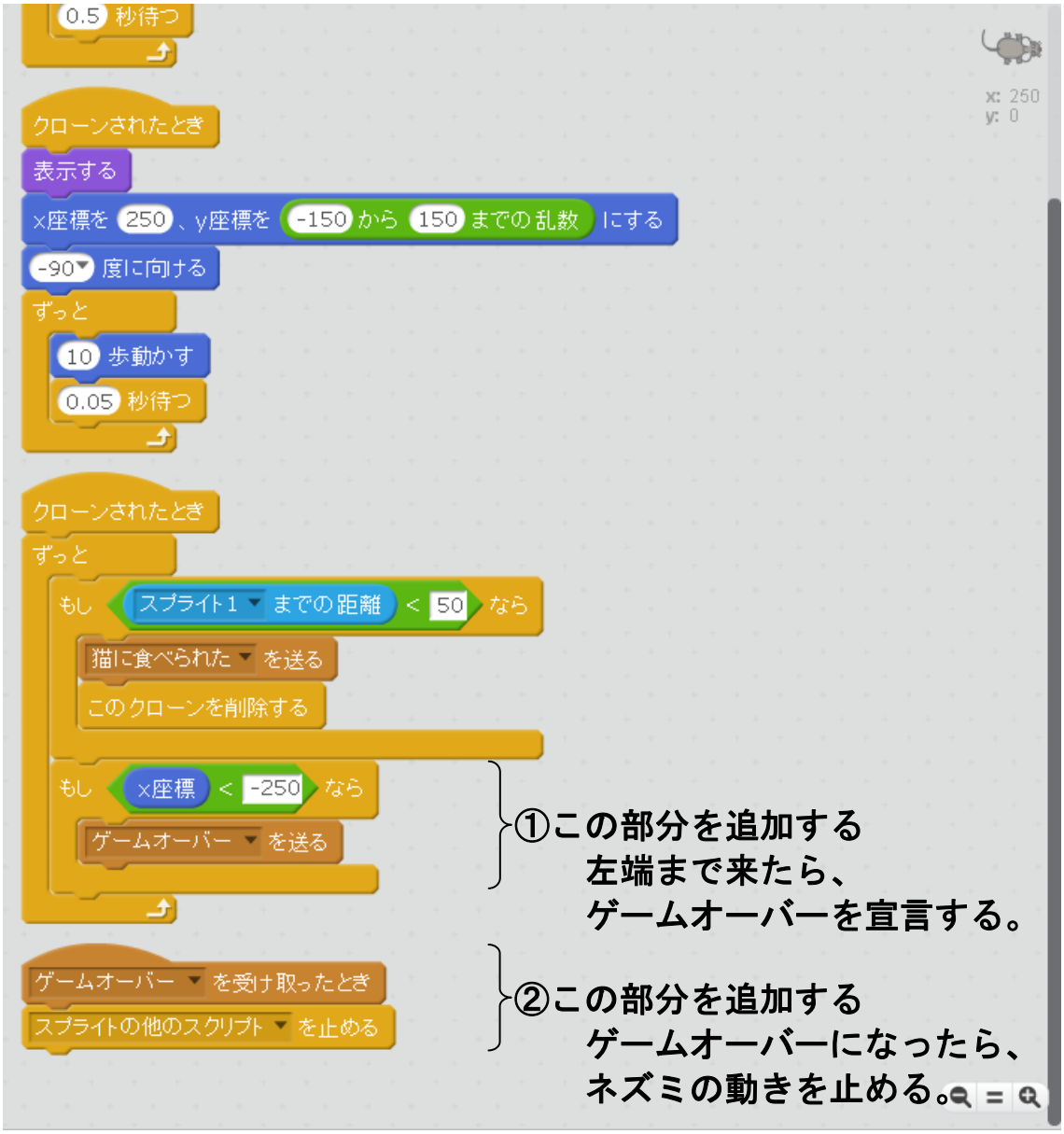
②「複製」を選ぶ
→その場でコピーされます

①「ゲームオーバーを受け取ったとき」を右クリック

③マウスを動かし、
「スプライト１」まで移動して
左クリック

④ネコ（スプライト１）のスクリプトエリアに移動して、
ブロックのレイアウトを調整する
（スクリプトエリアの最初の部分にコピーされています）

ネズミ（Ｍｏｕｓｅ１）



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。

これでいちおう完成しましたが、少しだけ手を加えて、さらに楽しんでみましょう。

⑰ネズミから逃げるゲームにしてみましょう

☞ネズミが苦手なネコもいるかもしれません。ネズミを取り逃がしたときではなく、ネコがネズミに捕まったら、ゲームを止めるようにします。

