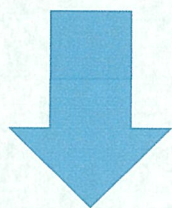


①

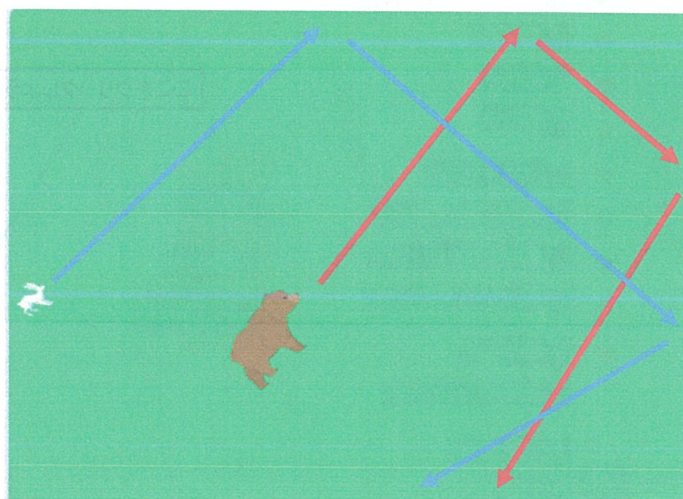
# ウサギを追う熊をつくろう

1

画面のはしっこにぶつかると、そのまま跳ね返って進むウサギと熊のプログラム

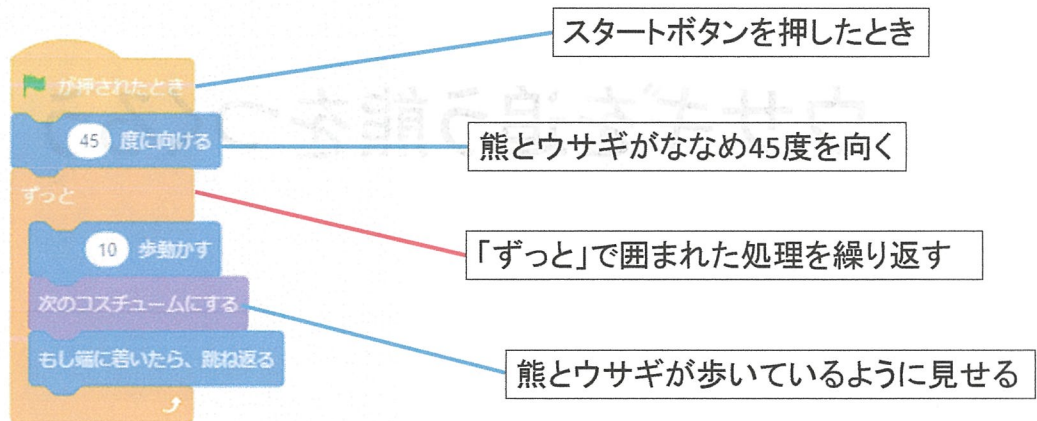


熊がウサギを追いかけるようにしたい！



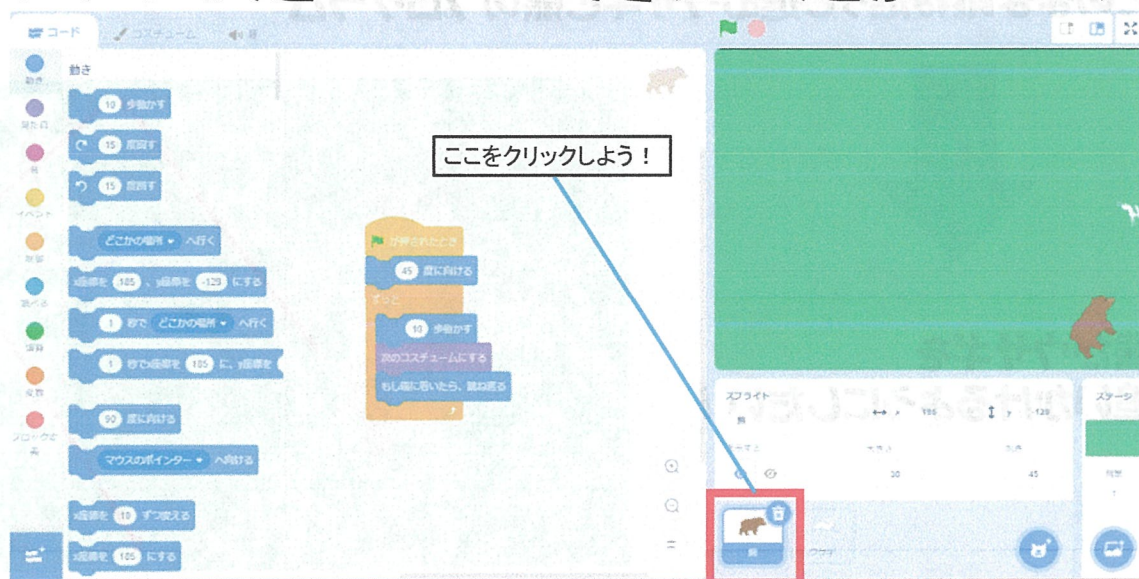
2

## まずはプログラムを見てみよう！

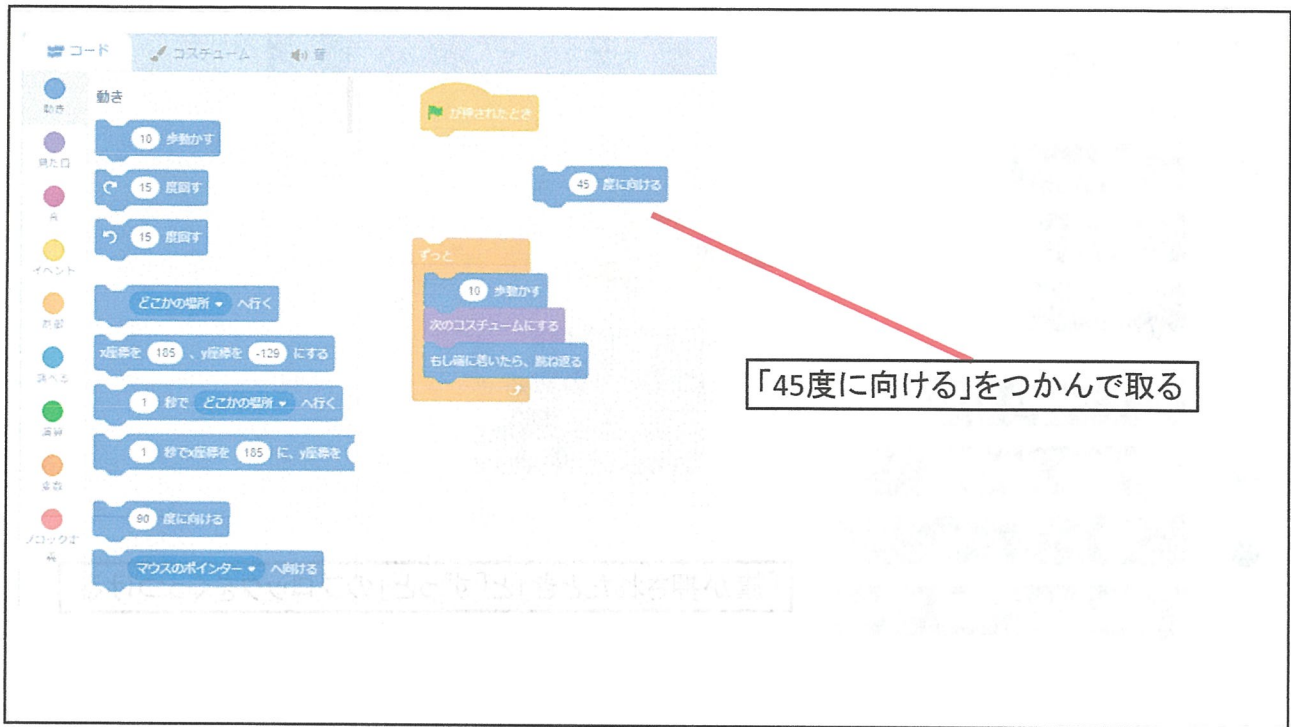


3

