

# スクラッチを使って ネズミのえさとリゲームを つくろう



<http://www.mathtext.info/>



数楽

## プログラミングとは

プログラミングとは、決まったことをコンピュータやロボットにしてもらうために書く、マニュアル(作業手順書)のことです。

身近なもので例をあげると、自動販売機がそうです。お金を入れて、ボタンを押すと希望の飲み物が出てくる。おつりが必要であれば、適切な金額のおつりが出てきます。そのようにプログラムされていなければ、だれかほかの人が代わりにそれをしなければなりません。

コンピュータやロボットにこんなことしてほしいなと思ったら、そのようにしてもらうように、マニュアルをつくっていきます。そうすることで、コンピュータやロボットはそのマニュアル通り動くことになります。このようなマニュアルをつくることをプログラミングといいます。

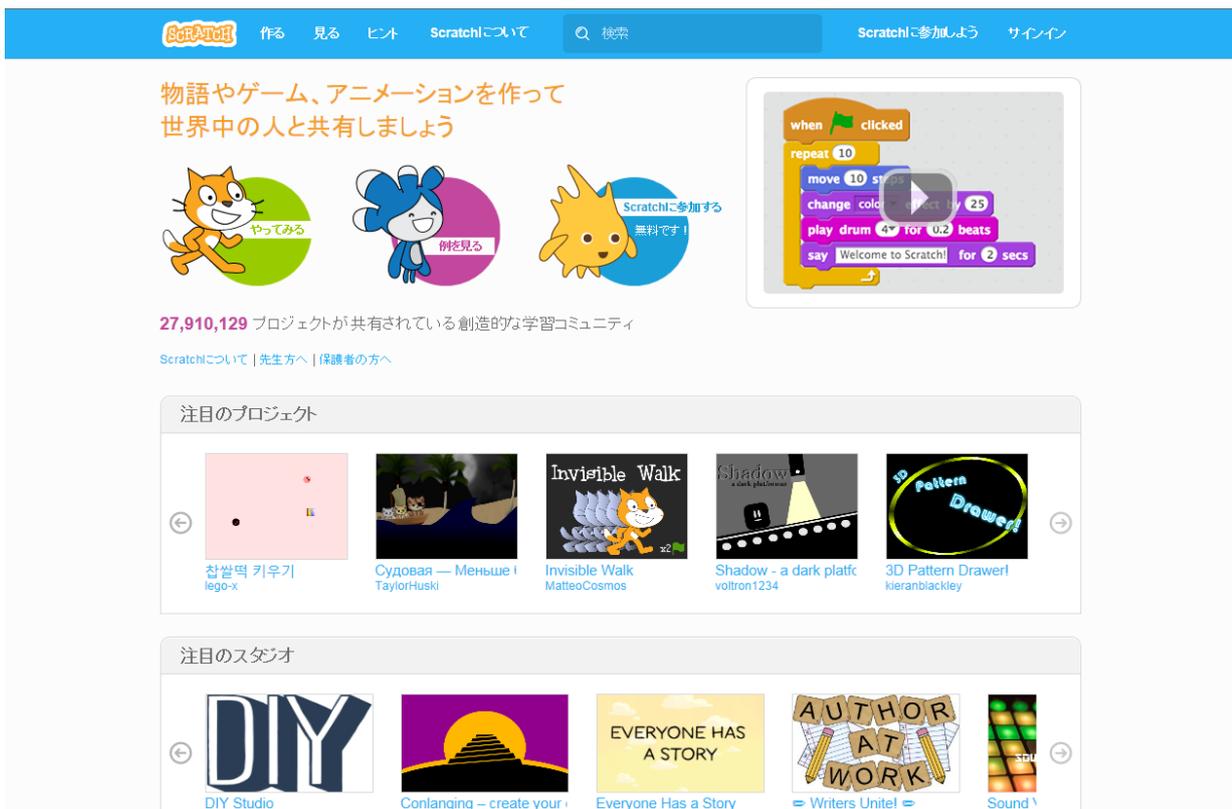
プログラミングができれば、人の役にものすごく役に立ちます。切符の券売機、インターネットオークション、スマホアプリ、Googleなどの検索画面など身近なものがほとんど、プログラミングで、できています。

## プログラムが動く仕組み

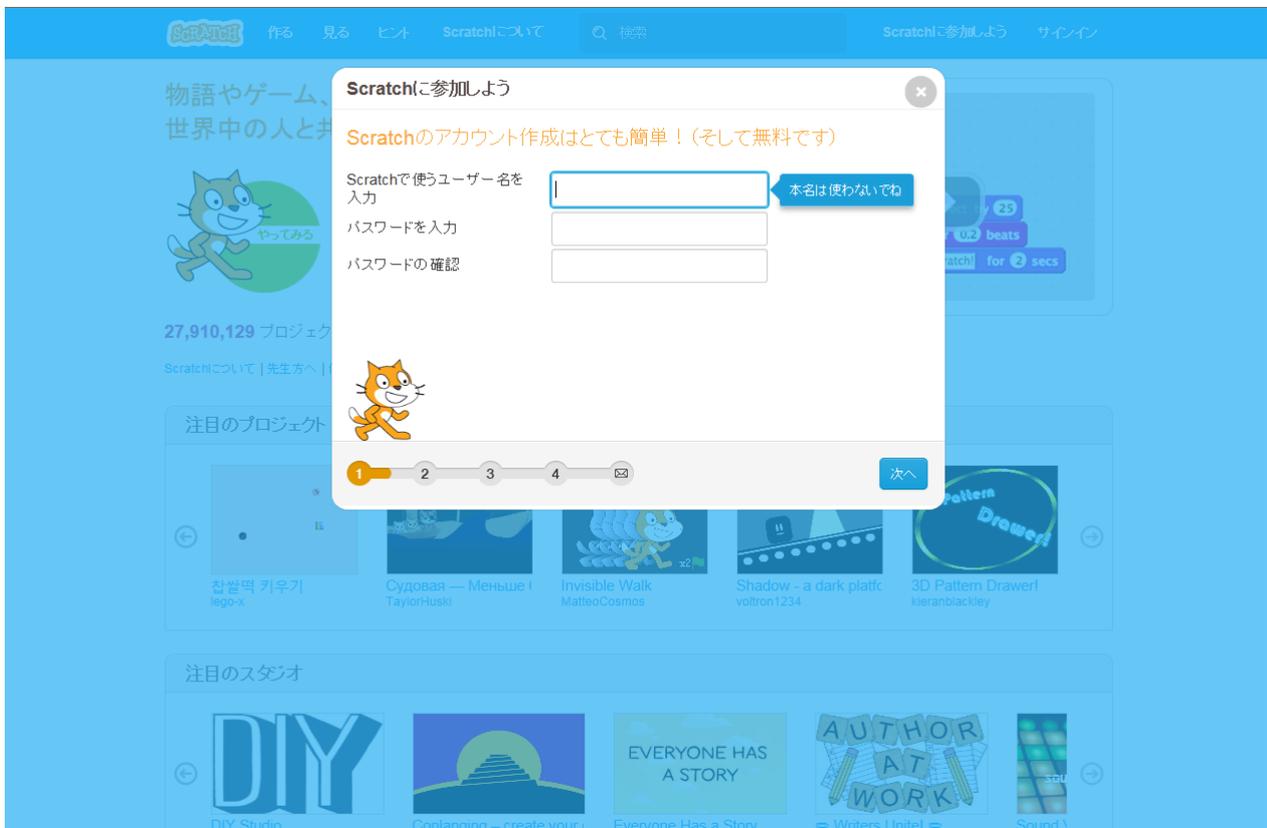
コンピュータなどの電子機械は信号があるかないかで動きます。専門的には2進数というもので動くことになります。どのように動くのか決めたのがマニュアルですが、コンピュータにはそのマニュアルは読めません。ですから、つくったマニュアルをコンピュータが読めるように電気信号に変換します。それがソフトウェアというものです。マニュアルをソフトウェアで電気信号に変えることで、コンピュータの頭脳であるCPUを動かして、期待する動作を得ることができます。

\*プログラムが正確に書かれていないと、コンピュータは正常に動きません。場合によっては、ピクリともしません。そのあたりをわかっておかないと、プログラミングが楽しくなくなります。今回取り上げるスクラッチには、たくさんのサンプルプログラミングの本が出ておりますから、やってみたいと思った人は、本屋さんでご購入ください。

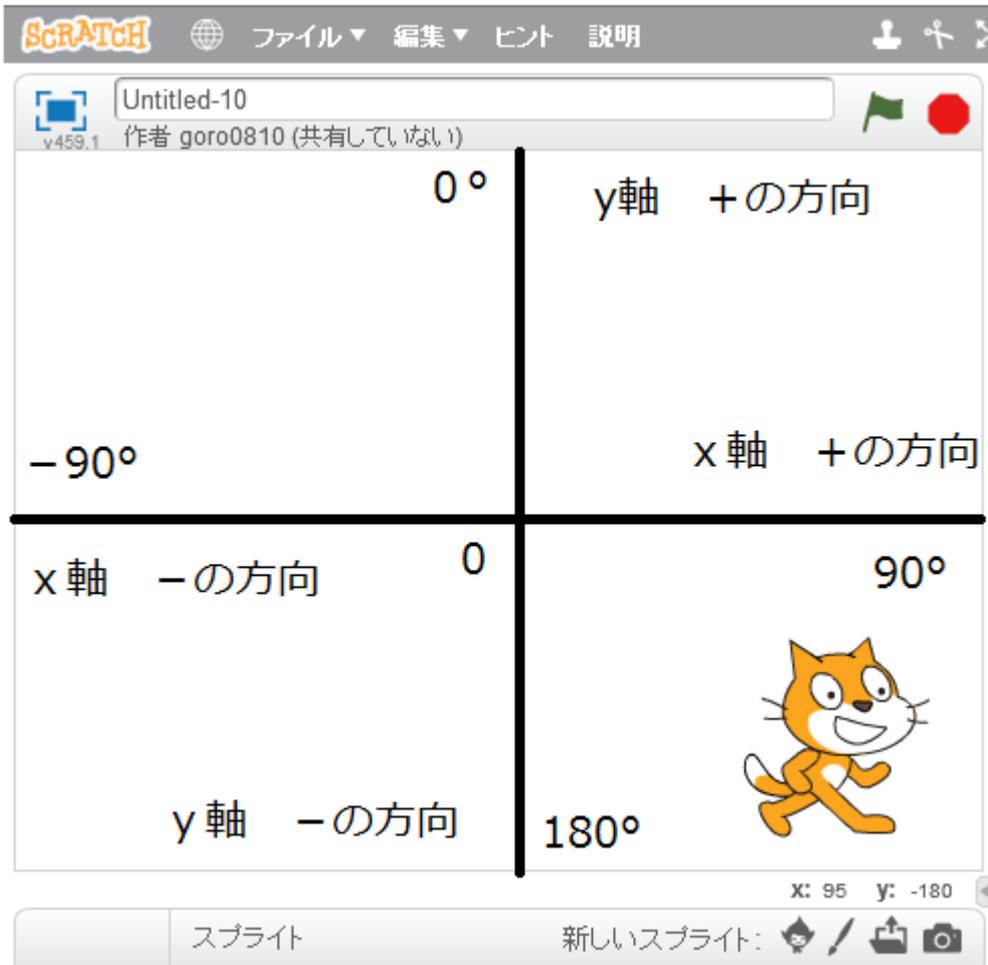
スクラッチを始めるには、スクラッチ MIT と検索をかけて、以下の画面に行きます。