はじめに、変更前のプログラムを保存しておきましょう。

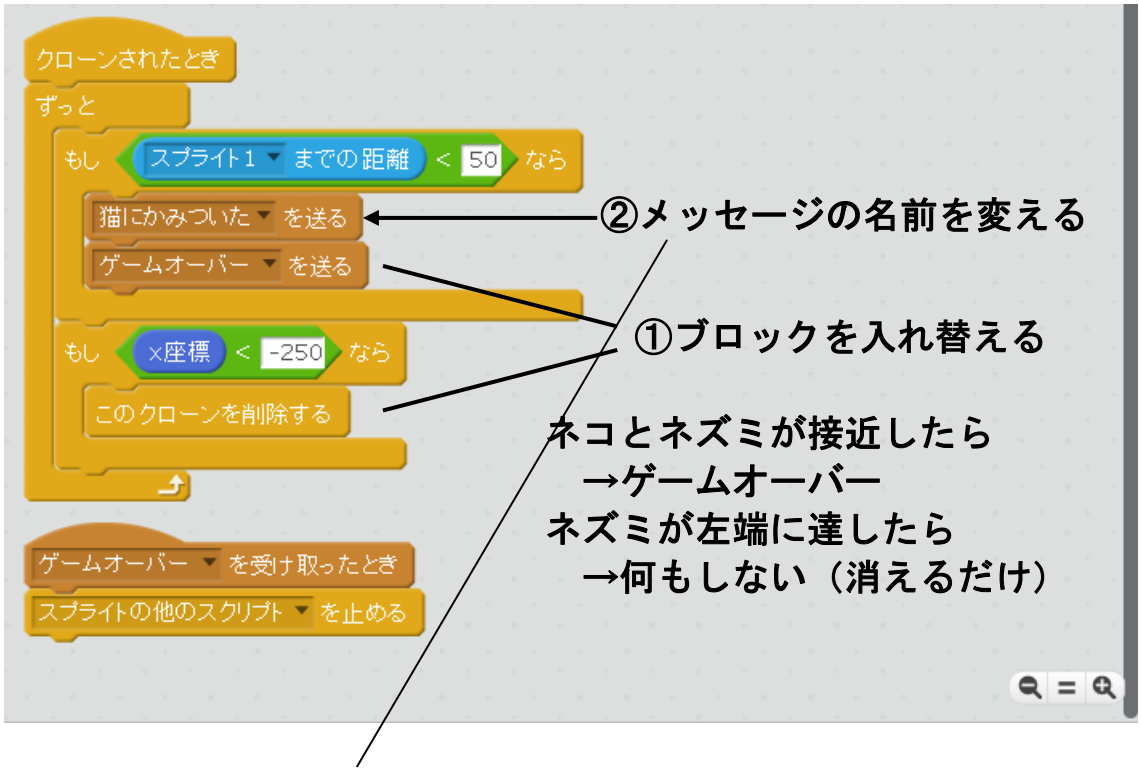
一旦、scratch を終了して、プログラムを右クリックして「コピー」、「貼り付け」で。

保存し終わったら…

ネコ（スプライト１）

変更はありません。

ネズミ（Ｍｏｕｓｅ１）



※メッセージの名前を変える方法

ネズミはこれまでと同じように登場します。実際にゲームをやってみると、少し変ですね。

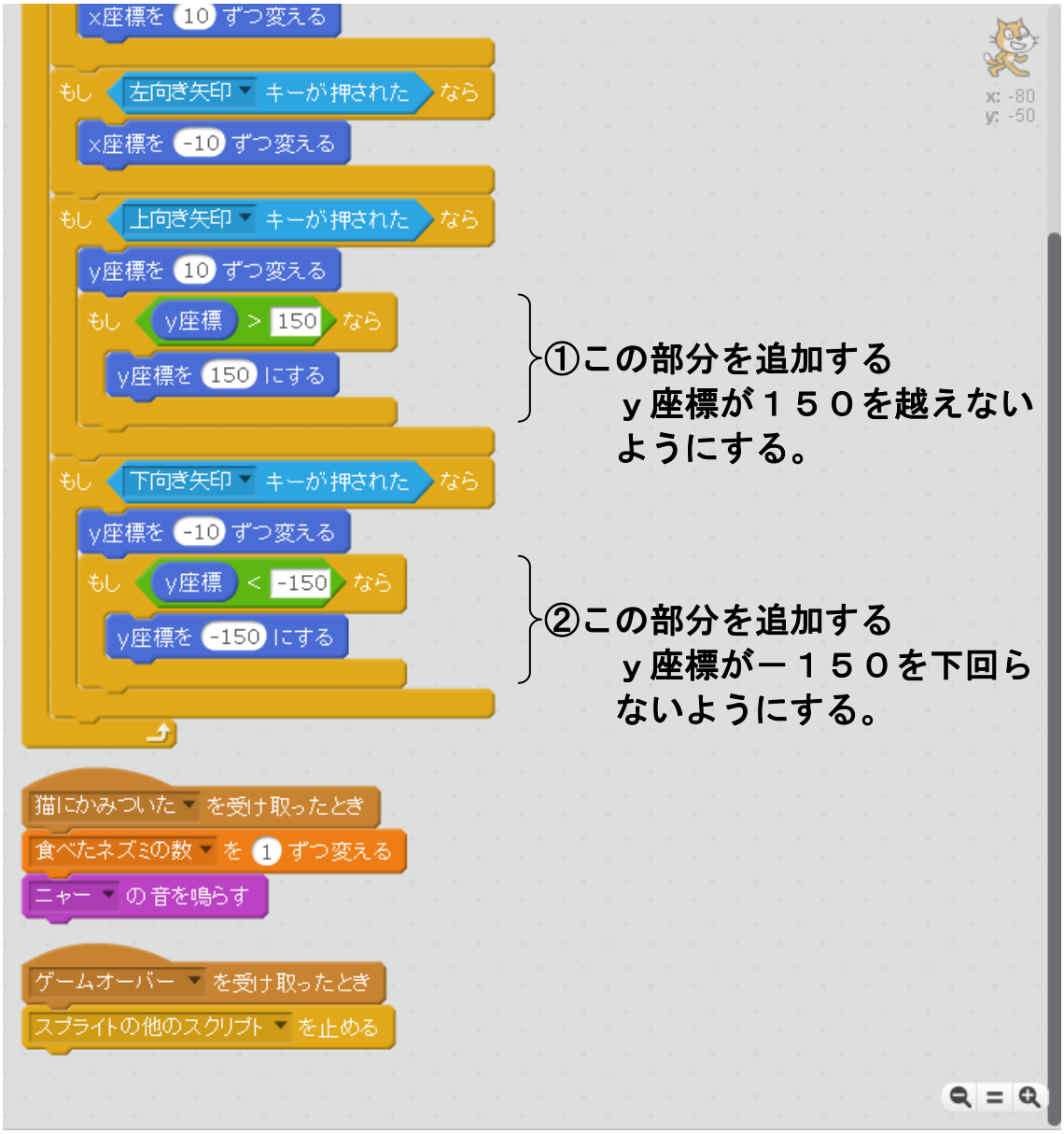
・すべてのネズミをよけるのは難しい。　→どんなにがんばってもゲームオーバー。。。
・画面の上下どちらかの端に逃げれば、絶対にネズミに捕まらない。　→つまらない裏技。。。