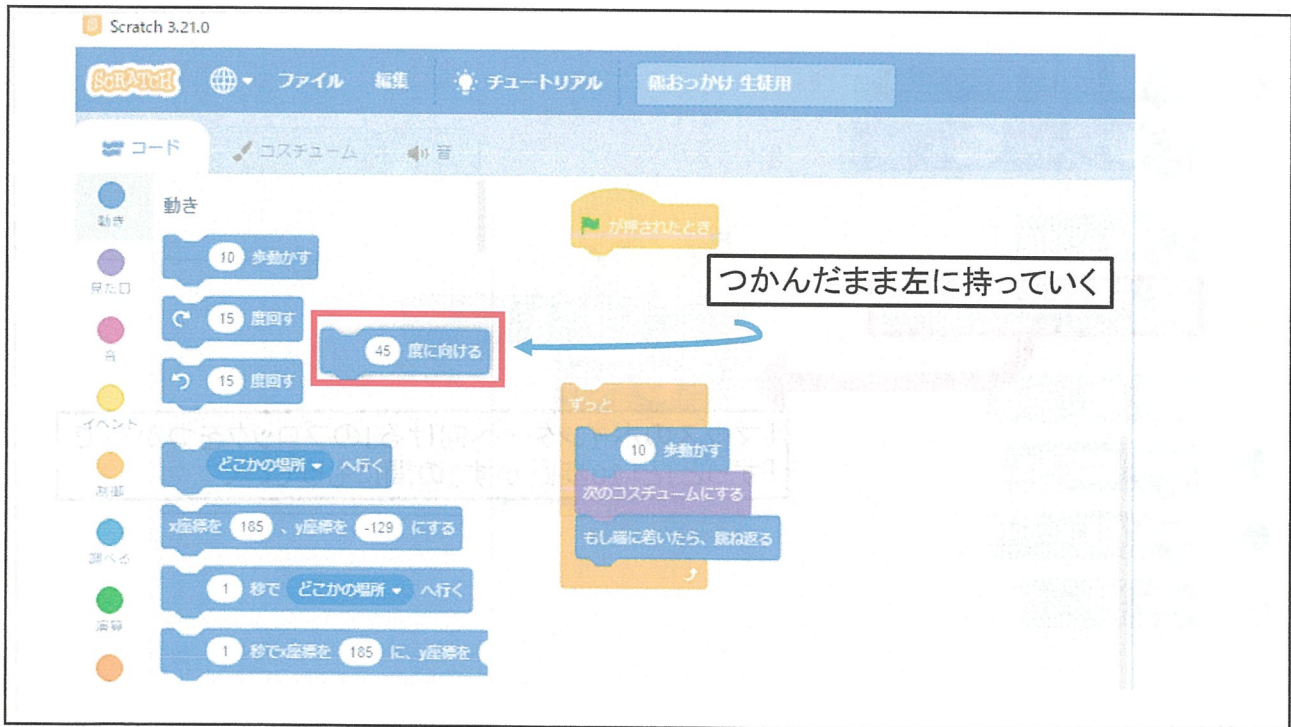
## ウサギを追いかける処理を追加しよう



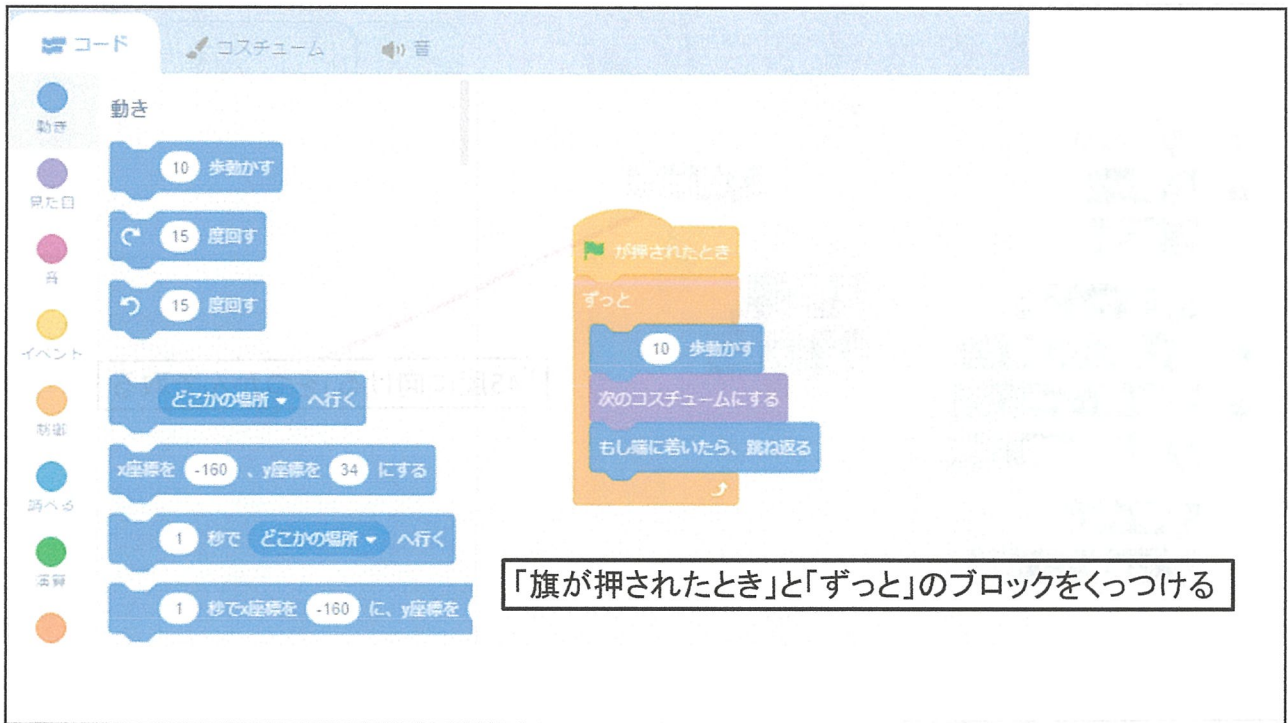
4



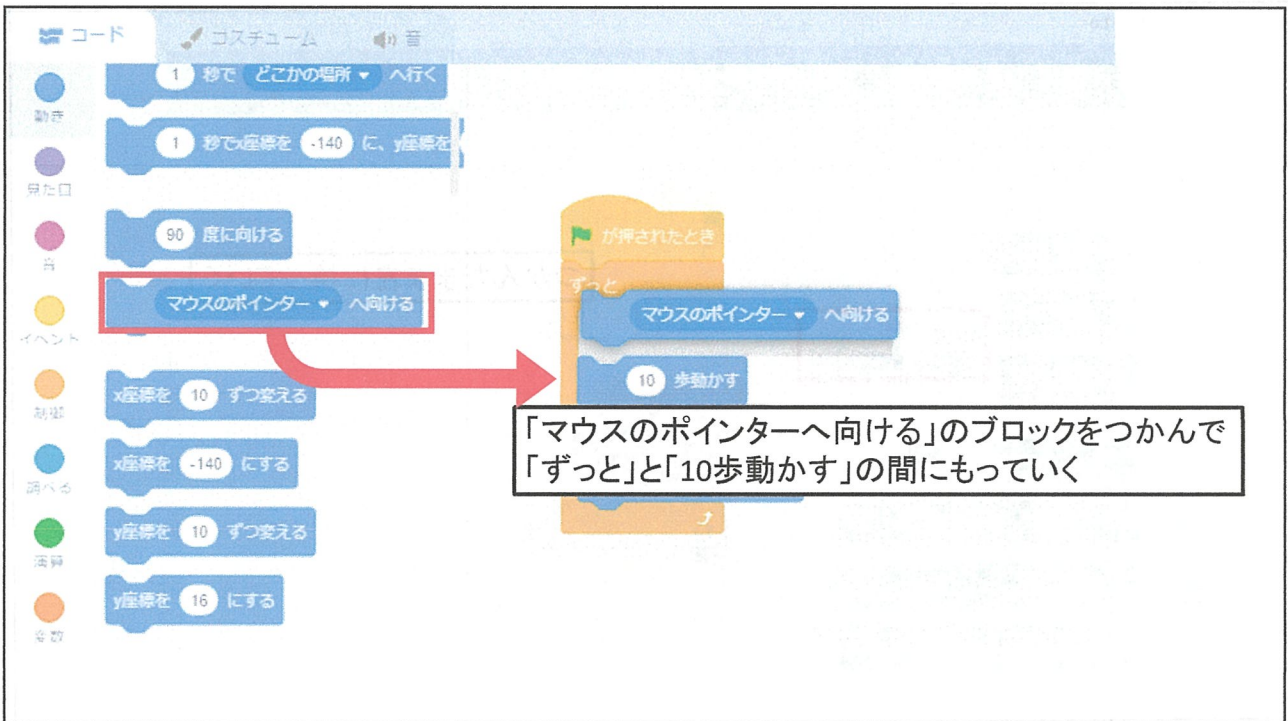
5



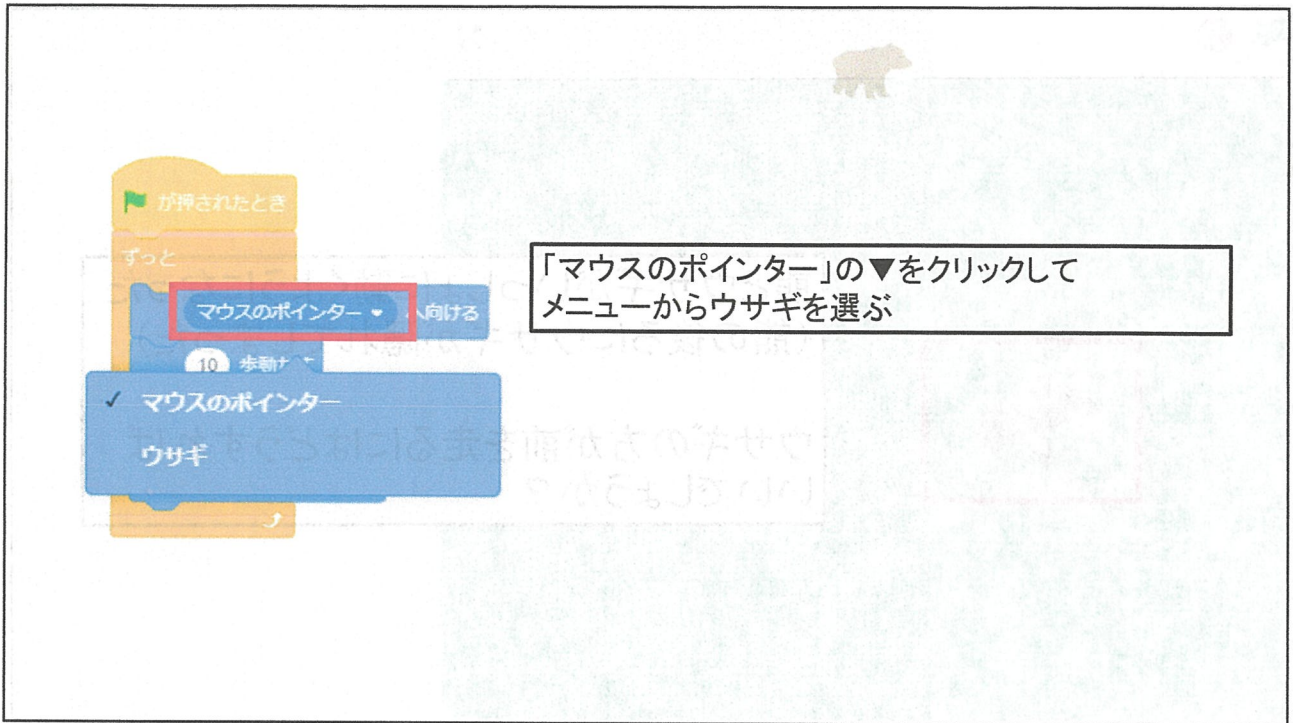
6



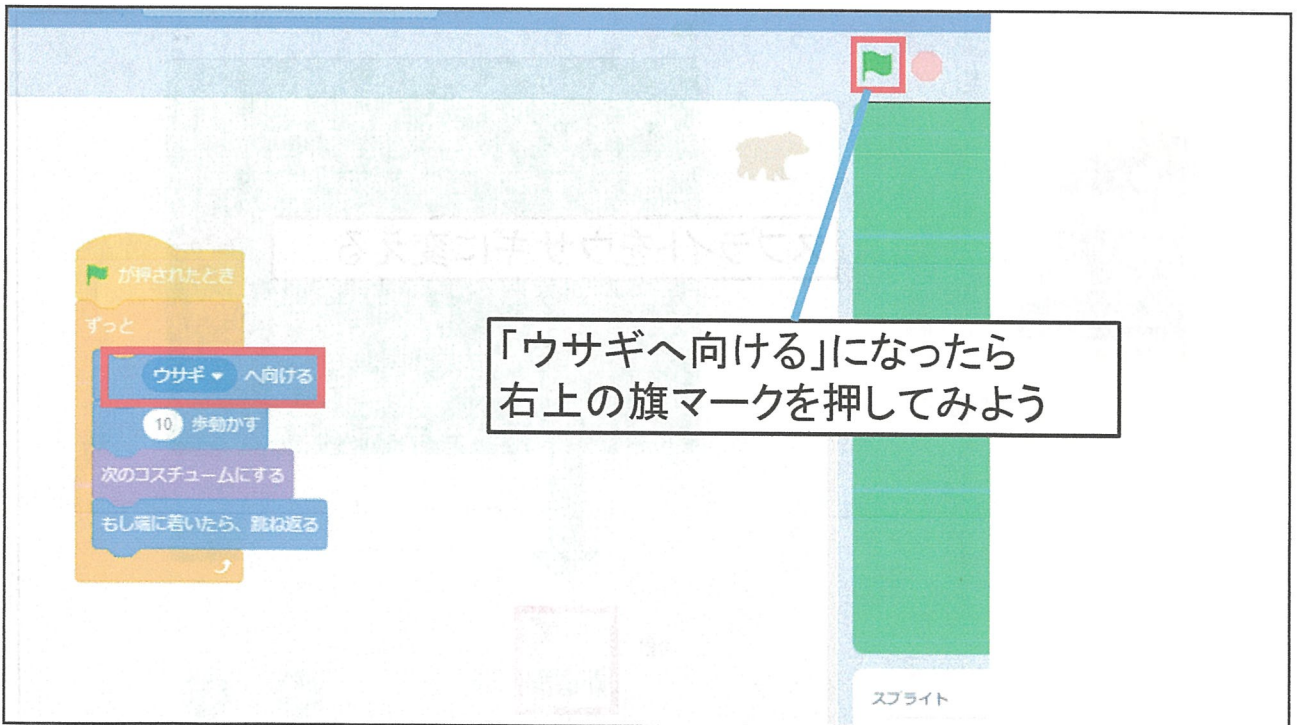
7



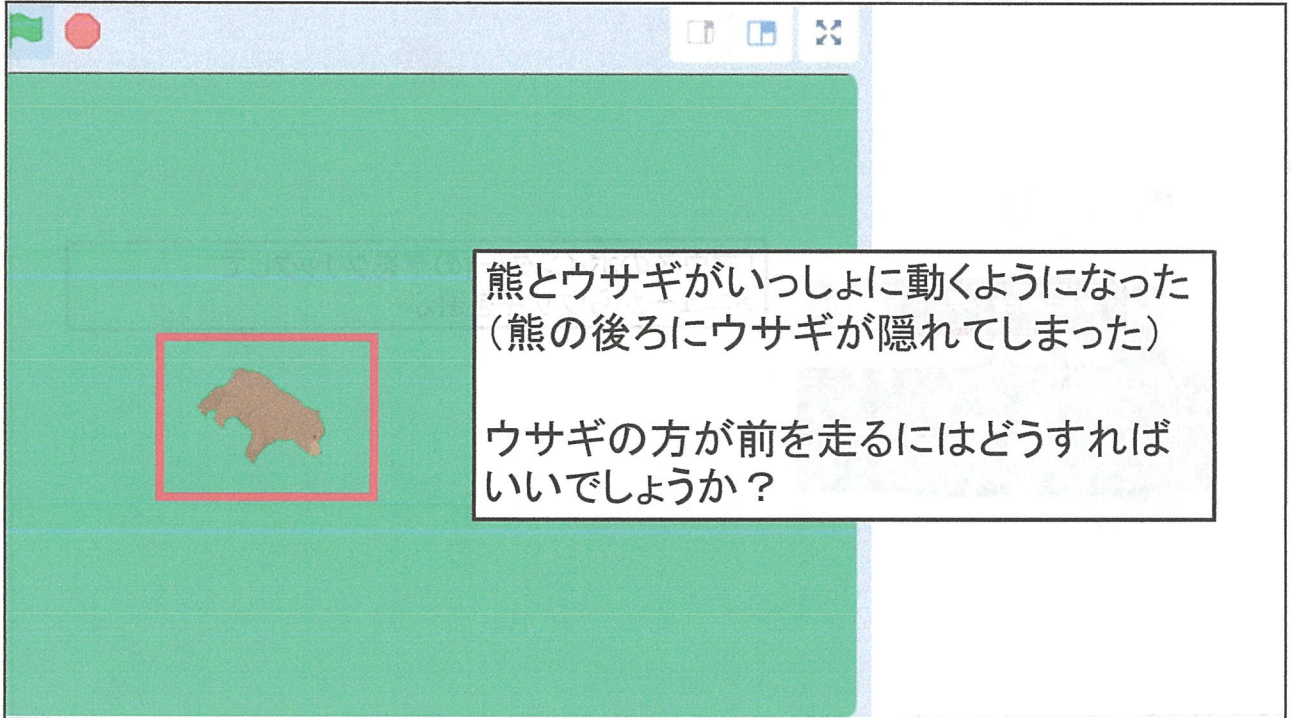
8



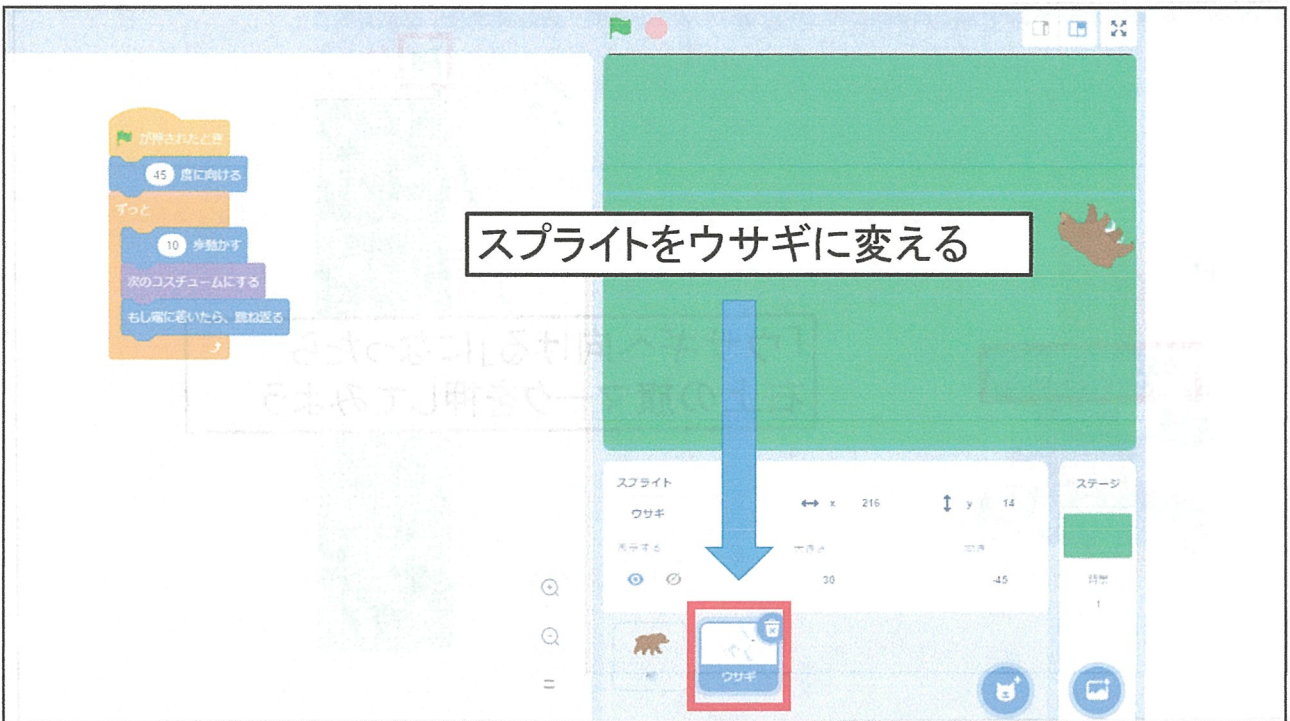