サインインから、初期設定を行います。ただし、保護者の方の許可は得ましょう。各家にはそれぞれ家のルールがありますからね。



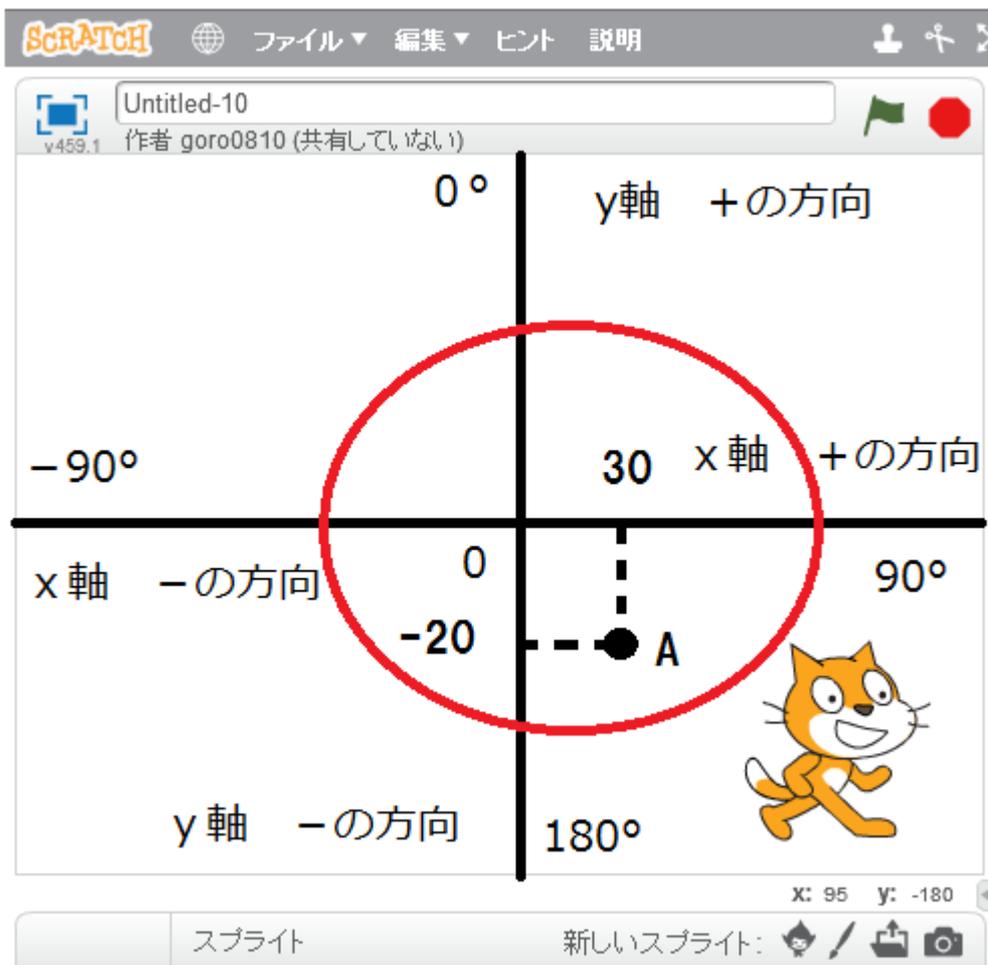
スクラッチでプログラミングしていくときは、画面が図のように設定されていますので、注意してつくっていきましょう。



画面の中心が0というところです。0から横に伸びた線をx軸、0から縦に伸びた線をy軸といいます。

0はxもyも0のところなので、(x、y)の順番に(0、0)というように書きます。これを座標といいます。

例えば下の図で、Aはxが30、yが-20のところにありますから、Aの位置を(30、-20)という座標で表します。キャラクターをAに表示させれば、xが30、yが-20のところにキャラクターを表示するプログラムを書けば、表示することができます。キャラクターなどをコントロールするために、座標は必要になります。



キャラクターや背景を設定する場合に使うボタンを○で囲っておきます。



 このボタンは自分で作ったものをアップロードするときに使います。

 このボタンは自分でキャラクターや背景をつくっていくときに使います。

まずゲームをつくるには設計図が必要です。ちょうど家を建てるのに設計図が必要なのに同じように。

ですから、今回はネズミがえさを取ったら、ゲームクリアというもので、つくっていきましょう。

ただし、えさを取りに行くときに障害物(壁)に当たれば、ゲームオーバーということをつくっていきましょう。

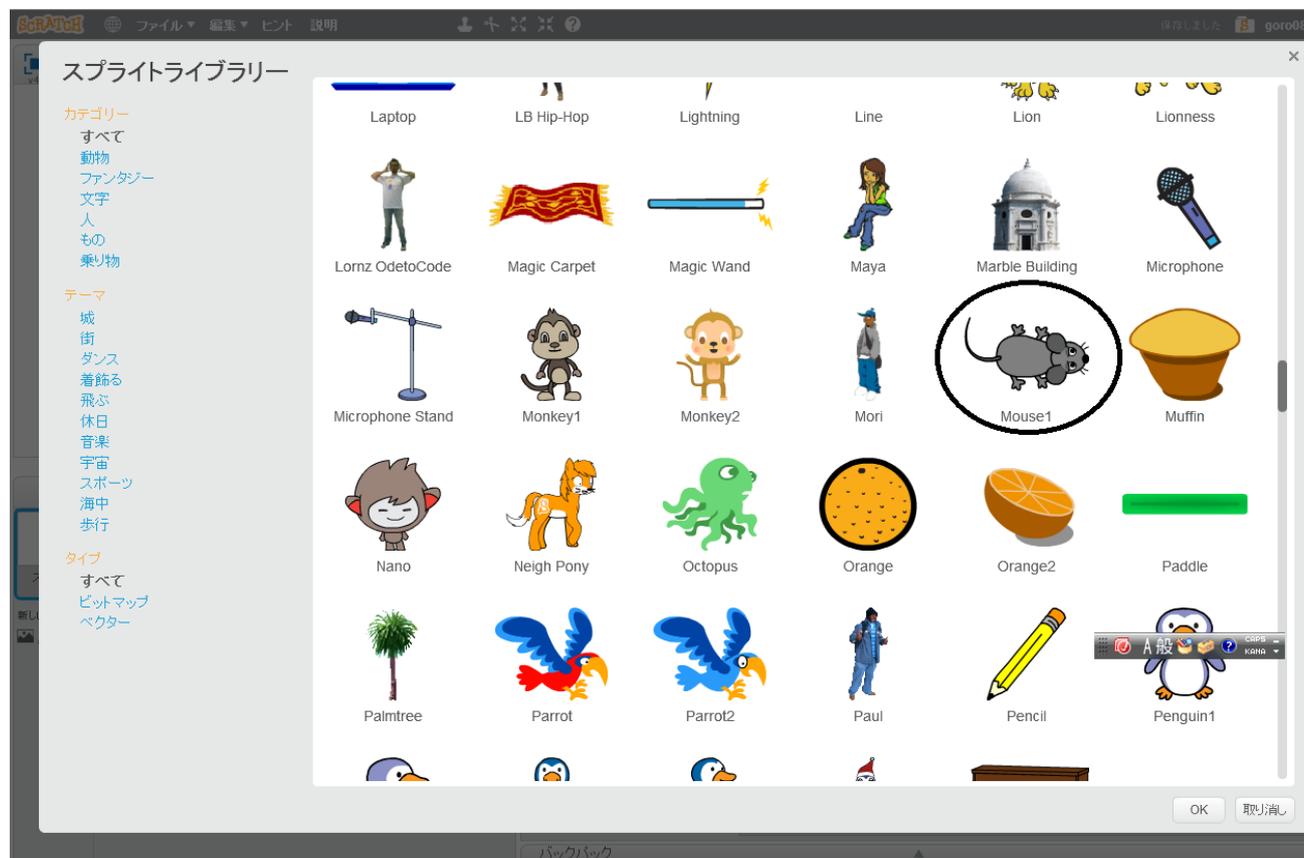
途中で設計図(ゲームの内容)を変えてもいいですし、途中で失敗してもいいです。とにかく上達するには失敗をくり返し、その分、上達していくことです。失敗してすることが、プログラミングでは許されています。