設定を調整してみましょう。

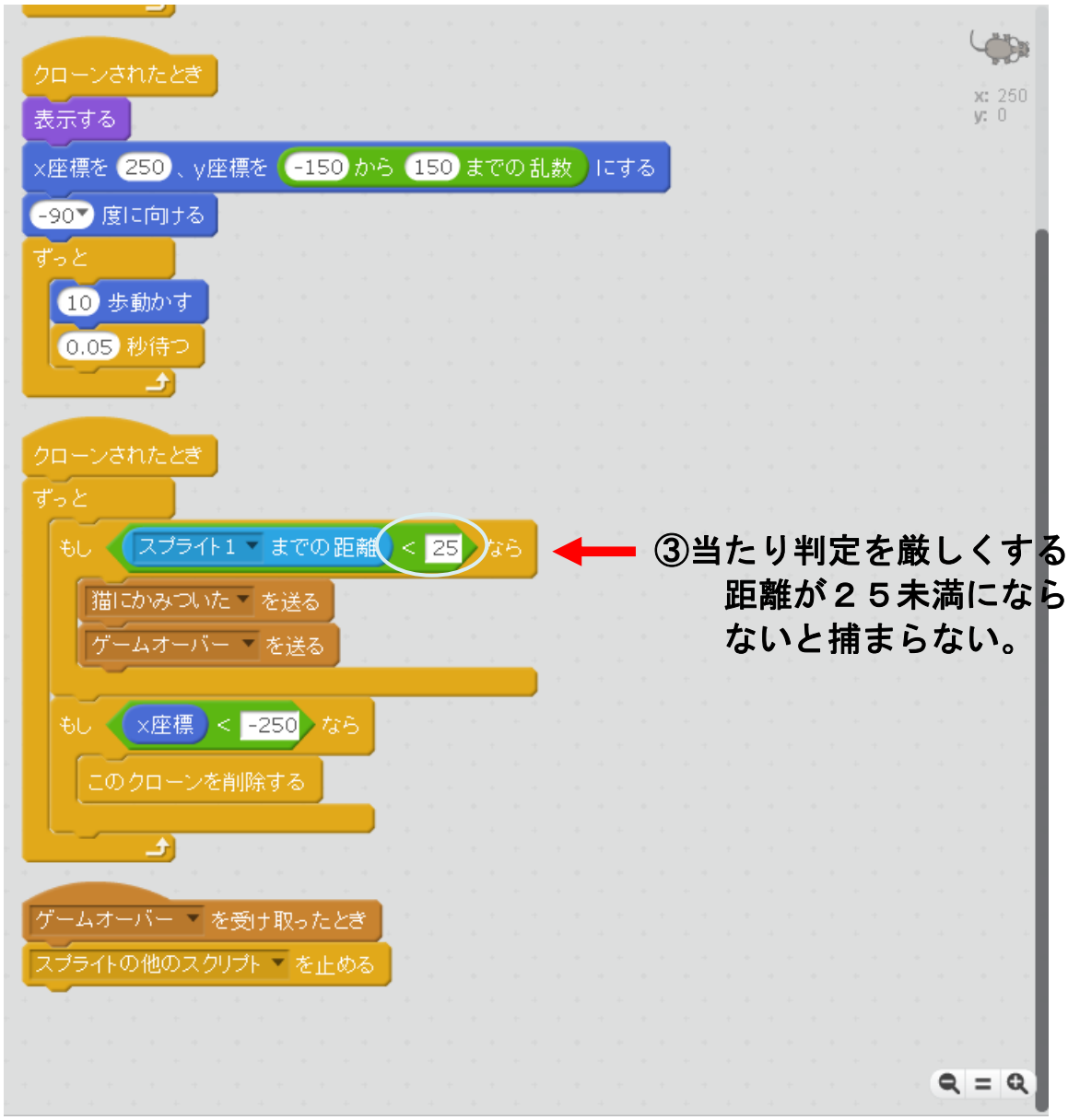
⑱設定を調整しよう

☞設定を調整します。ネコが動ける範囲を制限して、それから、当たり判定を厳しくします。

ネコ（スプライト１）



ネズミ（Ｍｏｕｓｅ１）



③当たり判定を厳しくする。