9



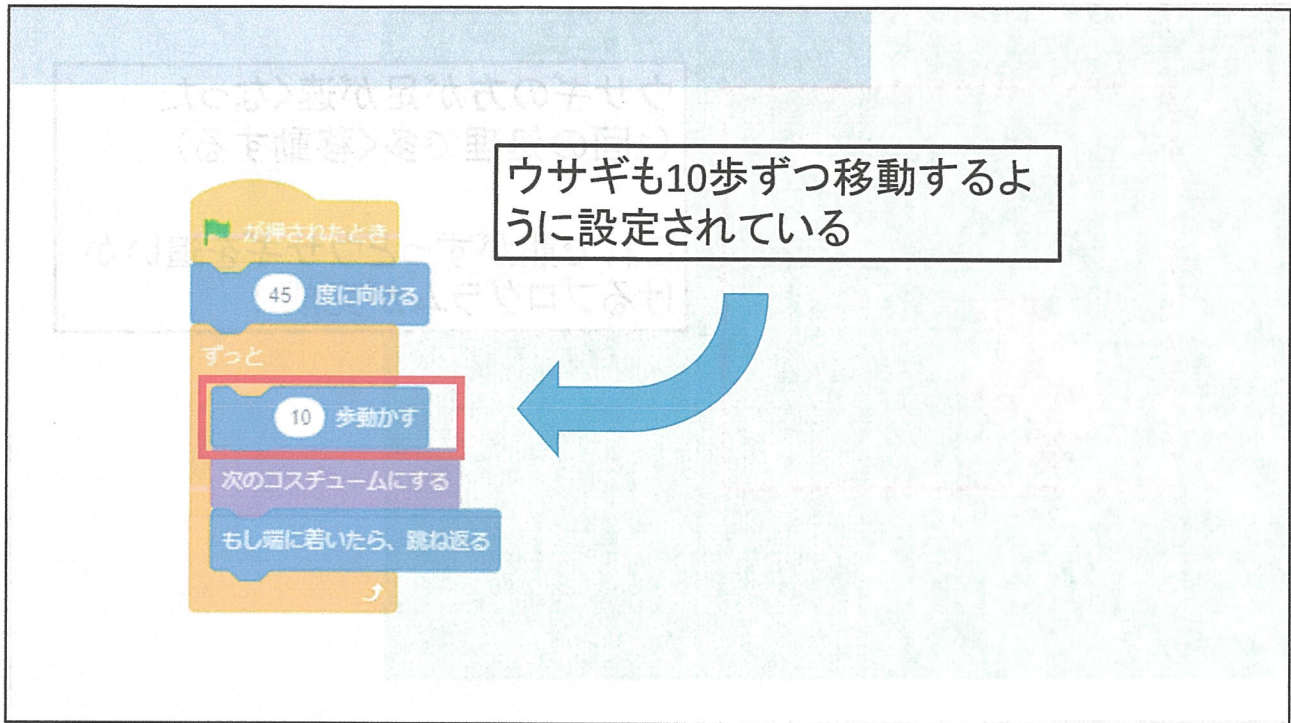
10



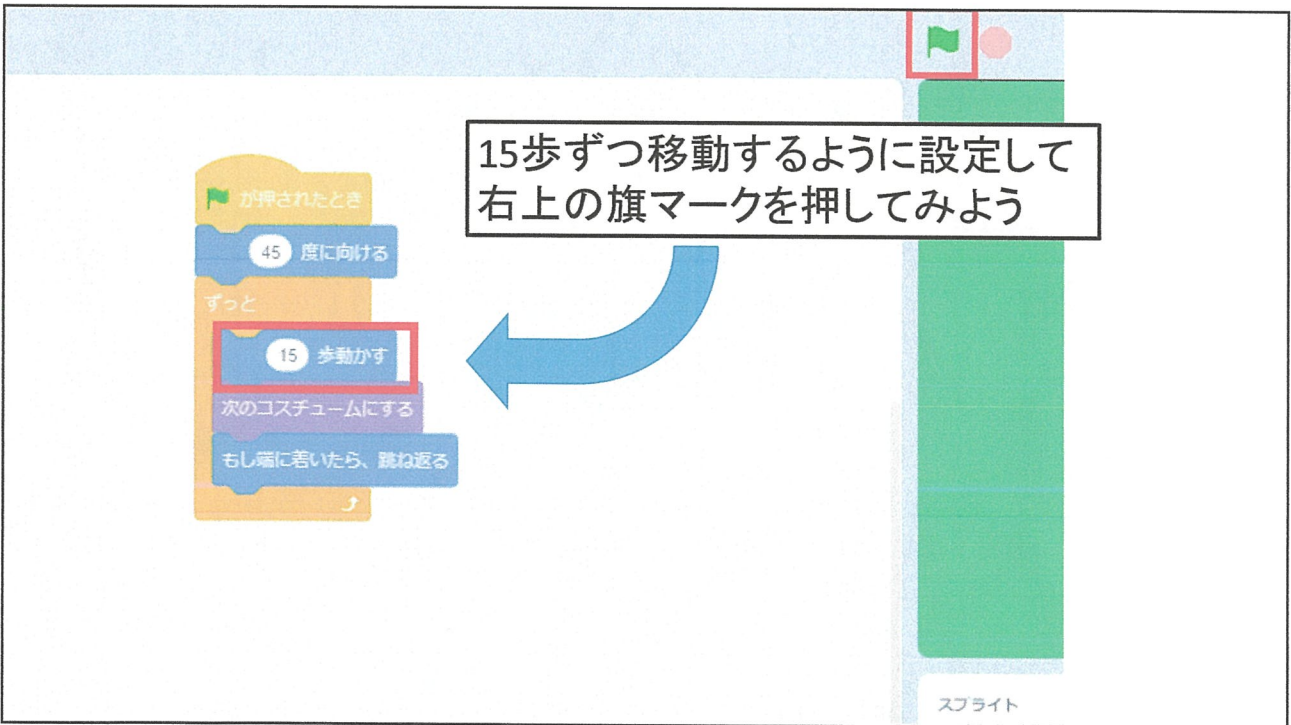
11



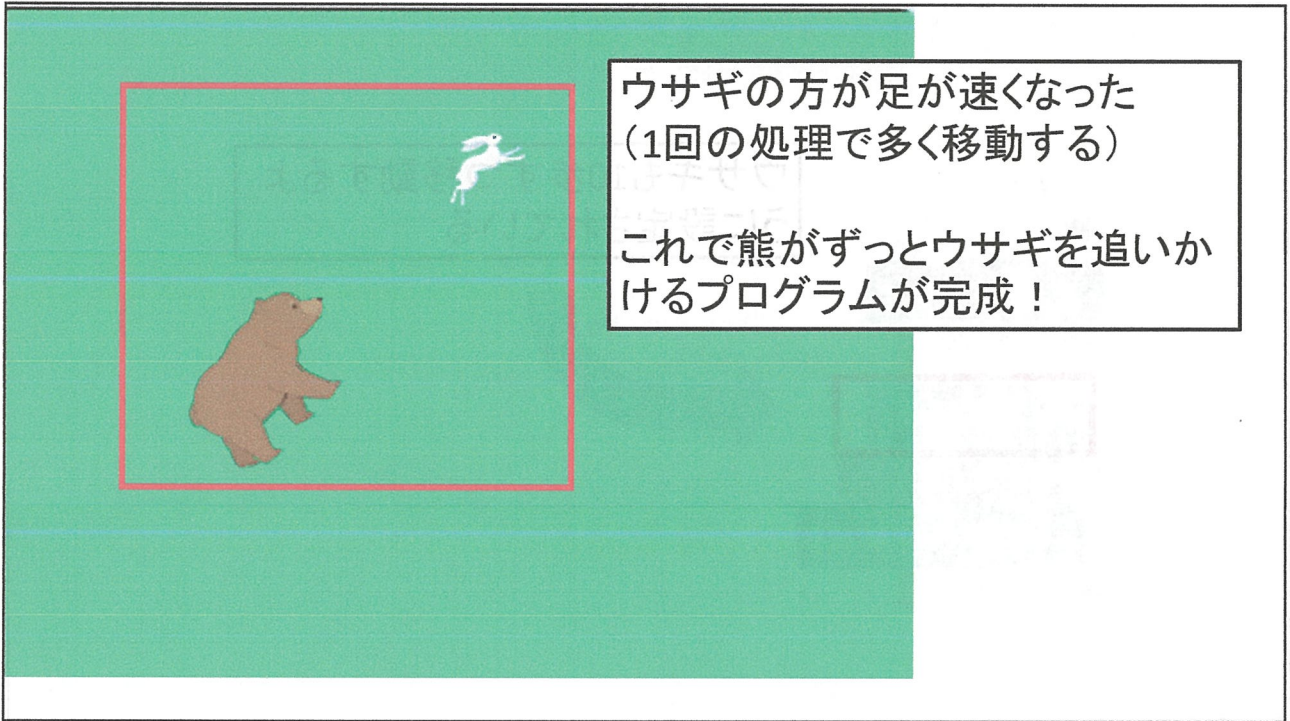
12



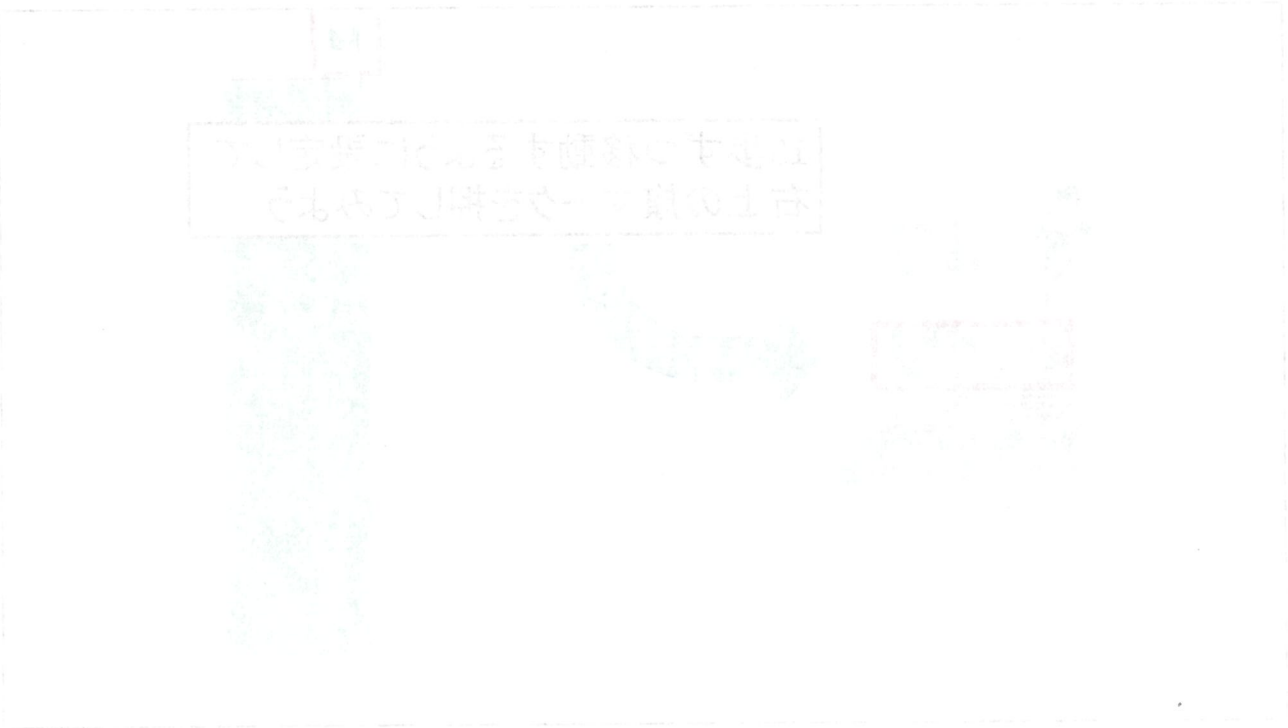
13



14



15





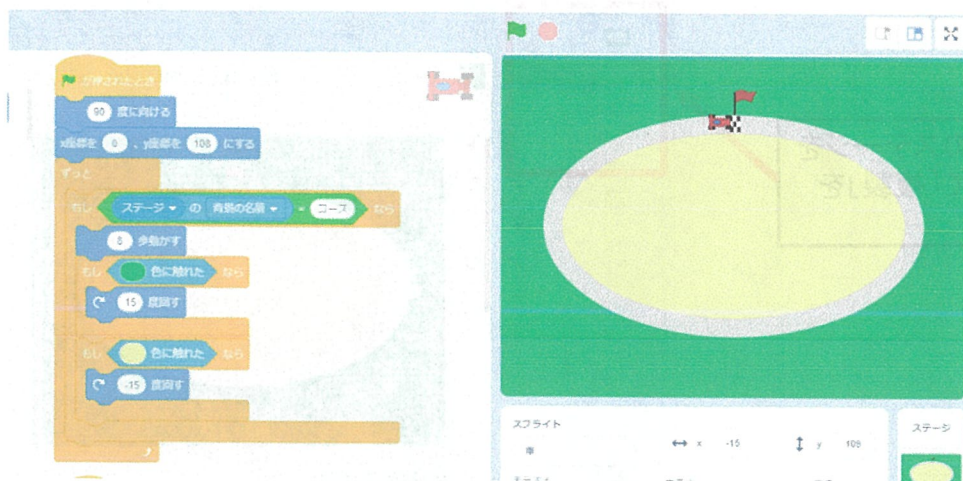
②

さらに大回り世界コースで走る画面  
 作り上げよう