完成イメージをつかんでおきましょう。

ではつくっていきましょう。

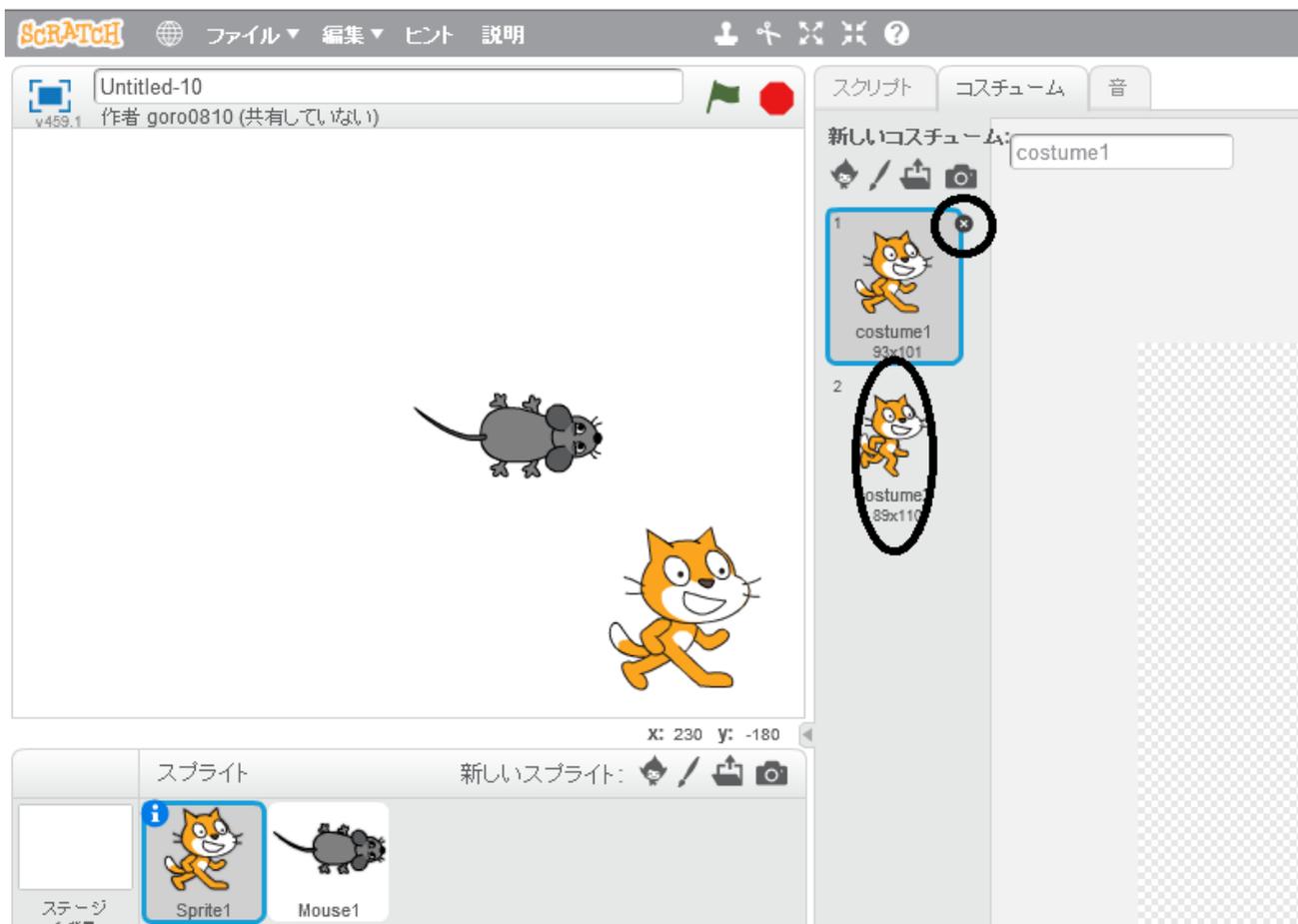
新しいキャラクター(スプライト)を追加のボタンを押してネズミを取り込みます。



クリックするとネズミが表示されます。

猫のキャラクターはいらないので、削除しましょう。

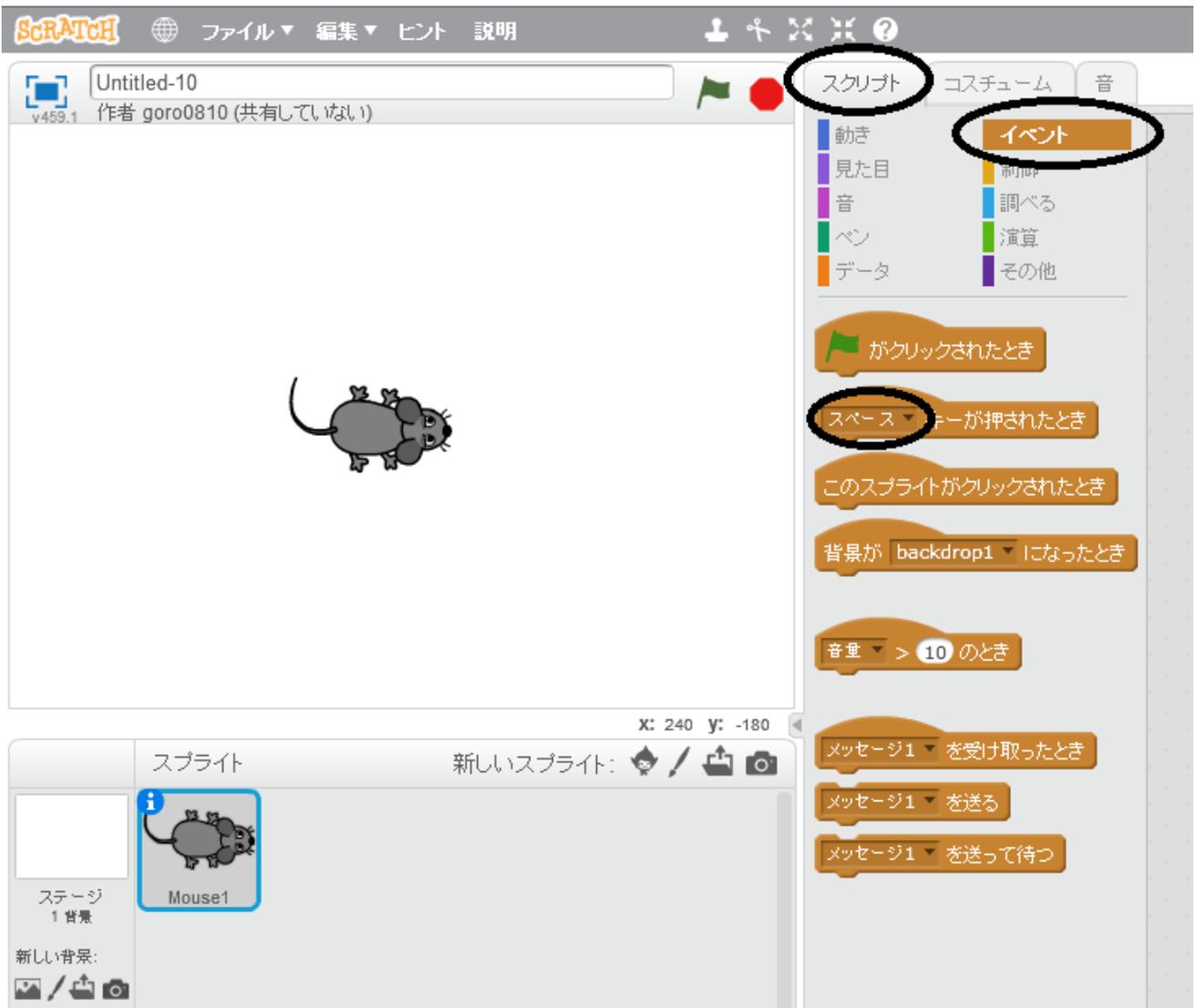
猫をクリックして、コスチュームを選ぶとこのような画面になるから○で囲ってある×印で削除していきましょう。消えないときは、猫の上で右クリック→削除でも消せることができます。



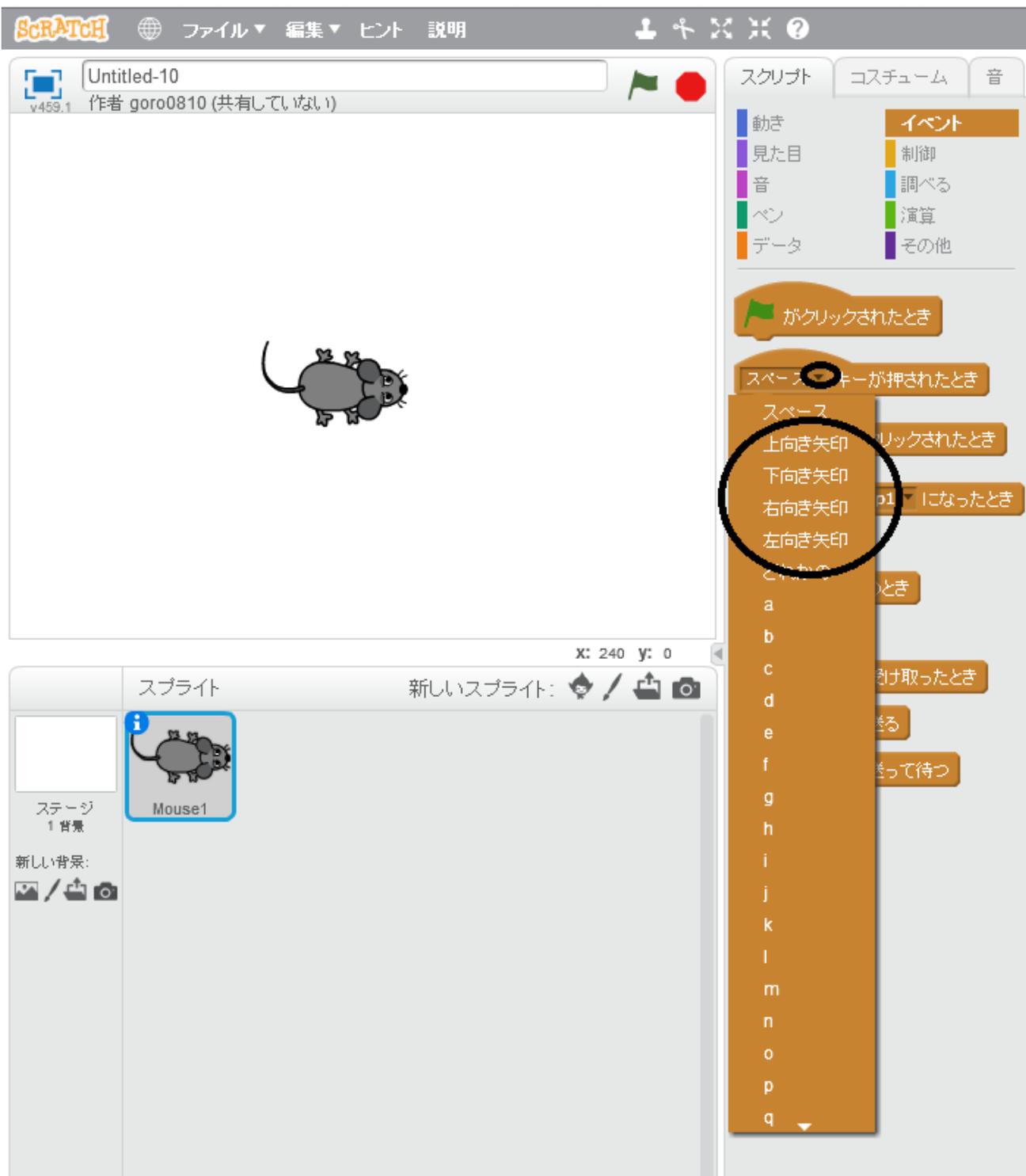
猫を消したら、ネズミのキャラクターを選択すると2つキャラクターがあるので、同じようにコスチュームを選択し、ネズミのキャラクターも1つ削除しましょう。場合によっては初めからネズミが1つのこともありますので、1つの場合は削除しなくてよいです。



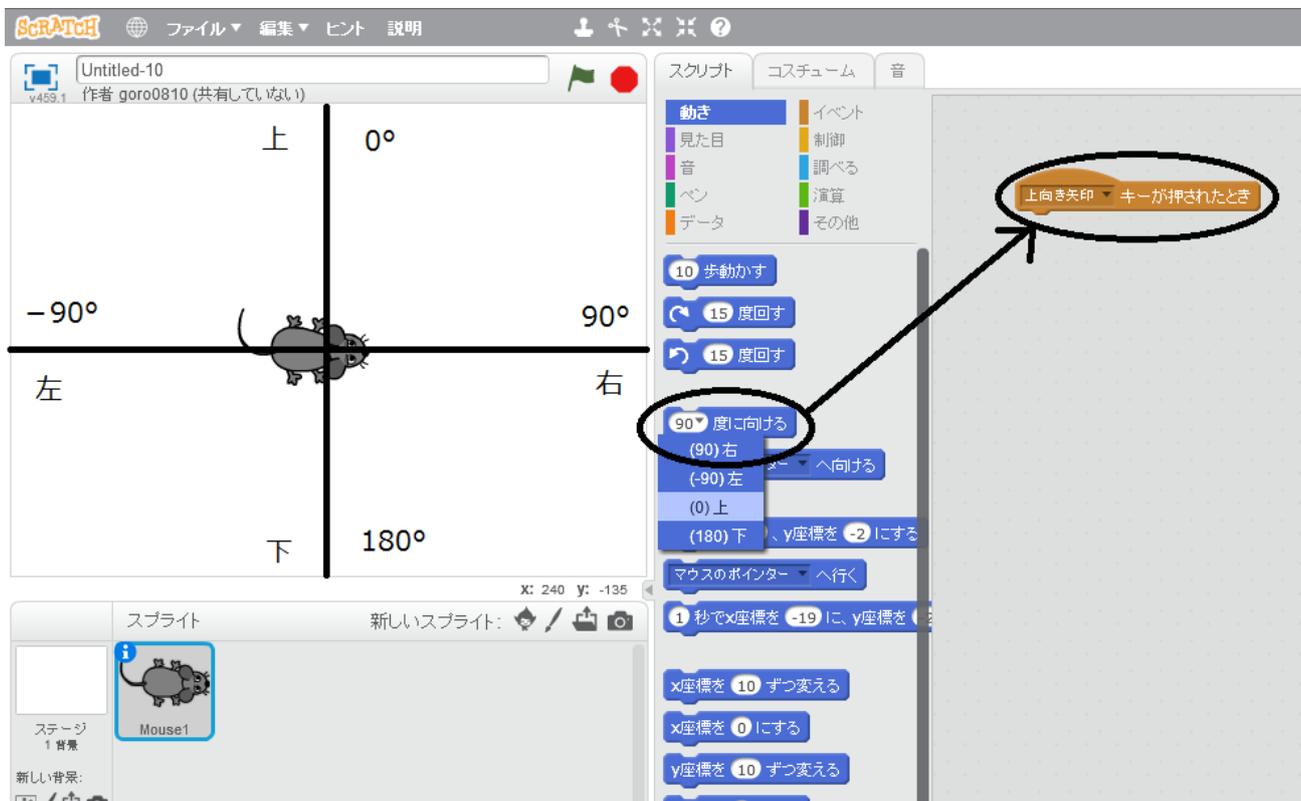
さて、いよいよネズミに魂を込めていきます。  
ネズミをクリックして、スクリプトをクリックしましょう。  
これで、ネズミに対するプログラミングができるようになります。  
スクリプトにそのプログラミングを書いていきます。  
まず、キーボードから命令をいれて上下左右に動かしたいと思います。  
そのためにイベントを選択します。  
そこに上から2番目のコマンドの矢印を押してみてください。