距離が２５未満にならないと捕まらない。

これで普通の難易度のゲームになったと思います。

⑲イヌをよけながらネズミを捕まえるゲームにしよう

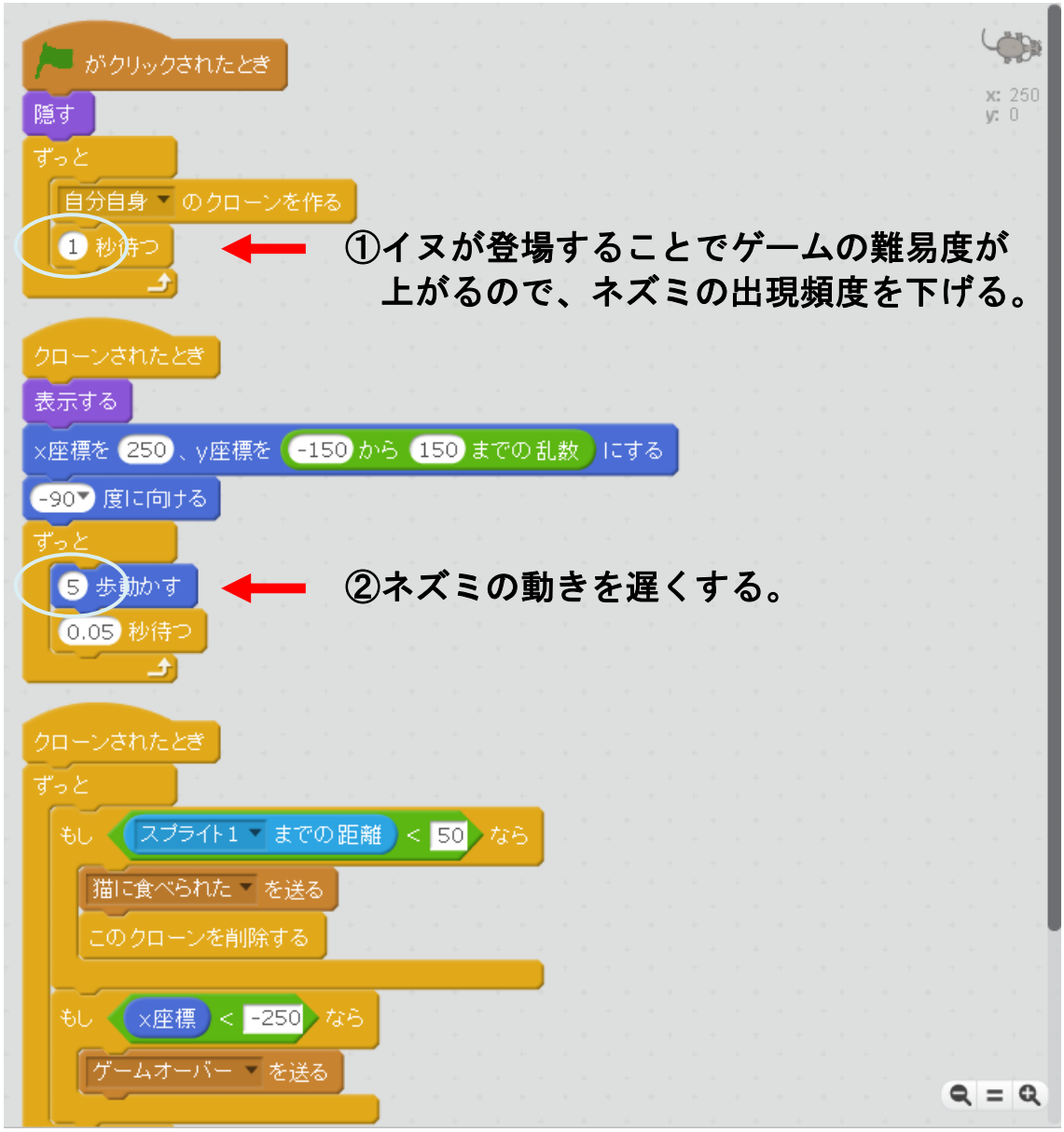
☞⑱で書いたプログラムを参考に、イヌをよけながらネズミを捕まえるゲームを作りましょう。