## 自動で走る車の コースを追加してみよう

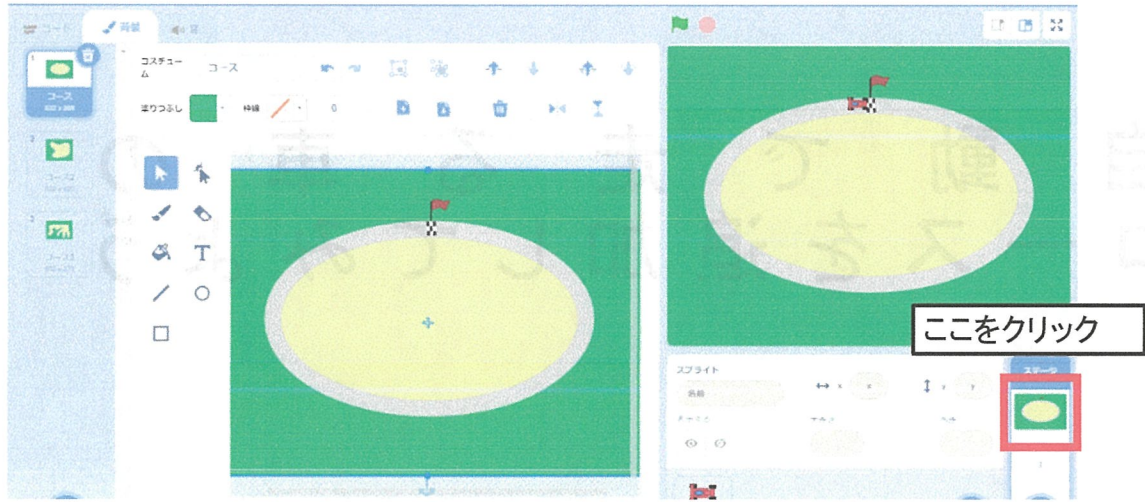
1

旗マークを押したときに自動的に指定されたコースを車が走るプログラムがある  
 4つ目のコース用のプログラムを作成して動かしてみよう



2

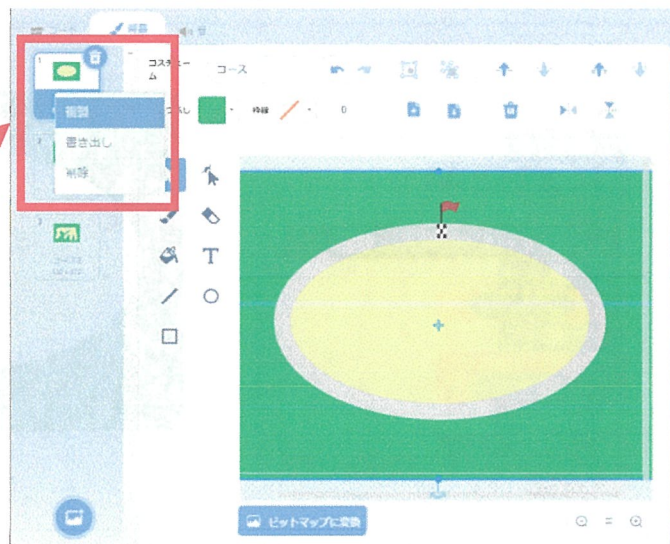
画面をステージに切り替えてみよう  
3つのコースが用意されている



3

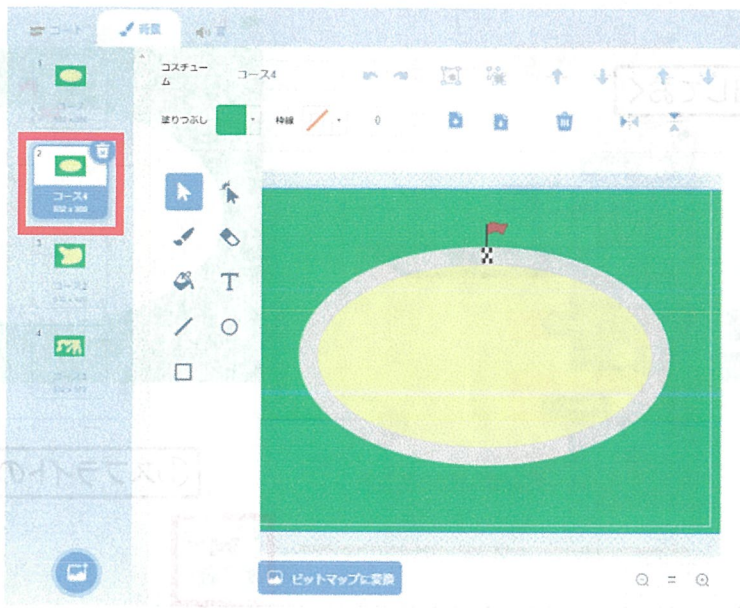
最初に4つめのコースを追加しておこう

いちばん上のステージを  
右クリックして「複製」を  
押そう



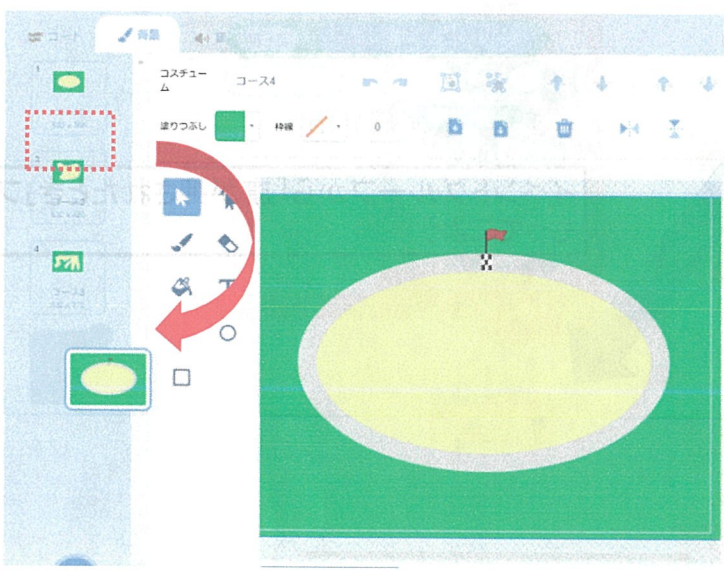