このように上向き矢印、下向き矢印、右向き矢印、左向き矢印などが出てきます。  
今回はこれらの矢印を押すとネズミがそれに従って動くということにしましょう。



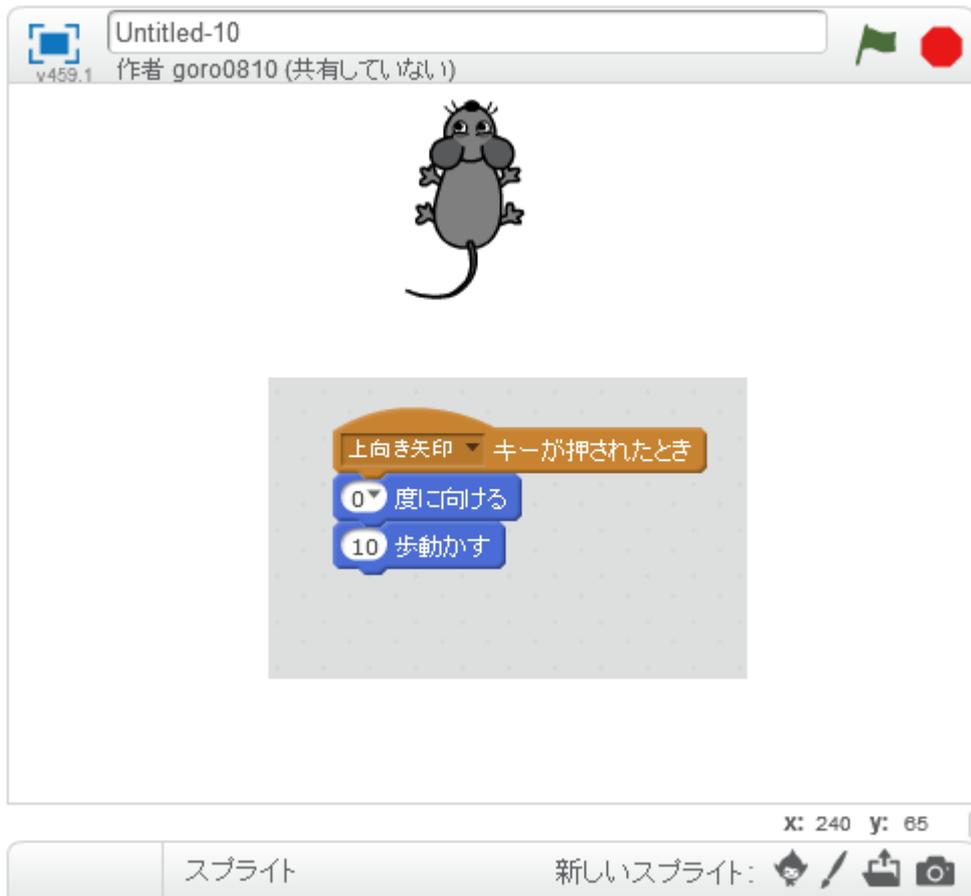
上向き矢印を押すと上に向けたい(0° にしたい)ので、  
イベントで、上向き矢印が押されたら「動き」から「(0)上度に向ける」を選択します。  
角度とキャラクターの向きとの関係は以下のようになっております。



上に向けたらその方向に動かしたいですよね？

キャラクターを向けた方向に動かすためにはスクリプトの「動き」から 10 歩動かすを選択し、先ほどの後ろにくっつけます。

これで、上向き矢印を押すとネズミが動くようになりました。



同じように下向き、右向き、左向きでもつくりたいので、先ほどのスクリプトの上で右クリックをすると、複製っていうのが出てきますから、複製を 3 つ(下左右の分)つくっておきましょう。



で複製をそれぞれ、下左右の設定をします。



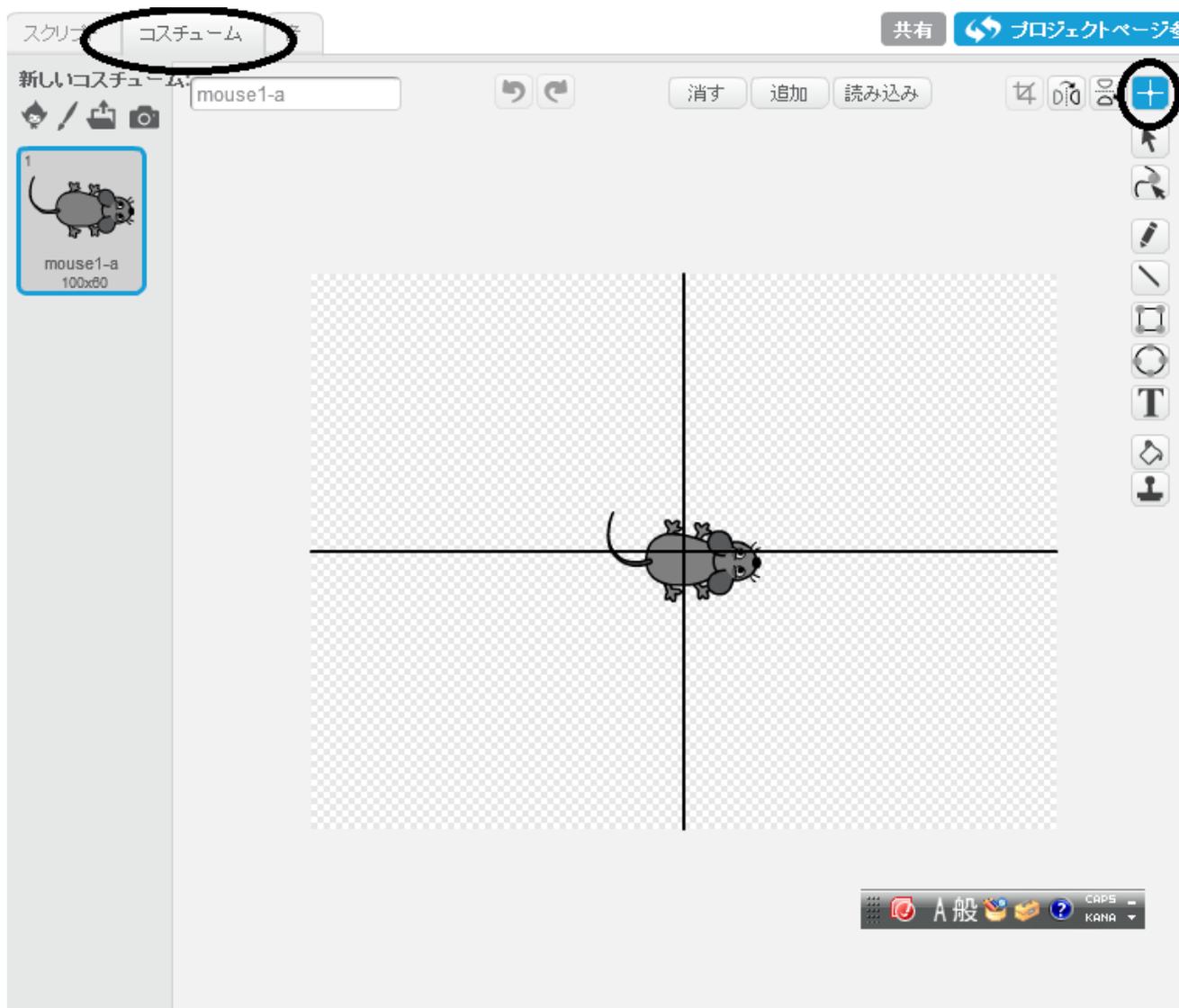
これでネズミが上下左右にあなたの思うように動き回ります。

ただし、動かしてみると少しおかしいことに気がきます。

ネズミがきちんとくるくる回っていないのです。

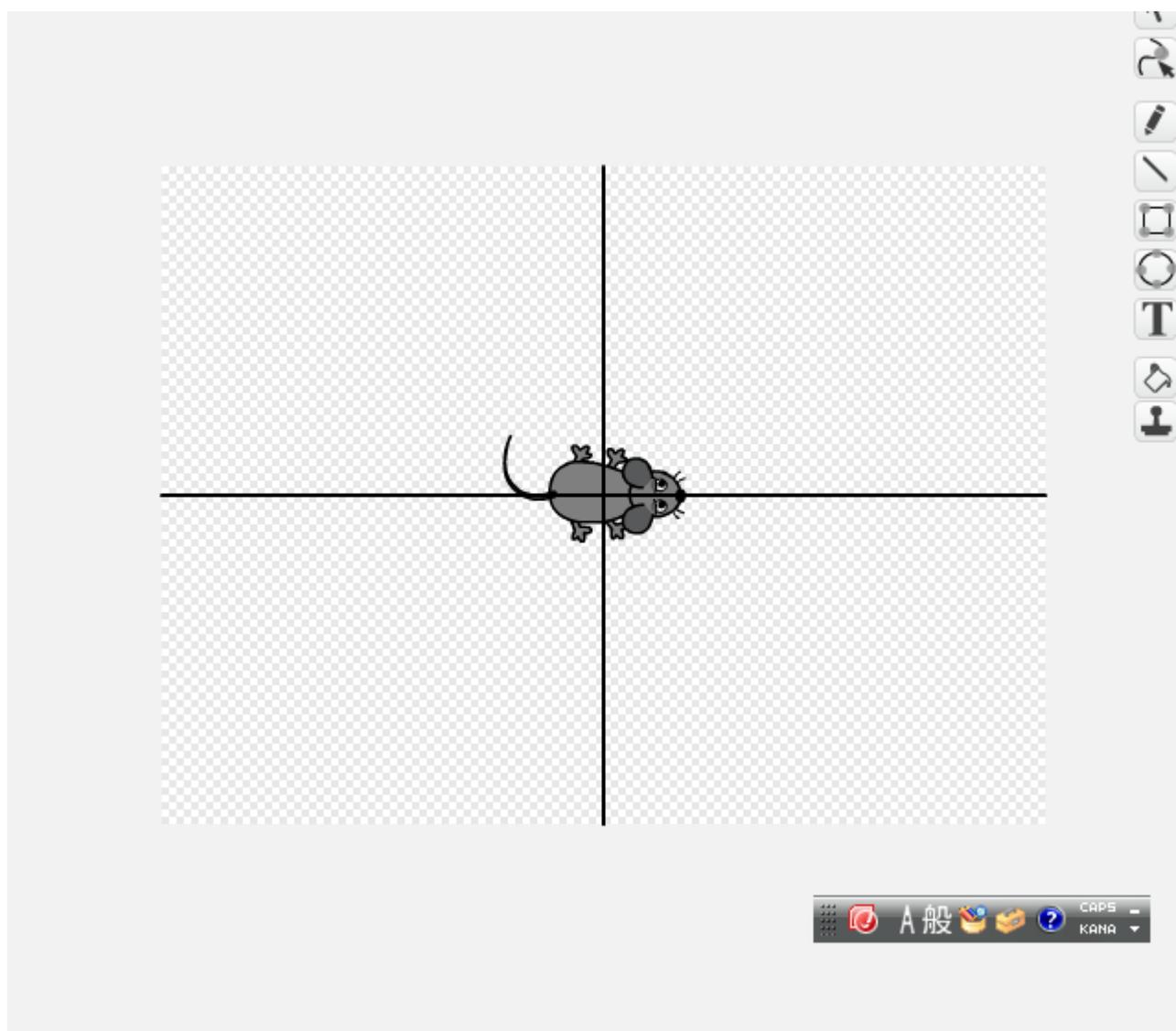
きちんと回っているように見えますが、ネズミのキャラクターの中心の設定がずれているため、きちんと左右上下対称には反転していないのです。