ネコ（スプライト１）

⑯に戻します。

ネズミ（Ｍｏｕｓｅ１）

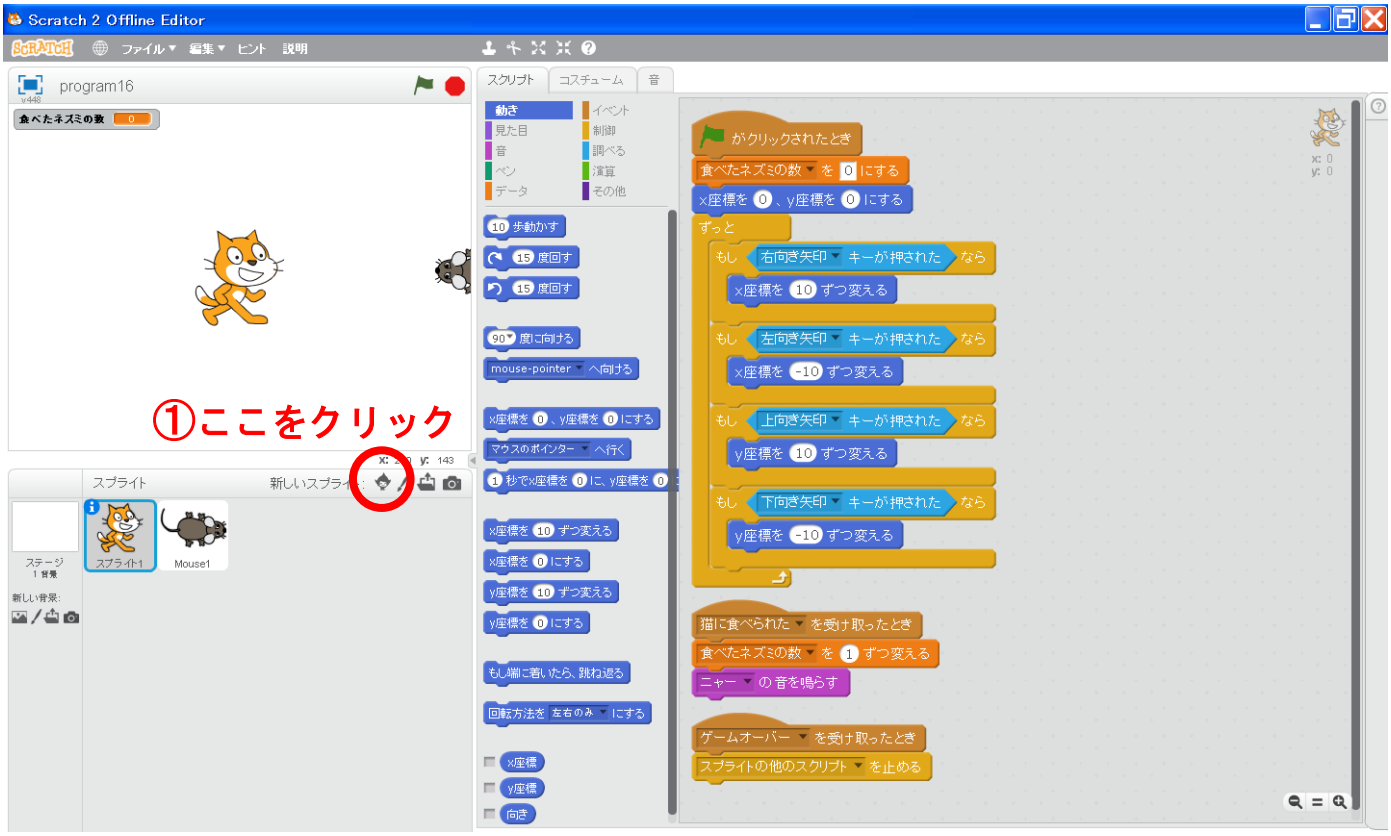
⑯に戻した後で、少しだけ変更します。

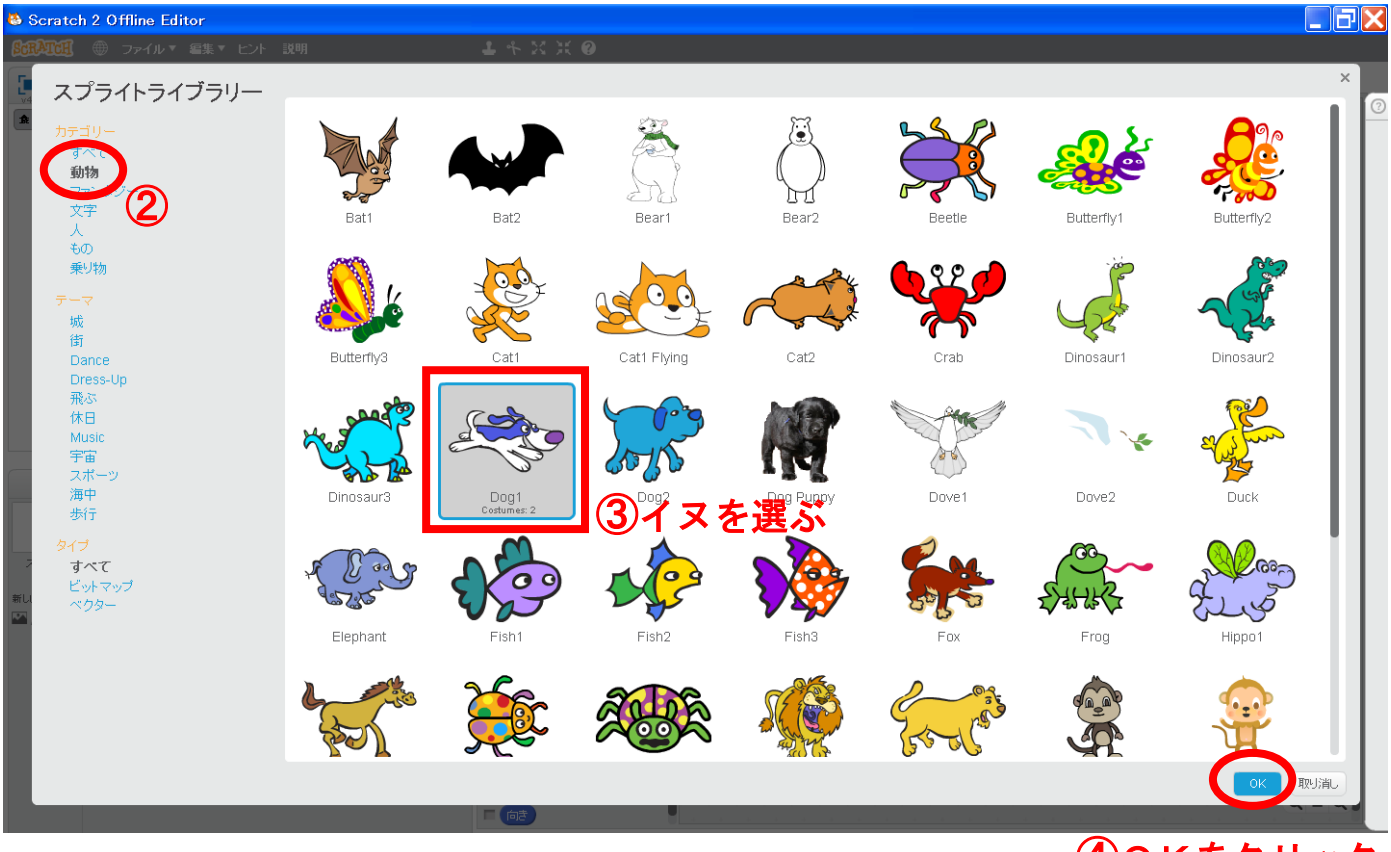