4

コース4が追加された



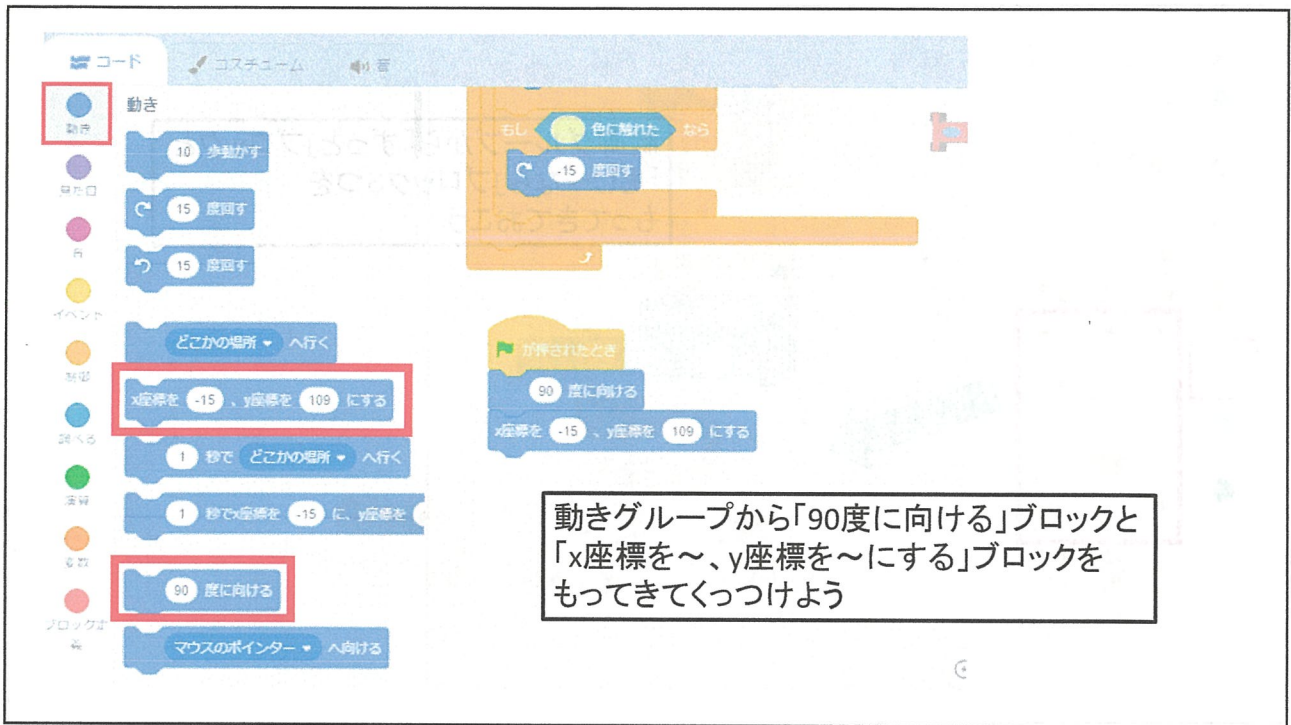
5

コース4をつかんでコース3の下にもっていこう

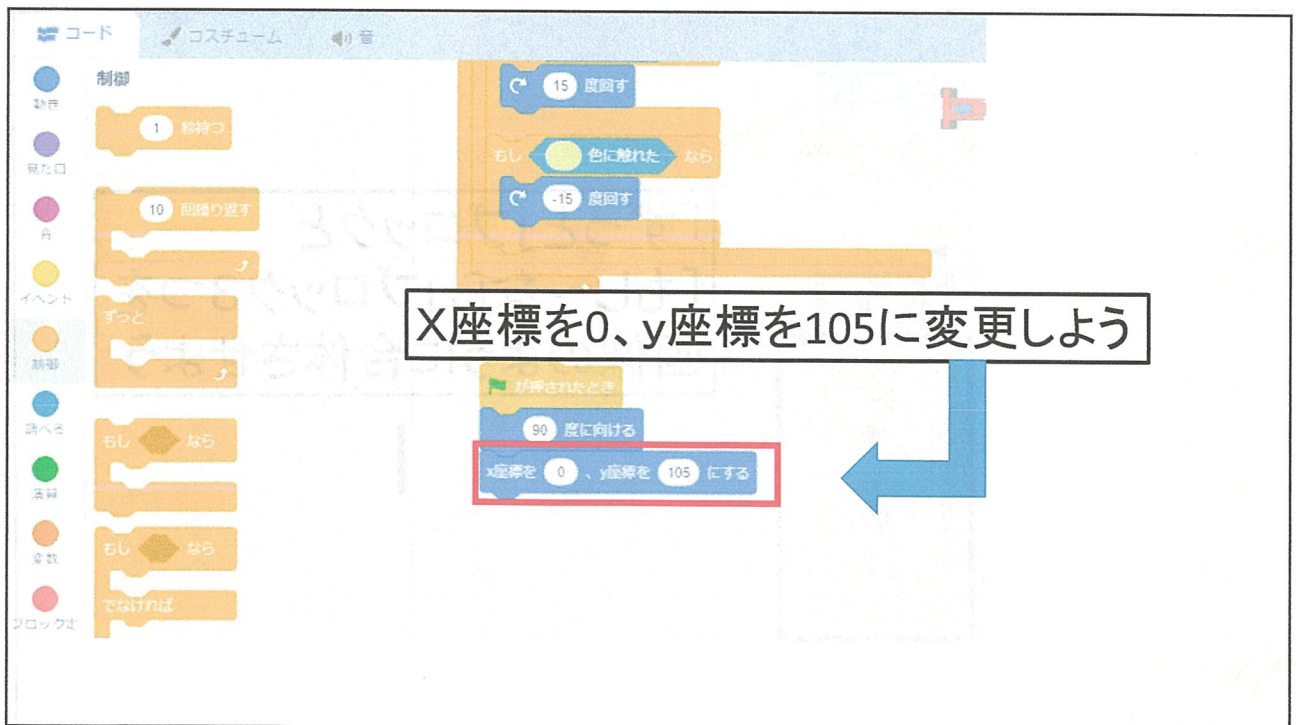


6

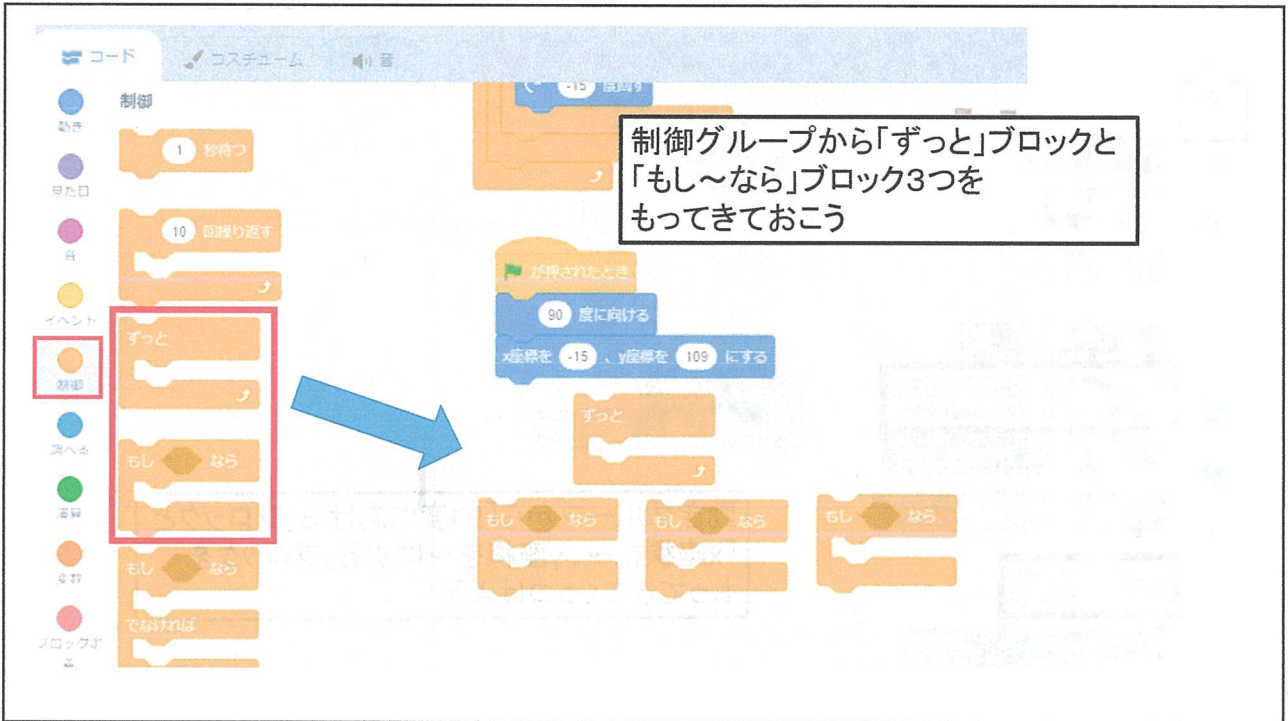




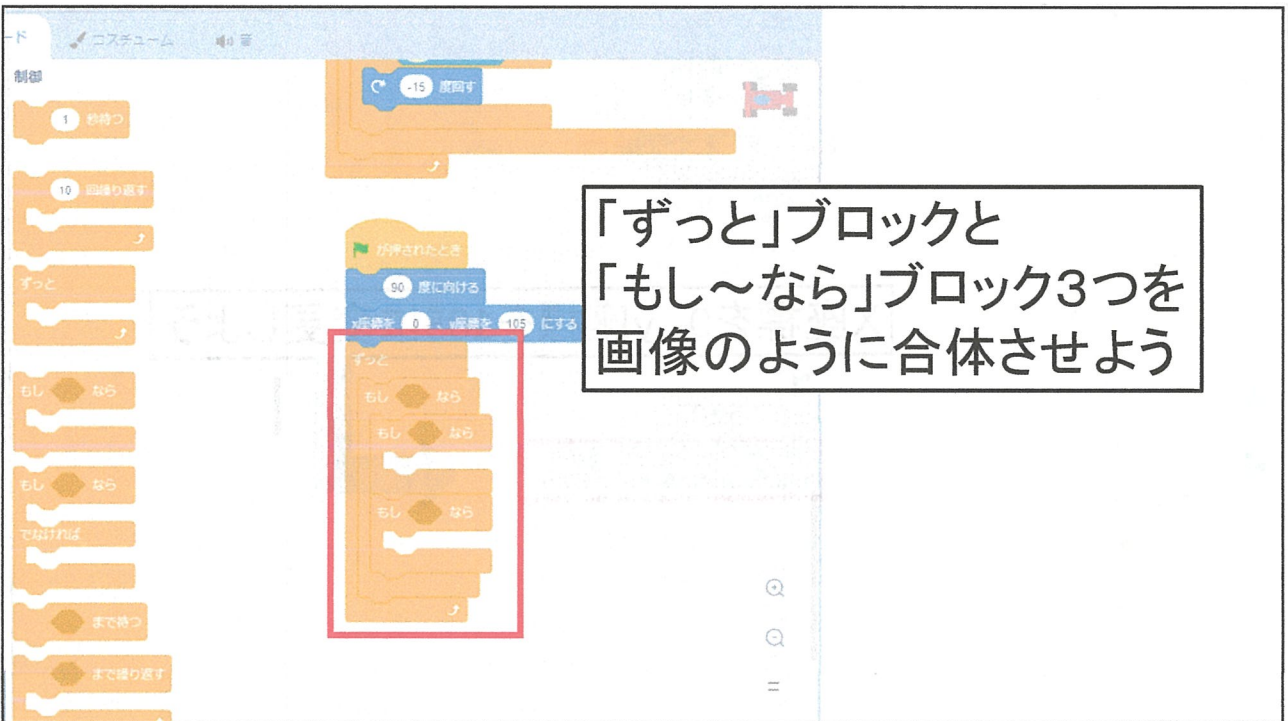
9



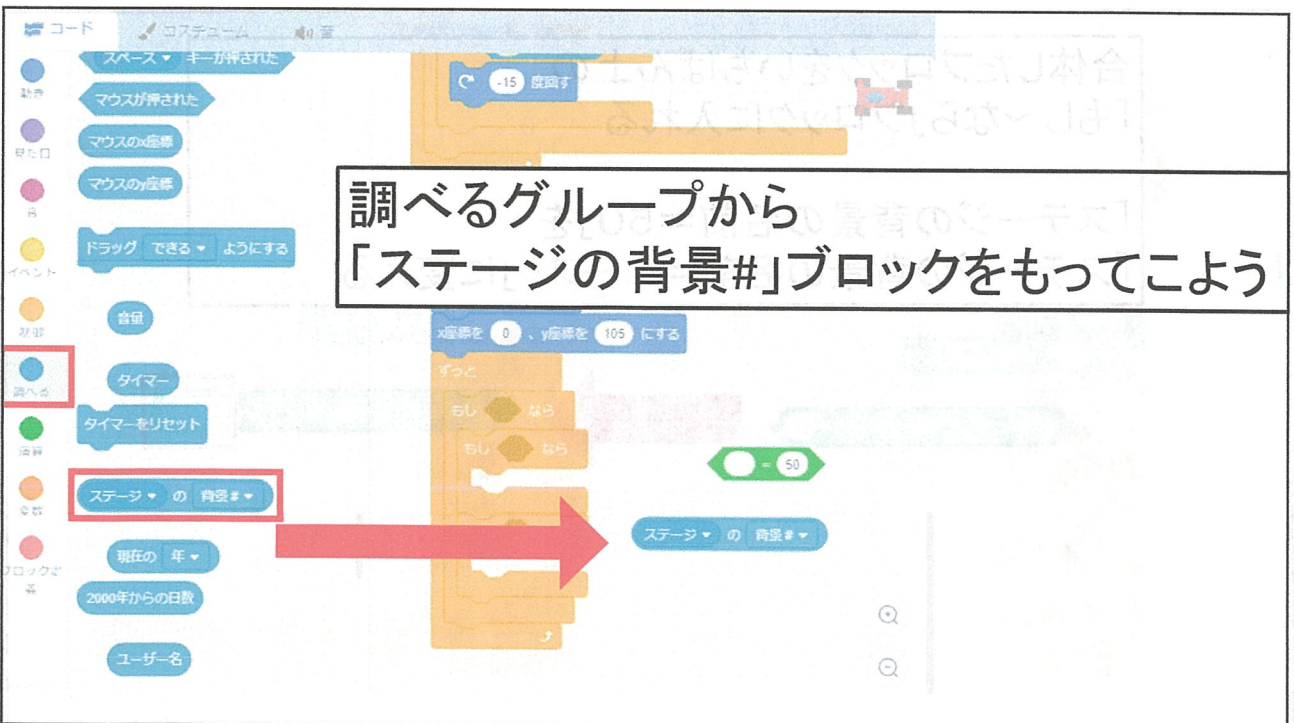
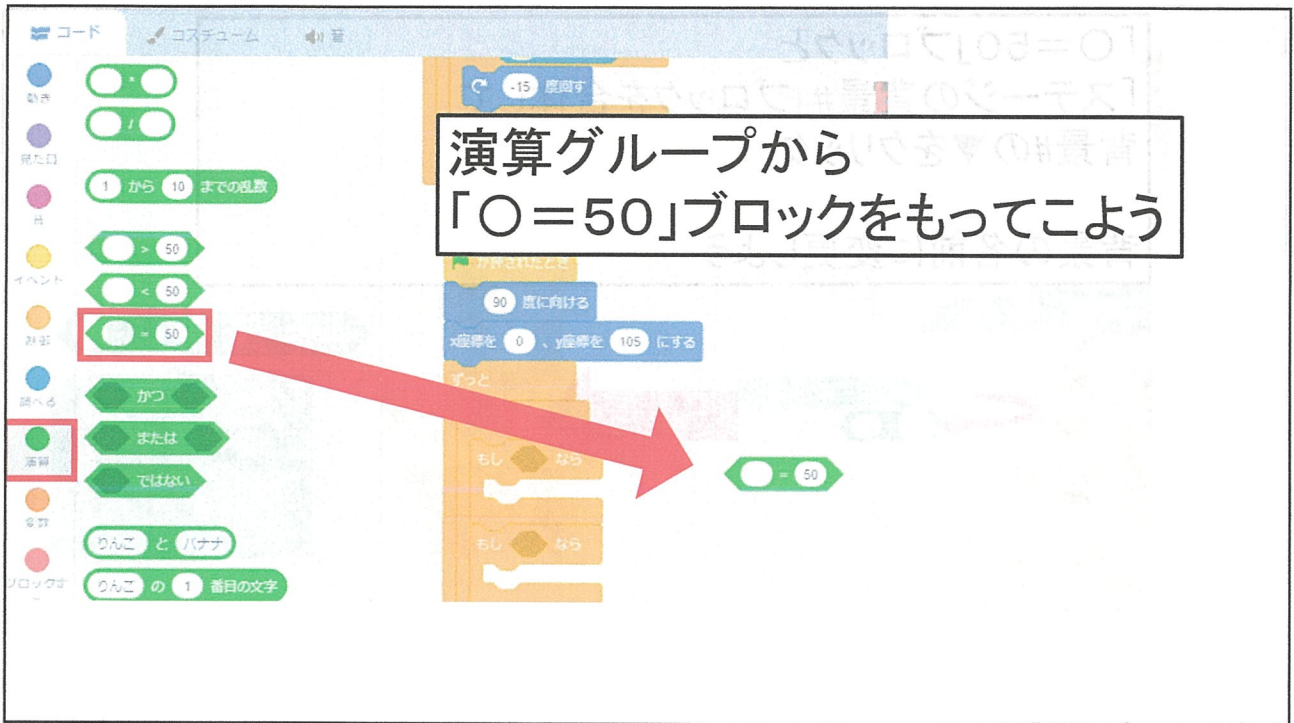
10