ネズミのコスチュームを選択して右上の+を押してみてください。  
下のように中心がずれています。

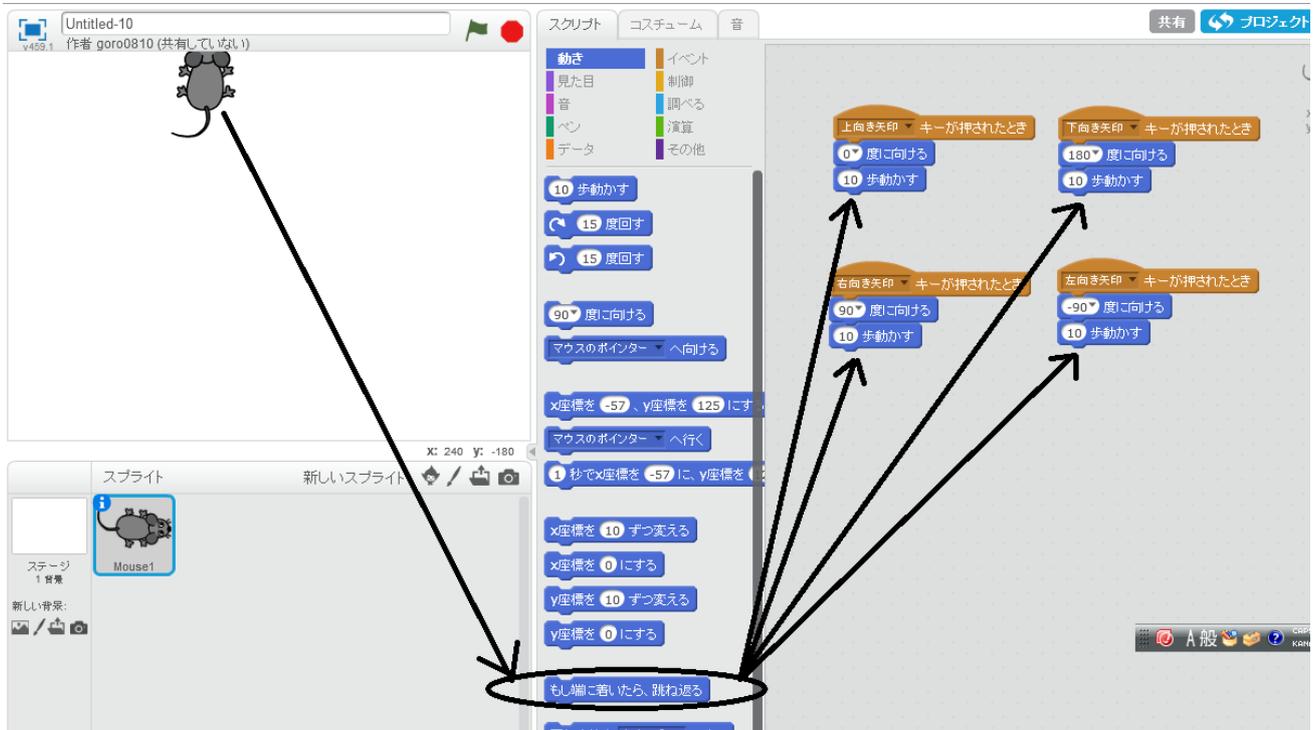


この中心のずれを直しましょう。

このように+を中心に持ってきて、ネズミの動きをスムーズにすることができます。

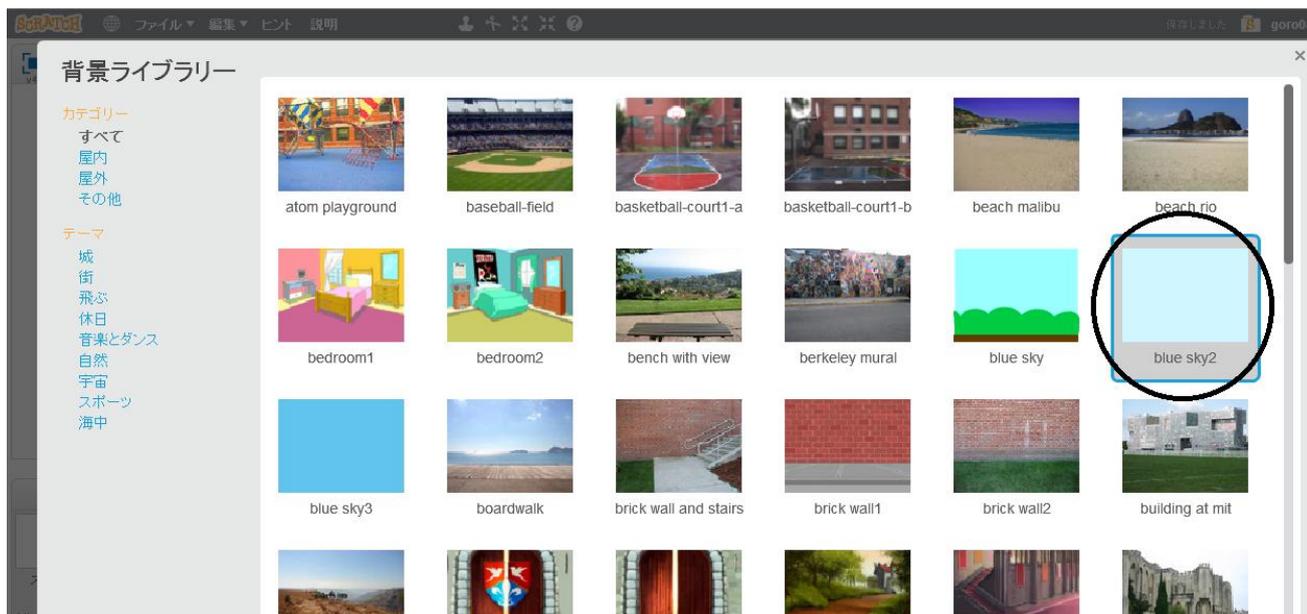


でもネズミを動かしてみると、壁にあたったら画面からはみ出してしまう。  
そういうのが気になったら、スクリプトを選んで「動き」から、「もし端に着いたら、跳ね返る」を選択します。そして上下左右のスクリプトの最後に追加していきます。これで、端に着いたらはみ出ることはありません。

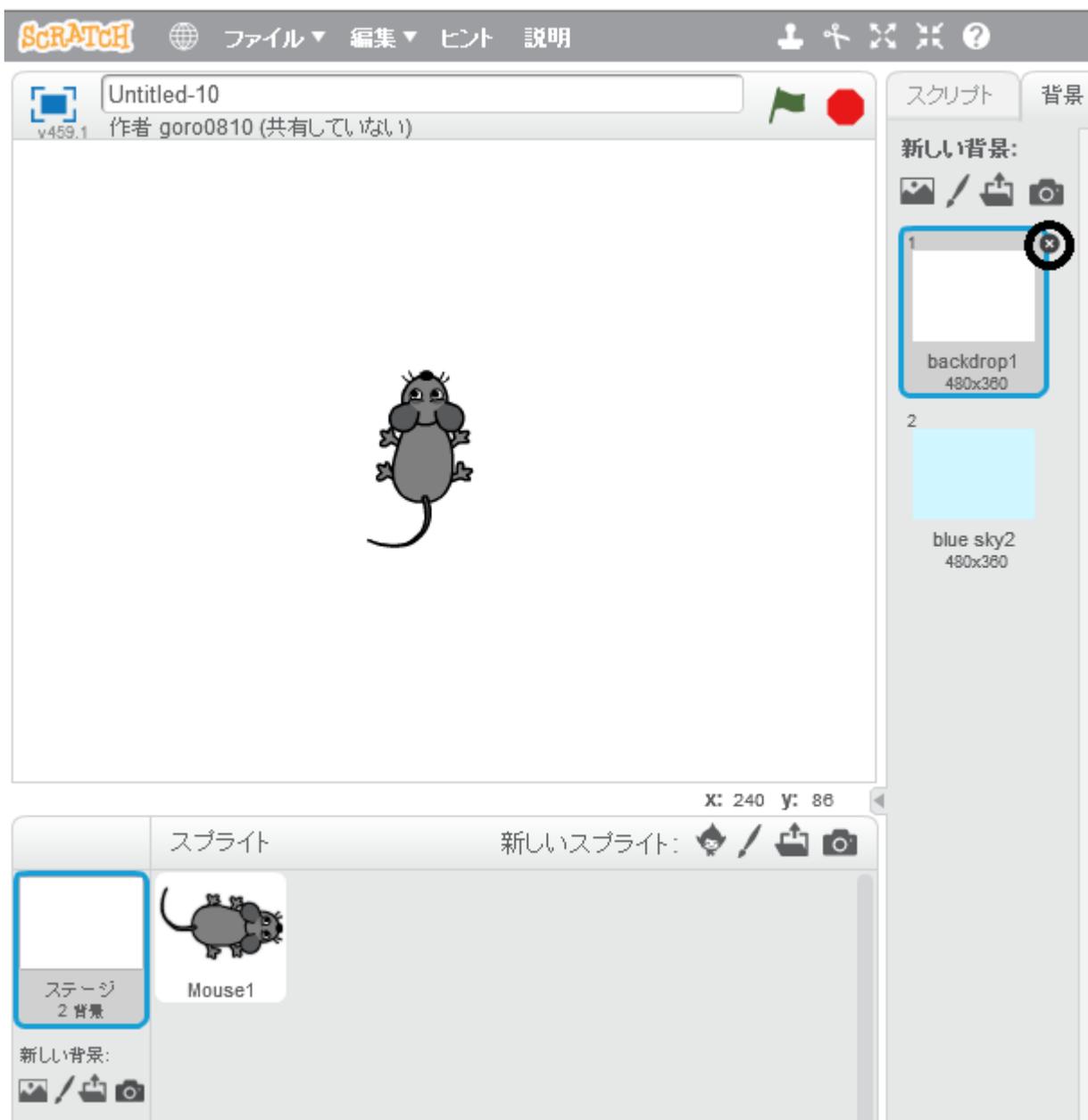


今回はこの命令は入れないでおきます。

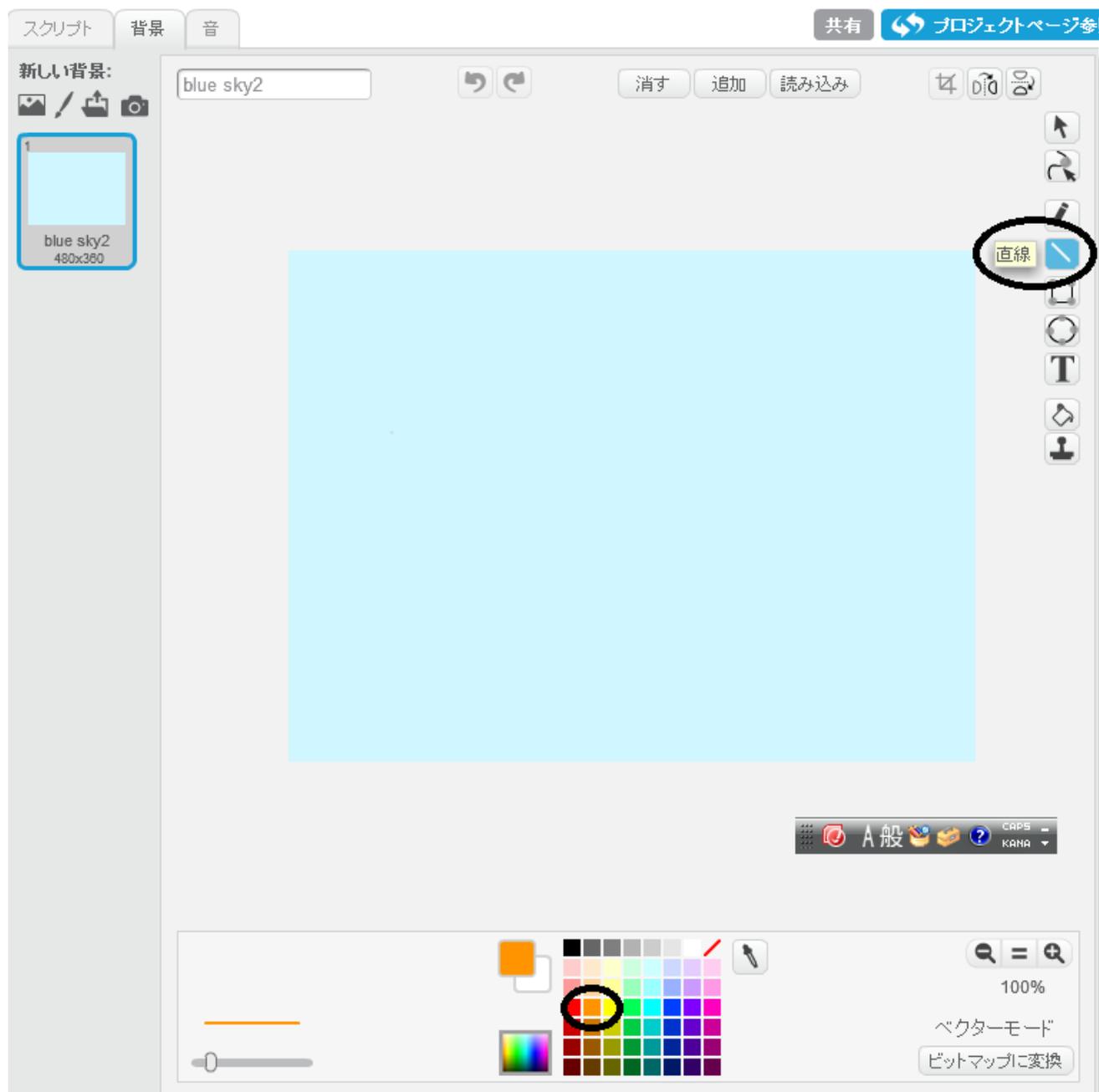
続いてネズミが走るコースの作成に取りかかります。  
新しい背景を選択して **blue sky2** を選びましょう。