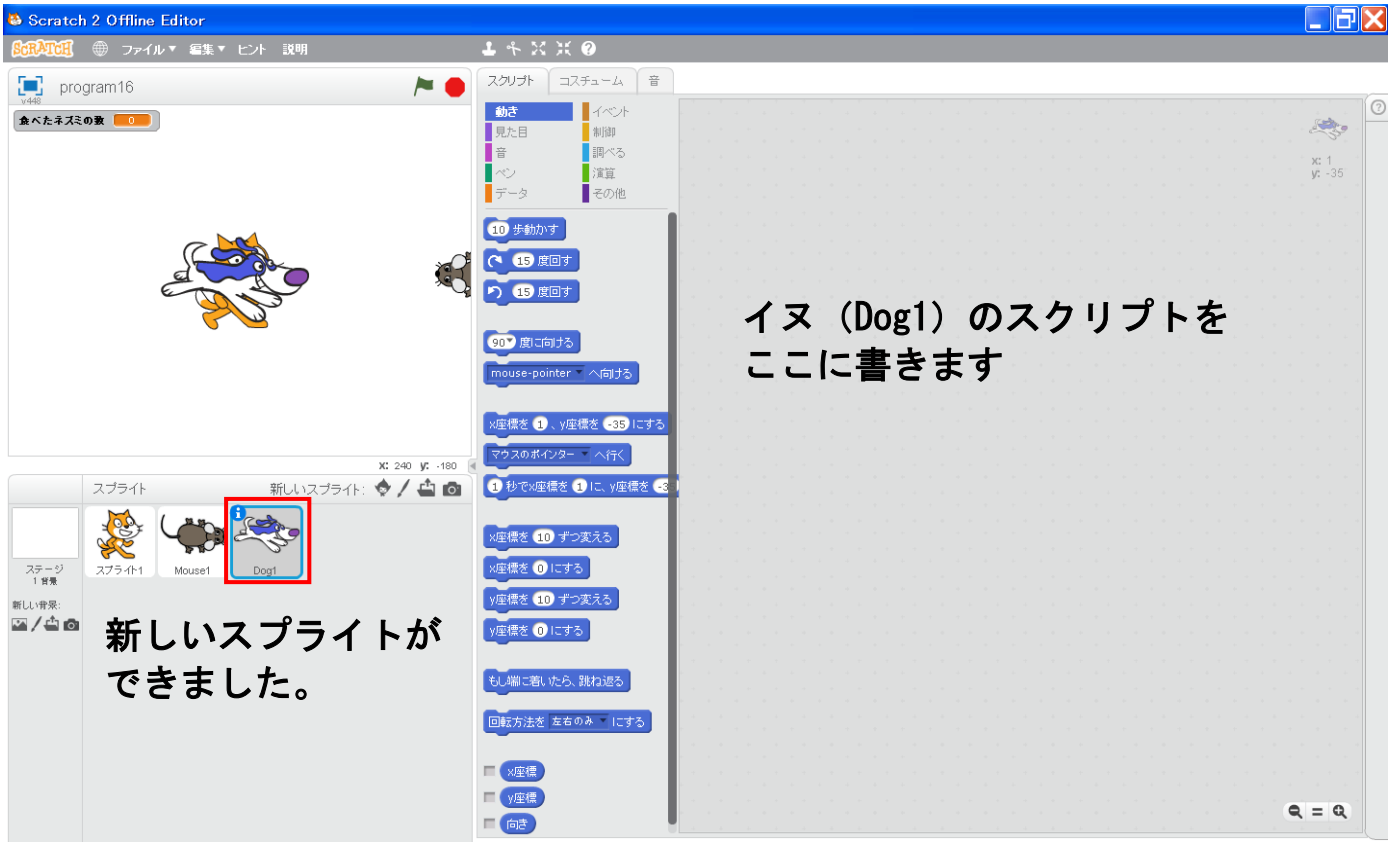
イヌ（Ｄｏｇ１）

イヌを登場させます。登場人物（スプライト）を追加し、イヌのスクリプトを書きます。

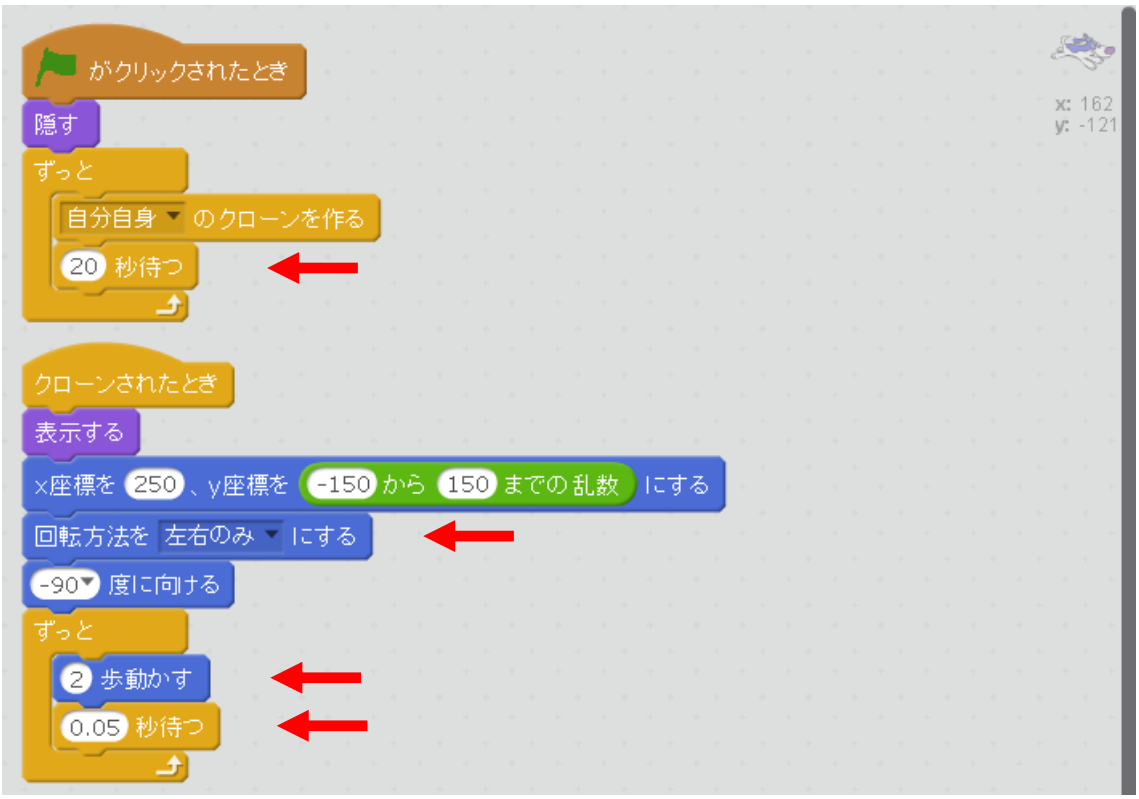