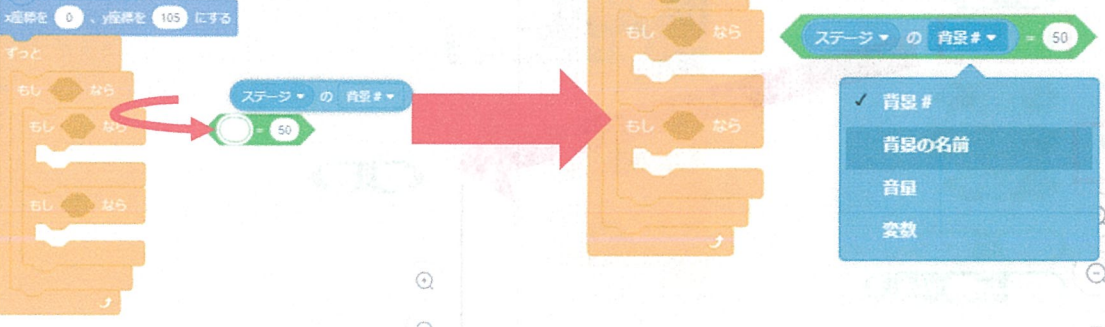
11



12

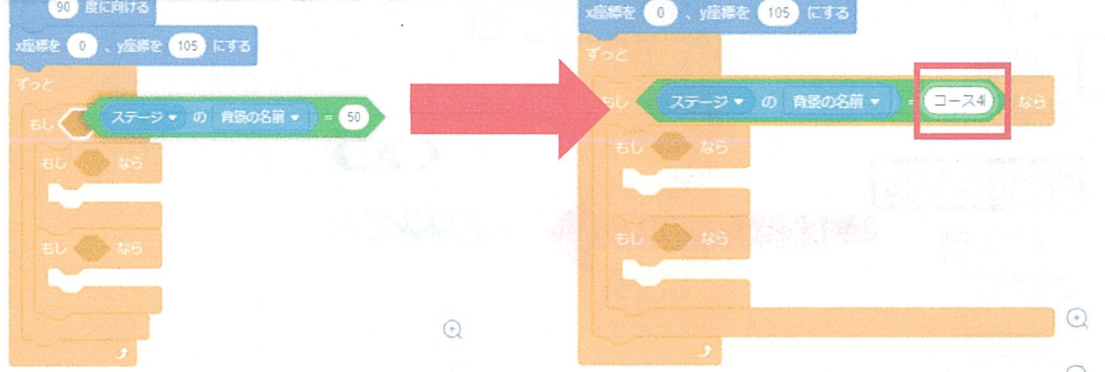


「○=50」ブロックと  
「ステージの背景#」ブロックを合体して  
背景#の▼をクリック  
  
背景の名前に変更しよう



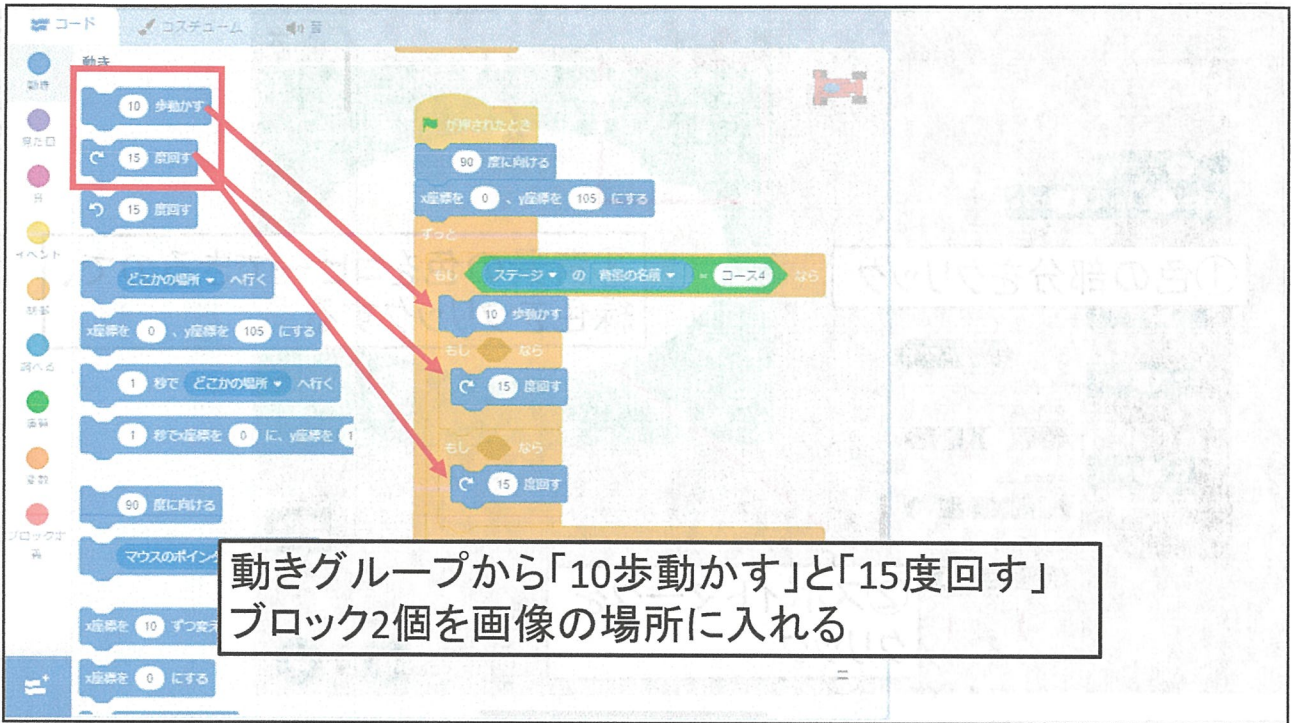
15

合体したブロックをいちばん上の  
「もし～なら」ブロックに入れる  
  