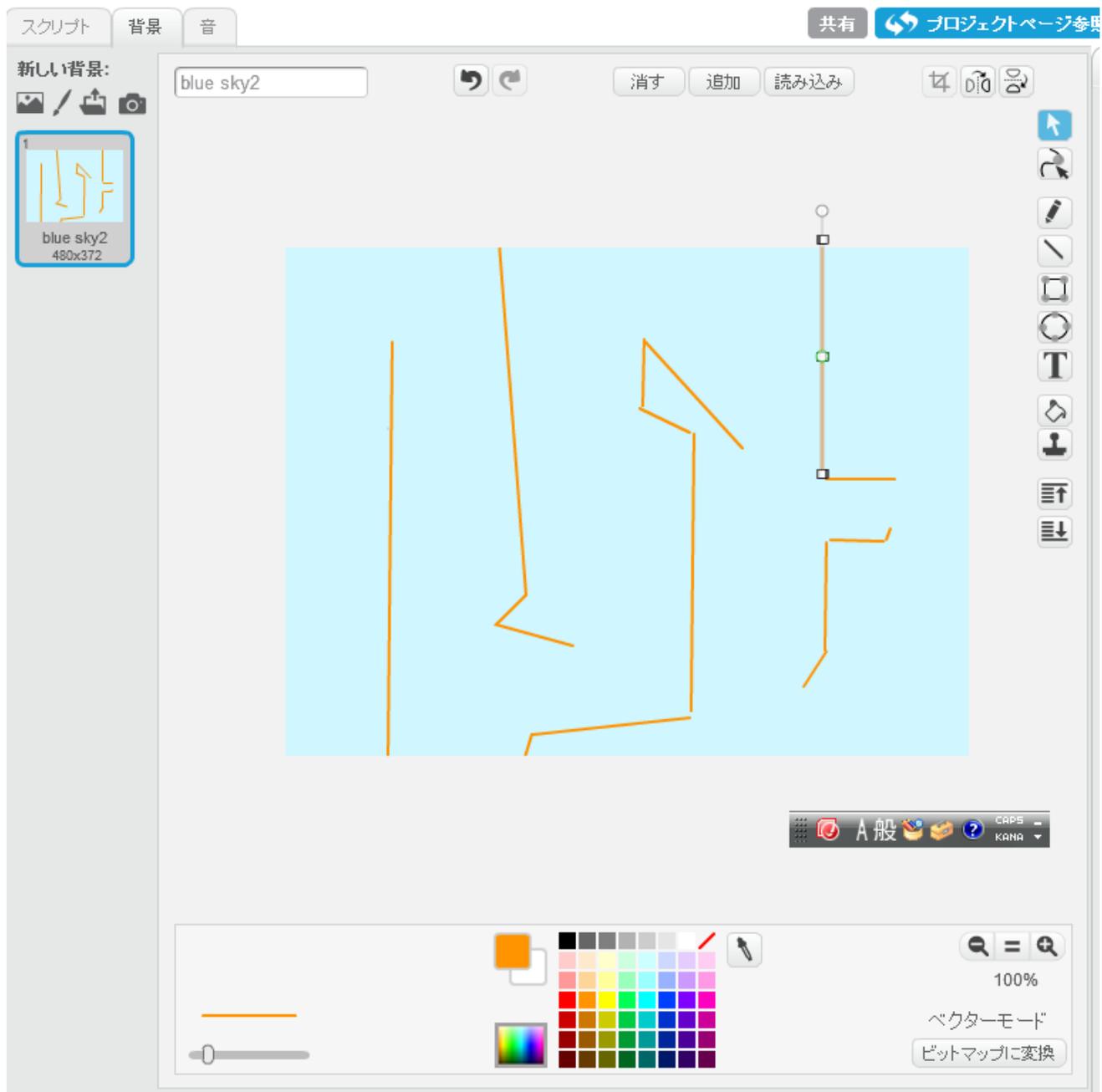
選んだら白(もと)の背景はいらないので×を押して削除しましょう。



右にある直線を使ってコースを書いていきます。



色は何色でもいいですよ。



間違ったらクリックして削除していきましょう。

迷路みたいなコースをつくりましょう。どこがゴールになるか自分で決めてつくりましょう。

The image shows the Scratch IDE interface. The main stage area displays a maze made of orange lines on a light blue background. A mouse sprite is positioned at the start of the maze. The right-hand side shows the 'Scripts' palette with various blocks. A callout bubble with a black border and white background contains the text: 「大きさを指定してクリックするとサイズが変わる」. This bubble points to a 'Change size to 60%' block in the script area, which is also circled in black. The 'Sprites' palette on the left shows the mouse sprite selected. The top of the window displays the Scratch logo and menu options like 'ファイル', '編集', 'ヒント', and '説明'.

ネズミの大きさが大きくてコースに合わないときは、見た目から大きさを変更できるよ。

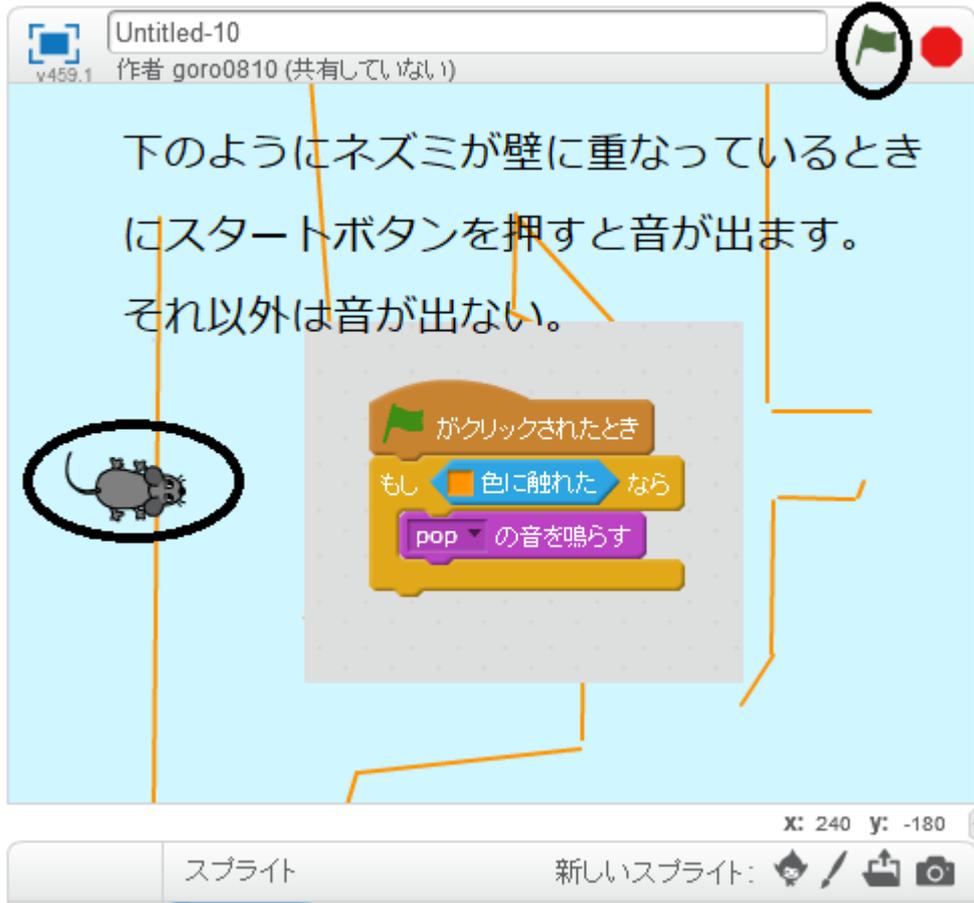
ゲーム画面の右上にある緑の旗はゲームのスタートボタン、赤い丸はゲームのストップボタンです。

スタートボタンが始まったら、ゲームが始まるようにしましょう。



ゲームのルールは、障害物(壁)に当たったら、ゲームオーバーというものです。ですからゲームを開始(緑の旗を押す)すると、壁に当たったとき、当たったことがわかるように音を出すというプログラムを組みます。

壁の色はオレンジなので、オレンジ色に触れたら POP という音が出るようにします。



しかし、これではスタートを押したときに、ネズミがオレンジ色の上にないと、音が出ません。

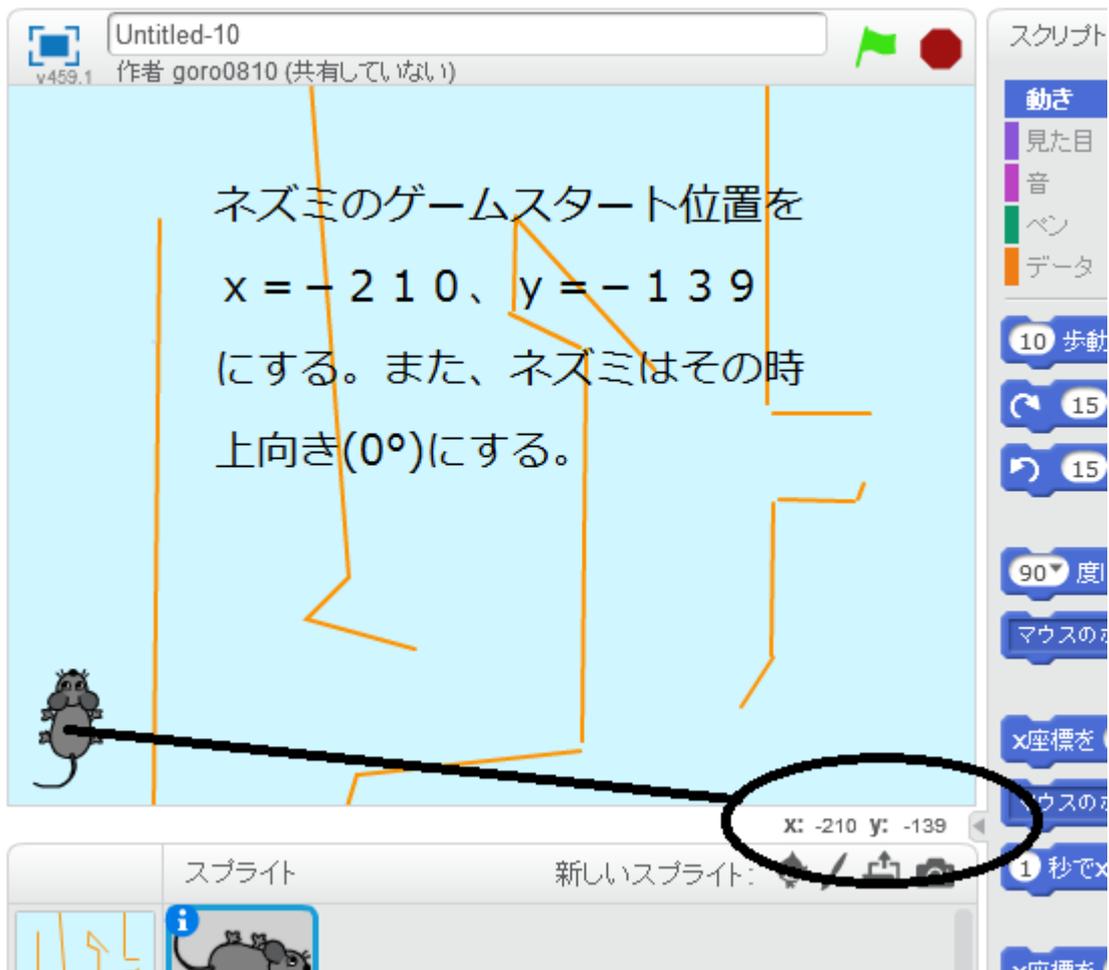
そこで、次のように改良します。

スタートボタンが押されたら、ずっと壁に当たったかどうか調べるようにします。



これで壁に触れるたびに音が出るようになりました。

次にスタートボタンが押されたら、ネズミの開始位置も決めなくてはなりませんね。このとき、マウスをネズミ近づけると、ネズミがどこら辺にいるのか、x、y の位置が右下に表示されています。はじめにも少し触れましたが、これを数学では座標と言います。