登場人物（スプライト）を追加します。



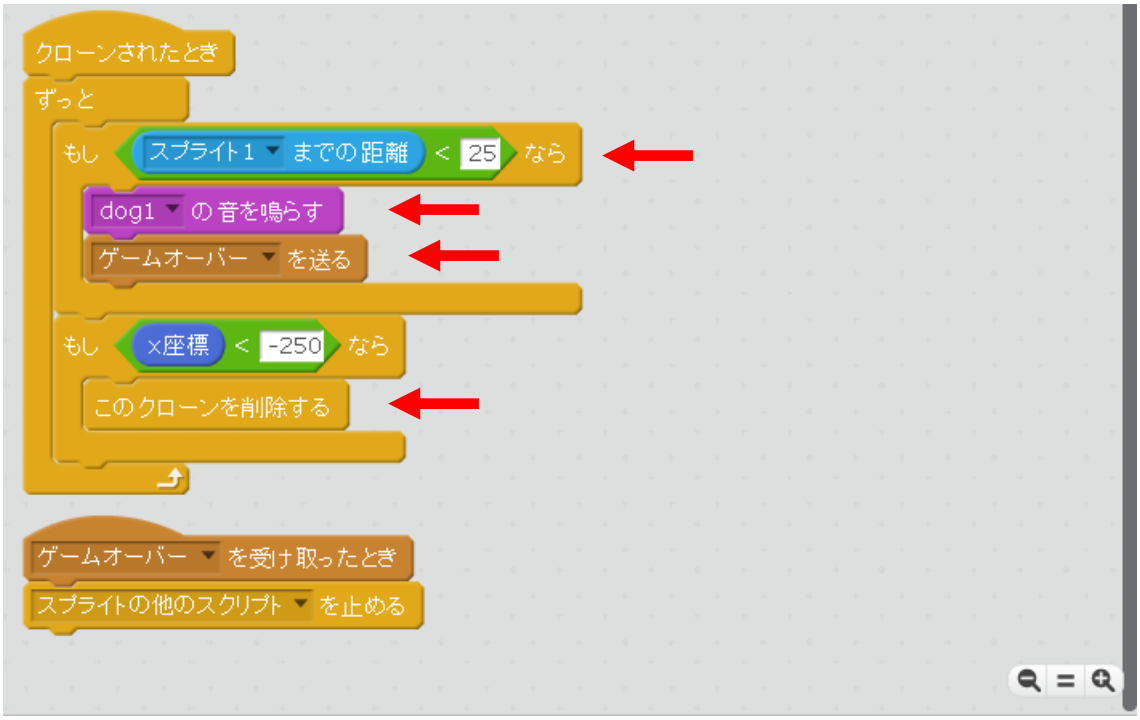




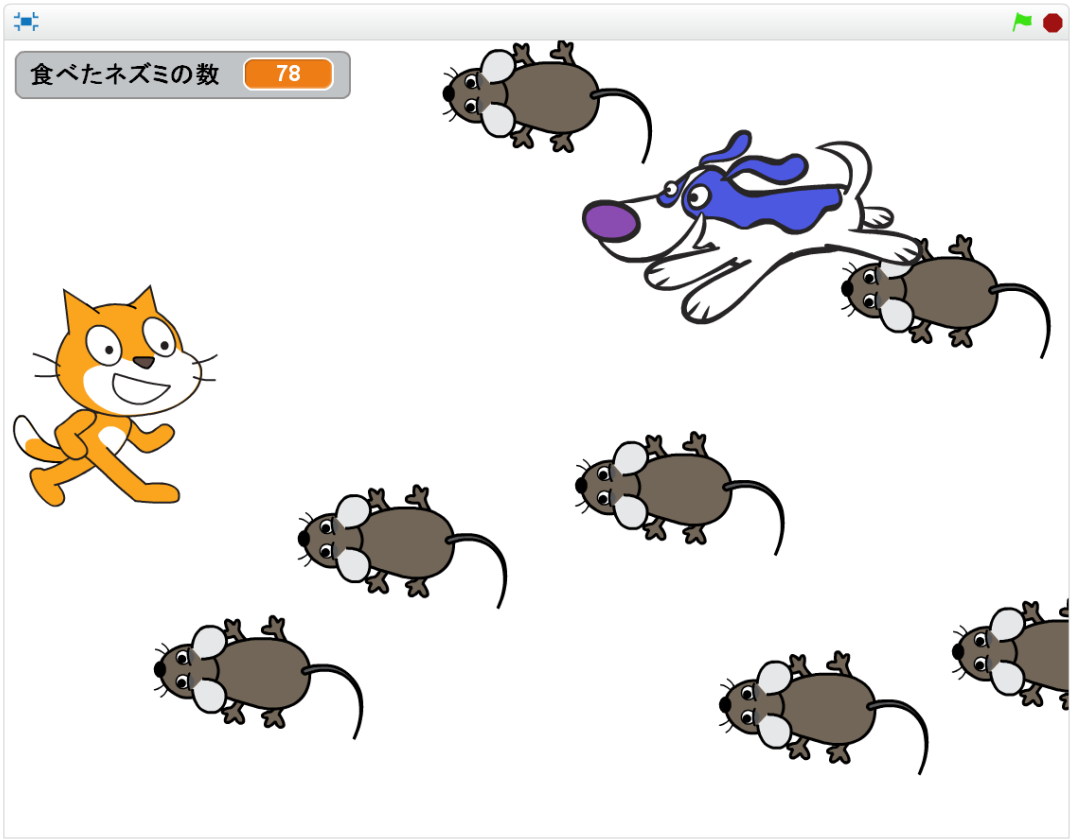
スクリプトの内容は、ネズミと基本的には同じですが、赤矢印で示した場所だけ変更します。



次のページに続く



実行例



いかがですか。少し工夫するだけで高度でおもしろいゲームになりましたね。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。