「ステージの背景の名前=50」を  
「ステージの背景の名前=コース4」に変える

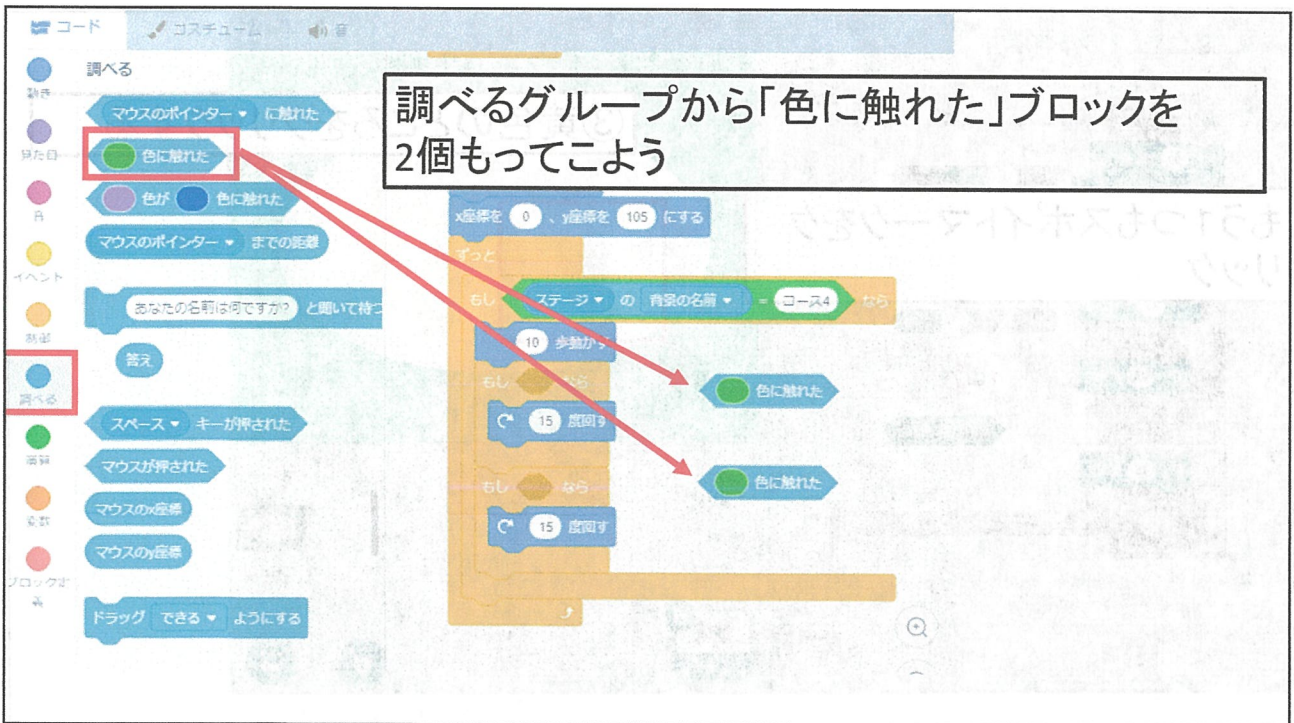


16

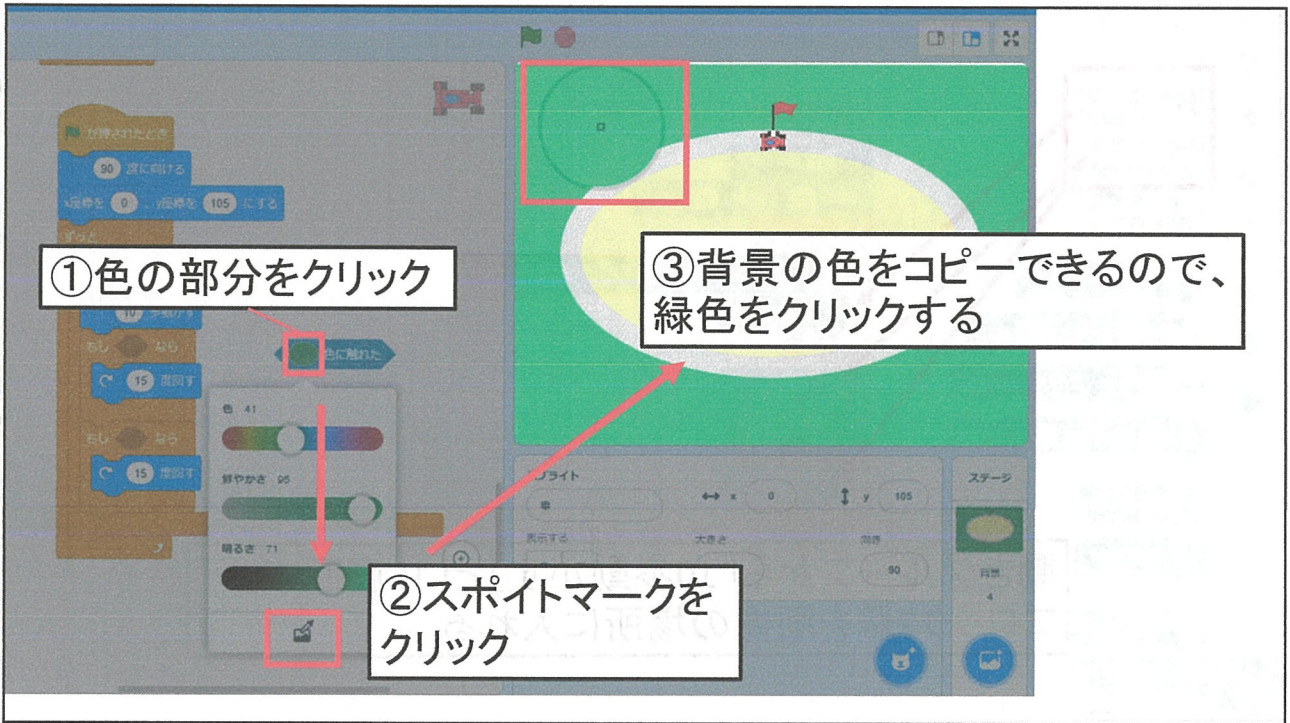




17



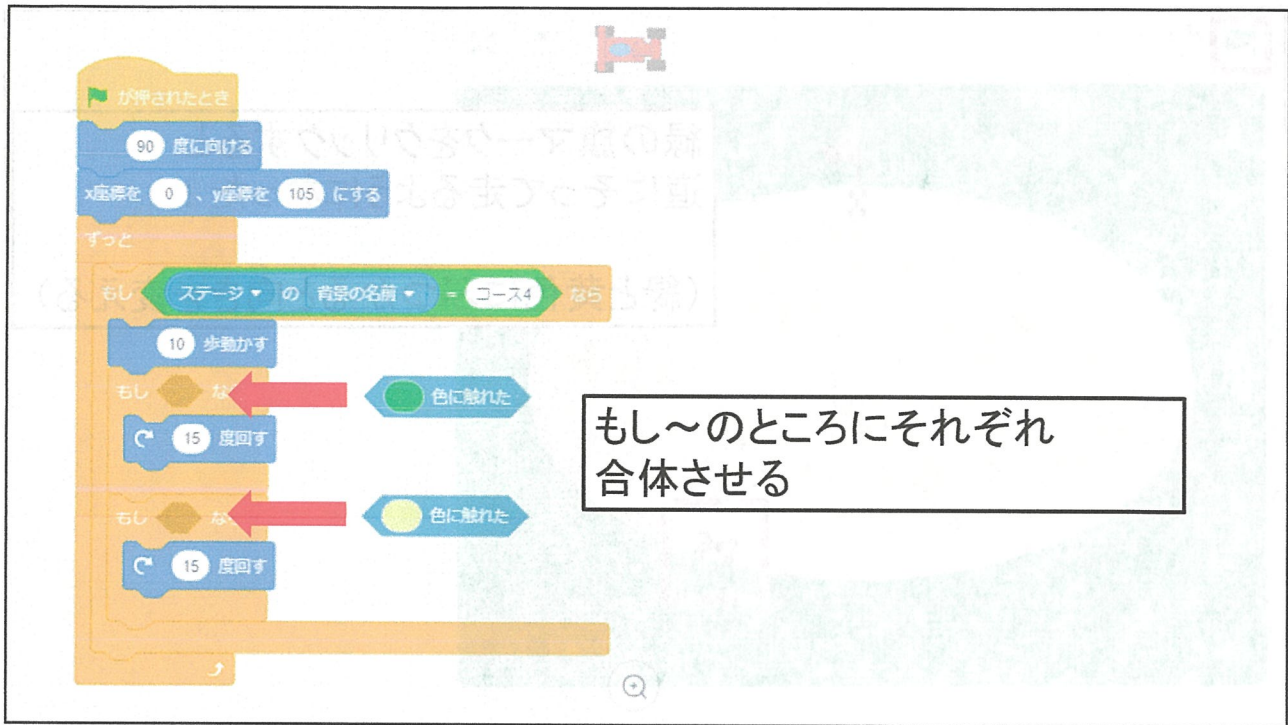
18



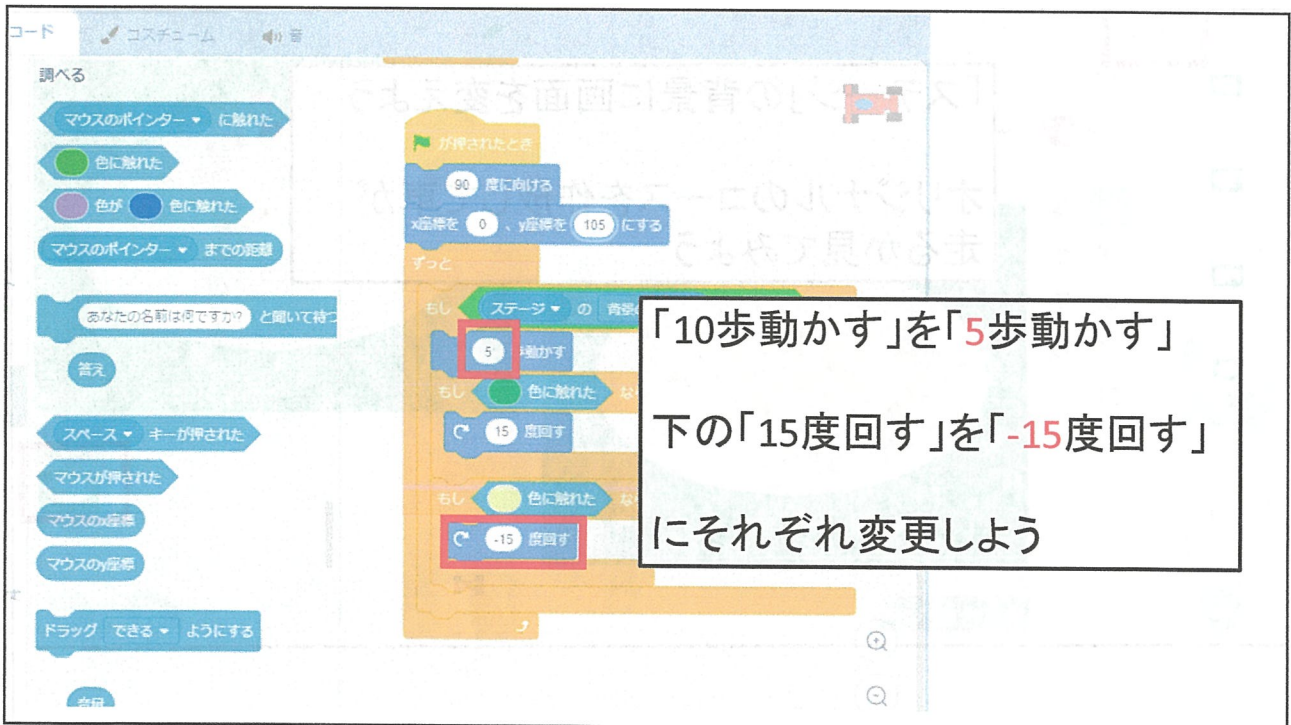