今回はスタートボタンが押されたら、ネズミは、  
 $x = -210$ 、 $y = -139$  の位置にくるようにプログラミングします。  
そして上に向けたいので、上向きの角度  $0^\circ$  もプログラミングします。  
以下がそのプログラム例です。



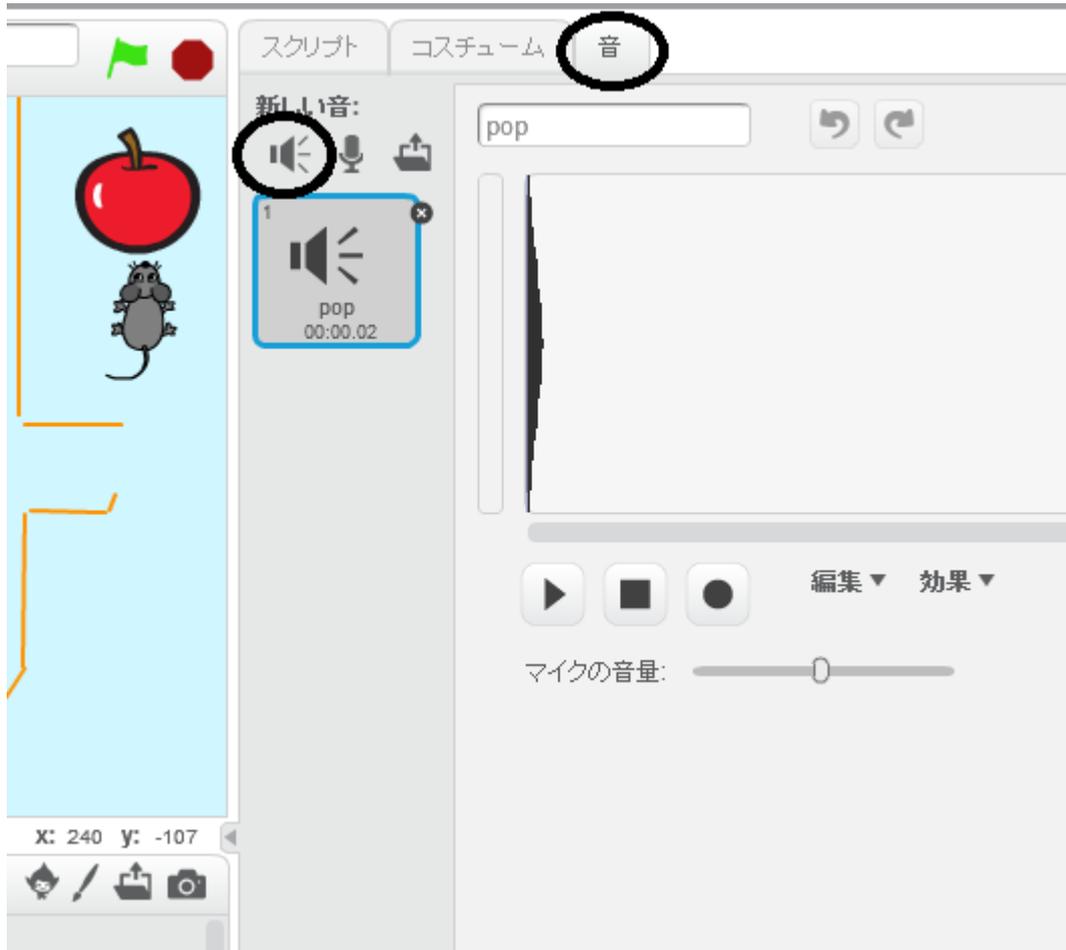
次にゴール地点に何かネズミのえさを置いておきましょう。  
新しいスプライトを追加から、リンゴを選択して、ゴール地点に置きましょう。



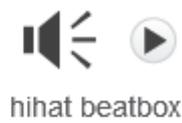
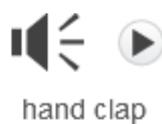
そして、壁に触れたら音が出たように、リンゴに触れたら、音が出るようにしましょう。  
以下がそのプログラム例です。



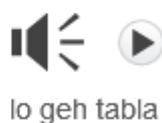
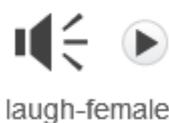
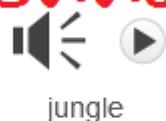
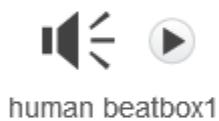
壁に当たっても、えさを取っても同じ音がするので、音を変えましょう・音を選択して、新しい音を追加しましょう。



今回は笑い声を選択してみました。▶ボタンでどんな音が聞けます。



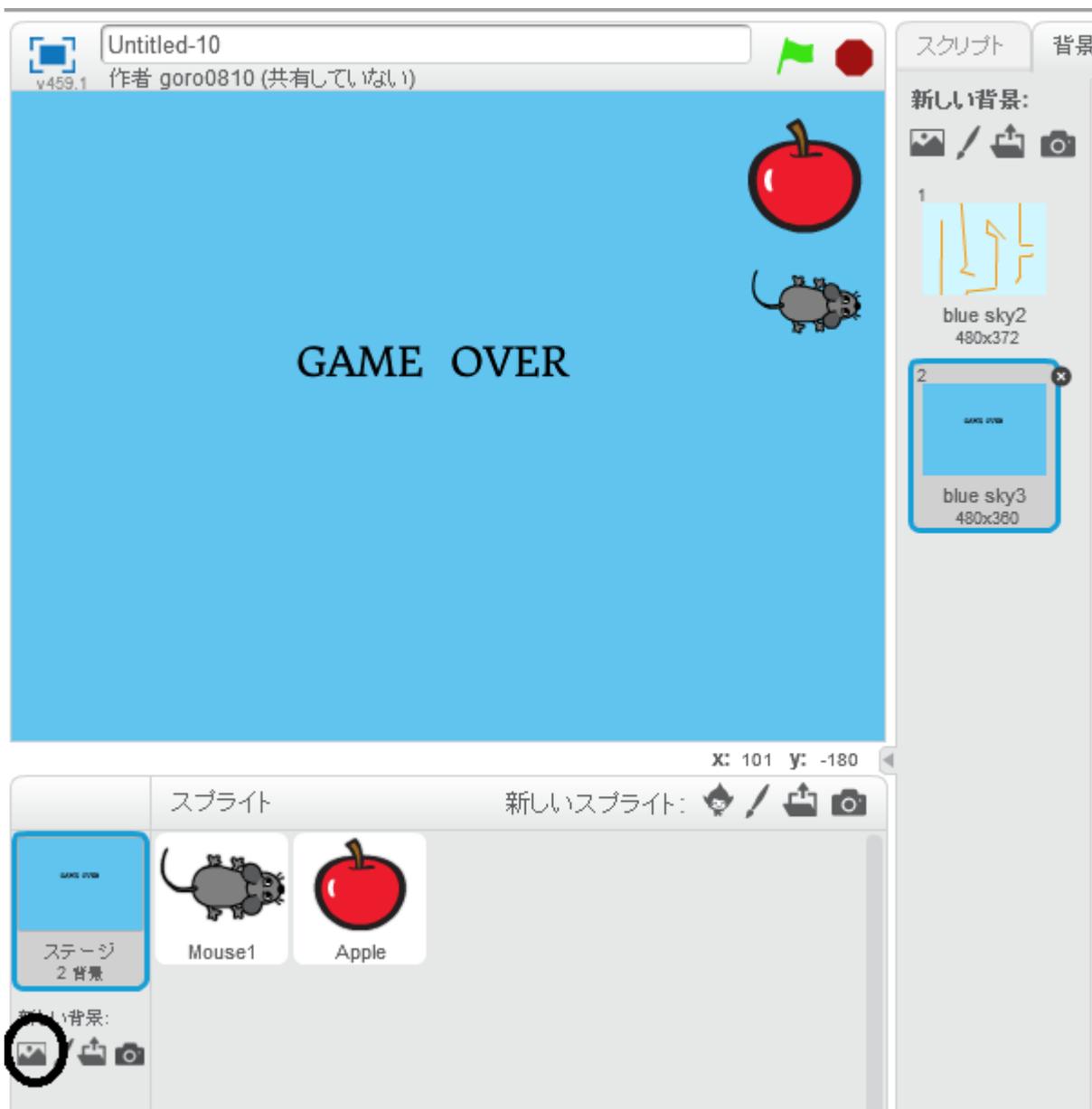
音は何でもいいよ



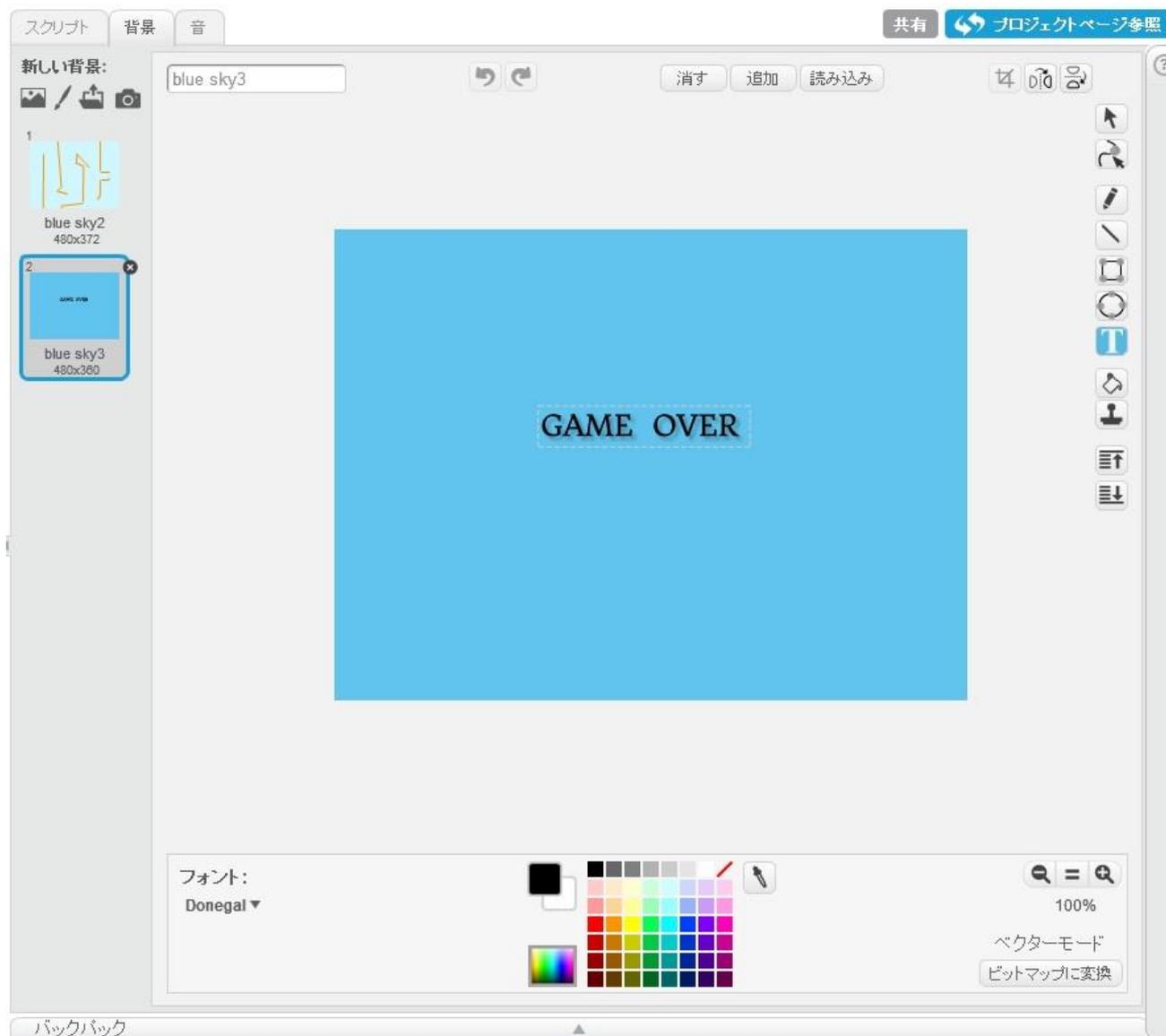
選択した音をプログラム上で選択しましょう。



次に壁に触れたら GAME OVER なので、GAME OVER の画面をつくりましょう。  
新しい背景を選択して、blue sky3 を選択し、右の T ボタン(テキストボタン)を選択し、  
GAME OVER と書きましょう。日本語は入力できないようです。



文字を使って書いてみよう。文字を書く背景を選んで、T(テキスト)を選んでかいていきます。



そして壁に触れたら、背景を blue sky3(GAME OVER)に変えればいいので、壁に触れたら、音が鳴るところに、それを追加しよう。

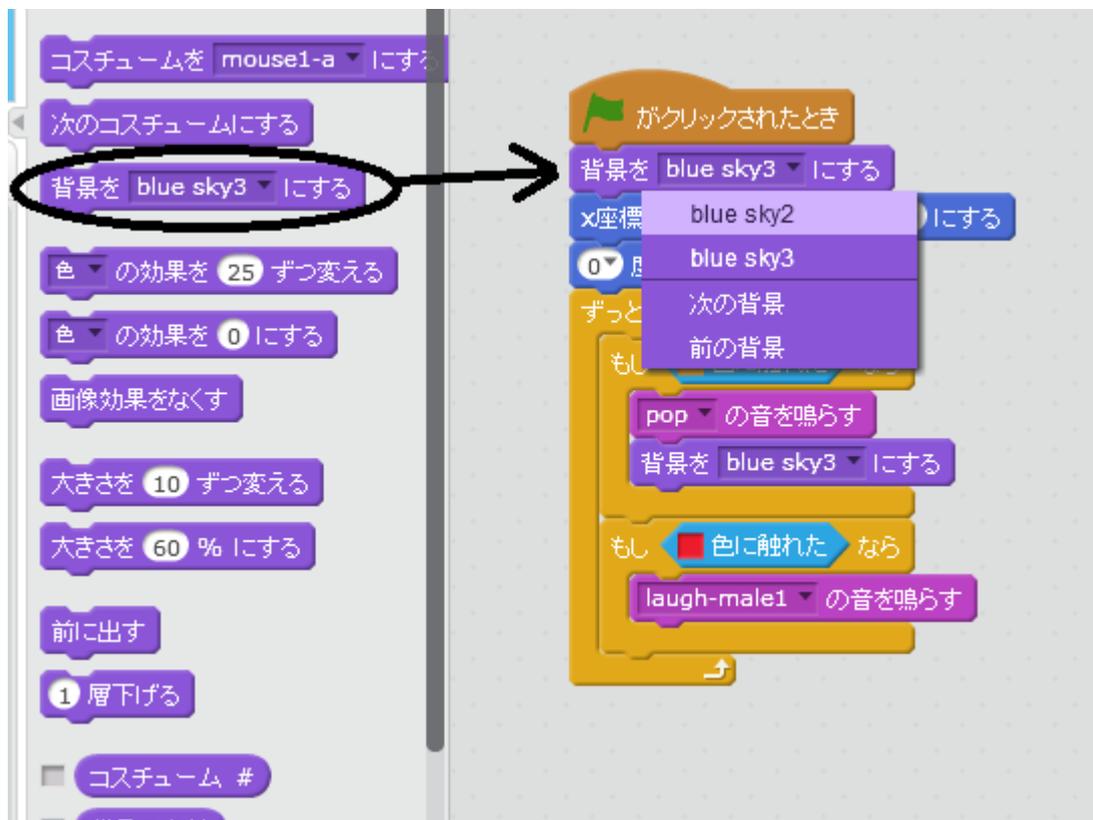
The screenshot shows a Scratch project window titled "Untitled-10" by user "goro0810". The stage has a blue sky background and the text "GAME OVER". A mouse sprite is on the left, and a red apple is on the right. A script is attached to the mouse sprite:

- When green flag clicked
- Set x coordinate to -210, y coordinate to -139
- Rotate 0 degrees
- Forever loop:
  - When touching orange color
  - Play sound pop
  - Set background to blue sky3
- Forever loop:
  - When touching red color
  - Play sound laugh-male1

さて、ここで問題。

GAME OVER になったあと、ゲームを開始しても画面が戻らないことが発覚。

これでは、ゲームが再開できないので、スタートボタンを押したら、初めの画面(blue sky2)を呼び出すことにしましょう。



これで完成です。

The image shows a Scratch project titled "Untitled-10" by user "goro0810". The stage features a maze with a mouse sprite at the entrance and an apple target at the exit. The interface includes a sprite palette with "Mouse1" and "Apple", a script area with various movement and event blocks, and a main script area with keypress and click events.

**スクリプト**

- 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

マウスのポインタを **へ** 向ける

X座標を **-210**、Y座標を **-139** に

マウスのポインタを **へ** 行く

**1** 秒でX座標を **-210** に、Y座標を

X座標を **10** ずつ減える

X座標を **0** にする

Y座標を **10** ずつ減える

Y座標を **0** にする

もし端に寄いたら、跳ね返る

回転方法を **左をのみ** にする

X座標

Y座標

バックバック

**共有** [プロジェクトページ参照](#)

**上向き矢印** キーが押されたとき

- 0** 度に向ける
- 10** 歩動かす

**下向き矢印** キーが押されたとき

- 180** 度に向ける
- 10** 歩動かす

**右向き矢印** キーが押されたとき

- 90** 度に向ける
- 10** 歩動かす

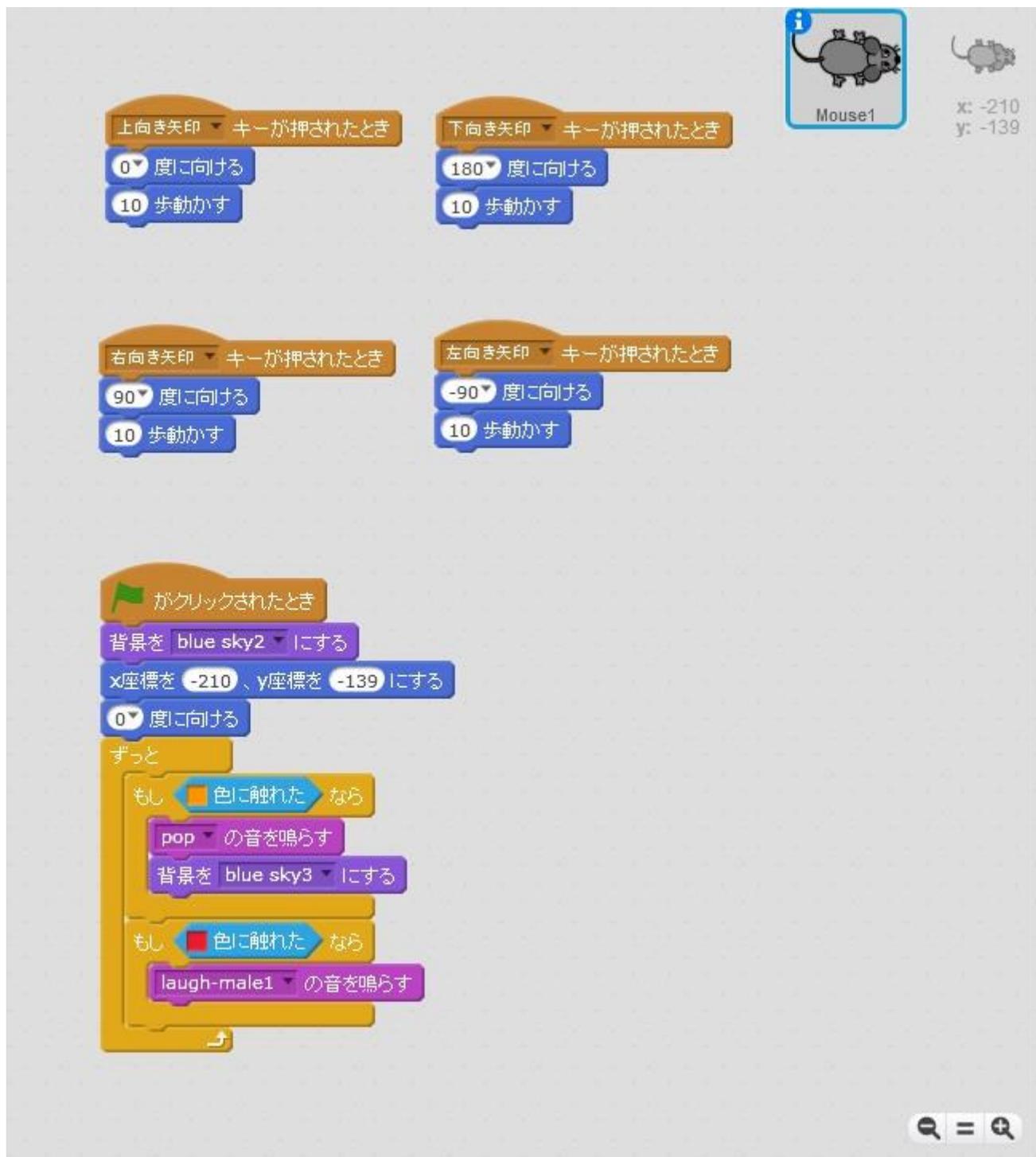
**左向き矢印** キーが押されたとき

- 90** 度に向ける
- 10** 歩動かす

**がクリックされたとき**

- 背景を **blue sky2** にする
- X座標を **-210**、Y座標を **-139** にする
- 0** 度に向ける
- ずっと
- もし **色に触れた** なら
  - pop** の音を鳴らす
  - 背景を **blue sky3** にする
- もし **色に触れた** なら
  - laugh-male1** の音を鳴らす

今回のゲームでは、ネズミだけにプログラムを組みました。  
その中身は以下のようにになりました。  
ゲームオーバーの画面をつかったので、ゲームクリアの画面もつくってみてもいいでしょう。  
これらを基本として、いろいろ広げていってください。



プログラムができれば、本当に楽しいですよ。  
いろんなものがつくれるように、勉強しましょう。  
そのためには、基本となる算数や国語をきちんと勉強しておきましょうね。