19



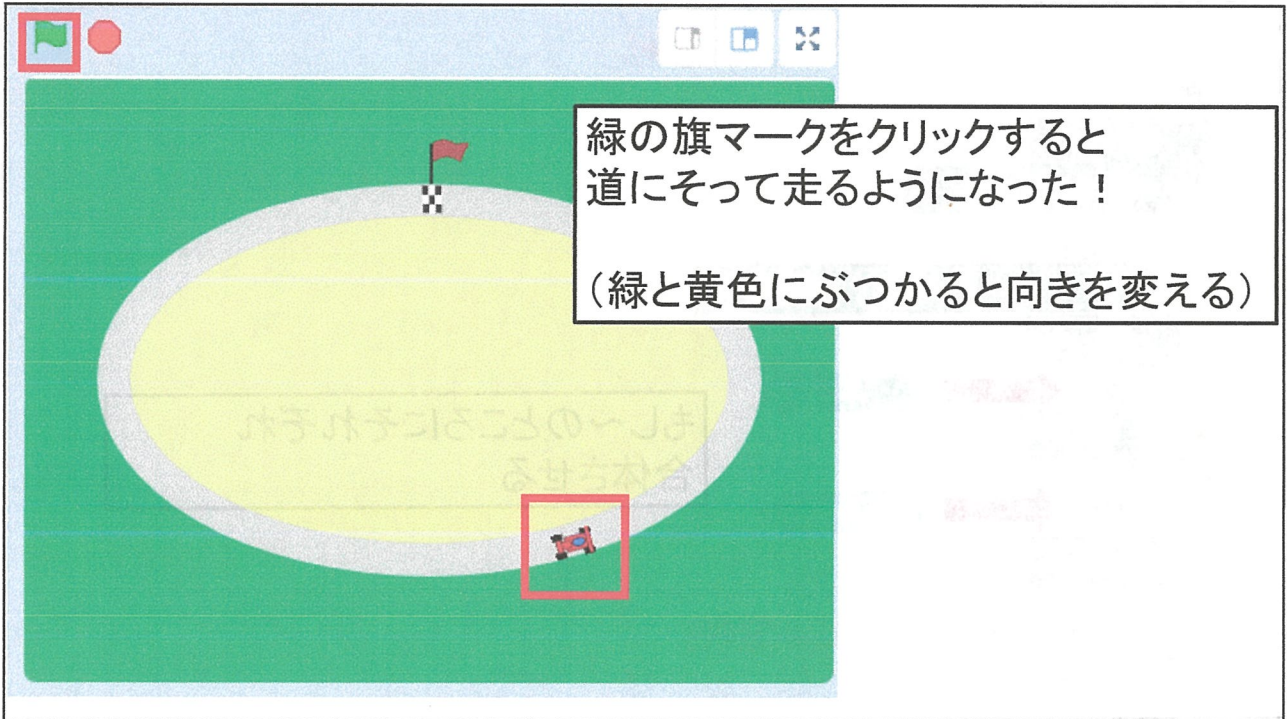
20



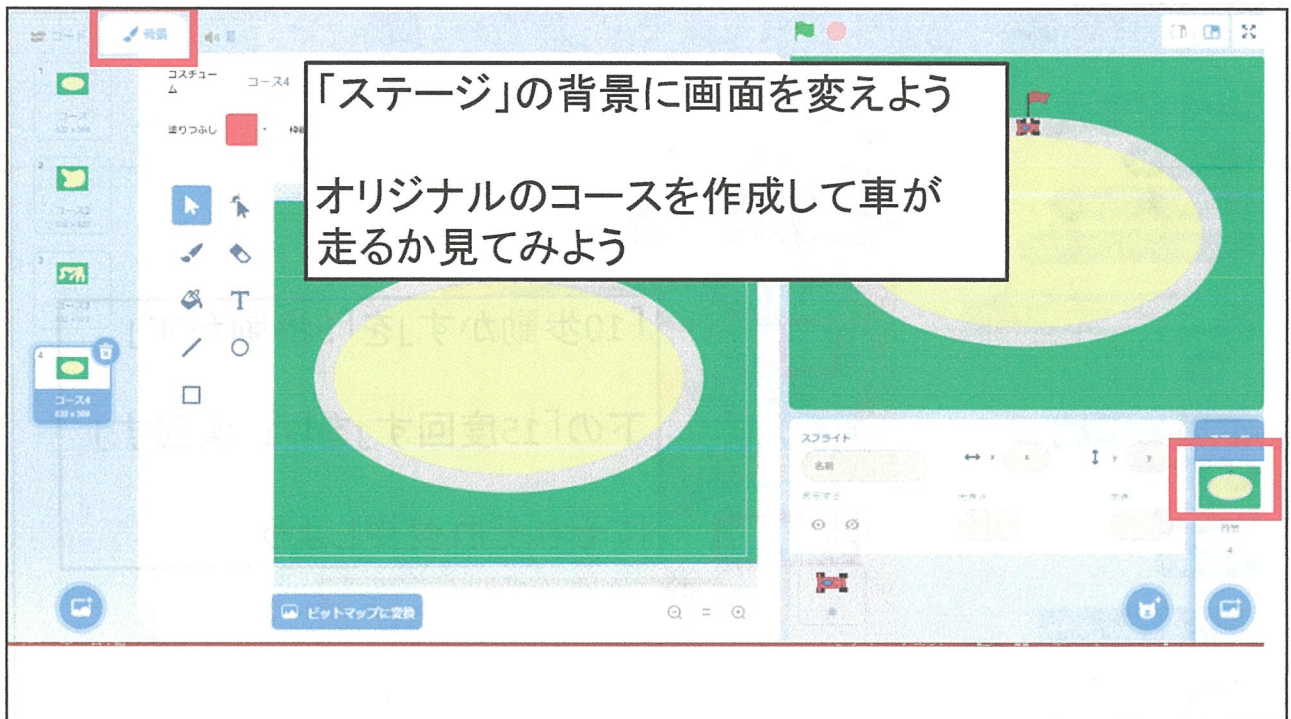
21



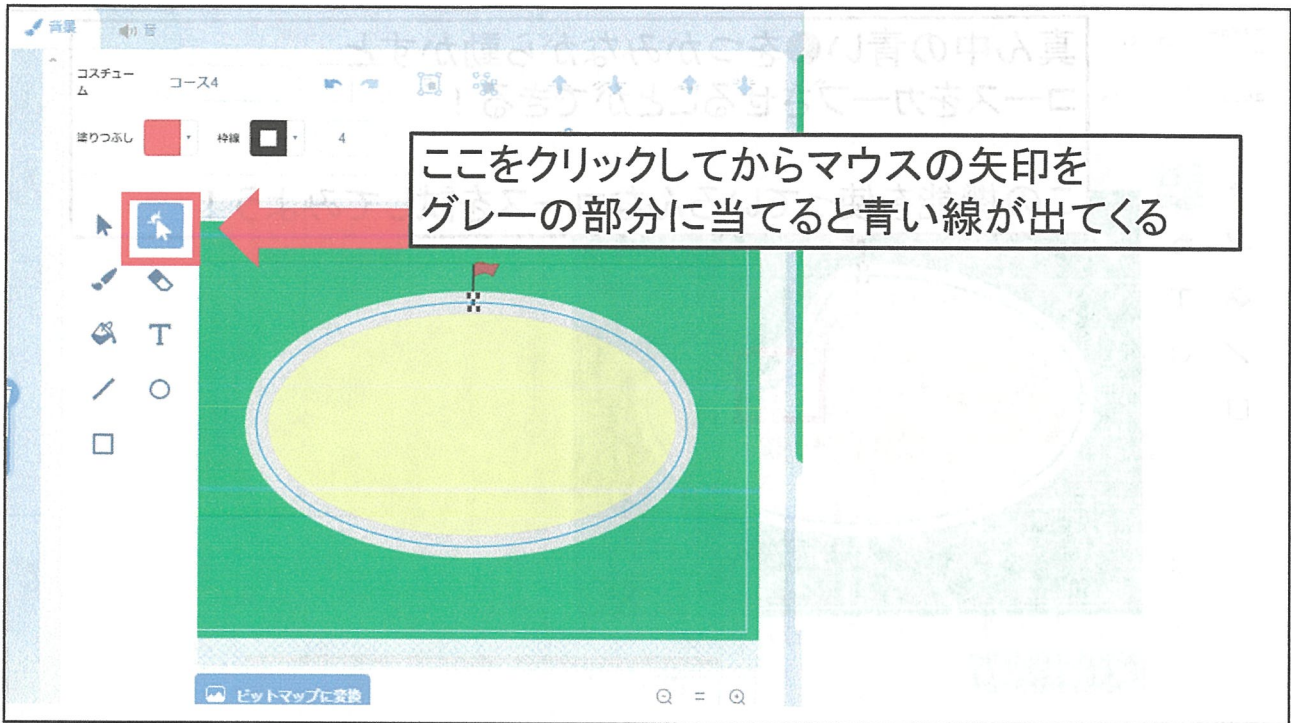
22



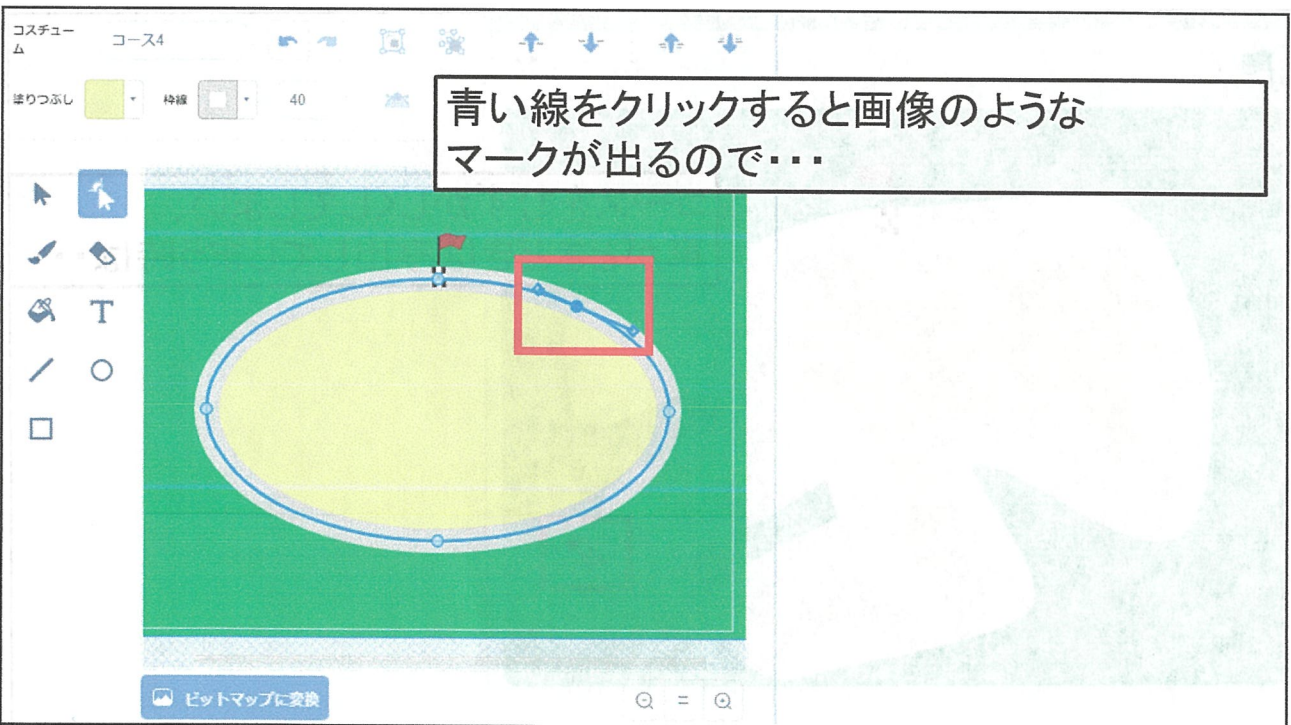
23



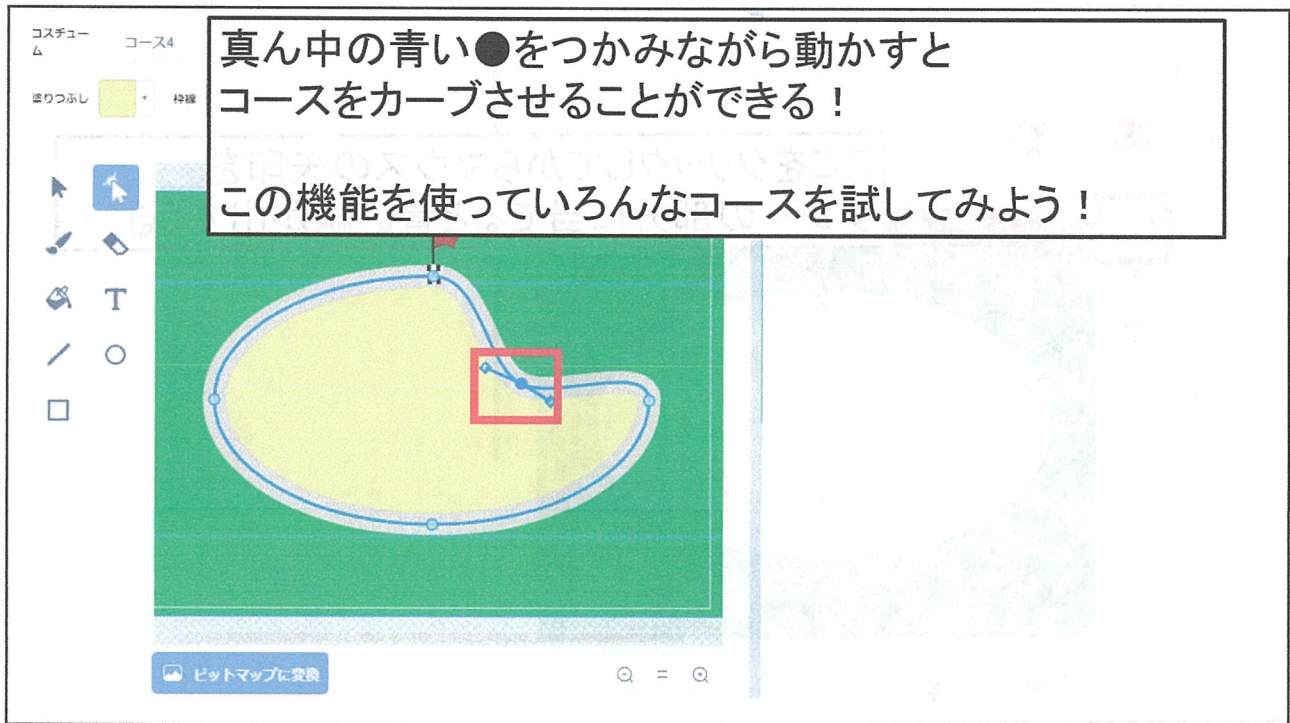
24



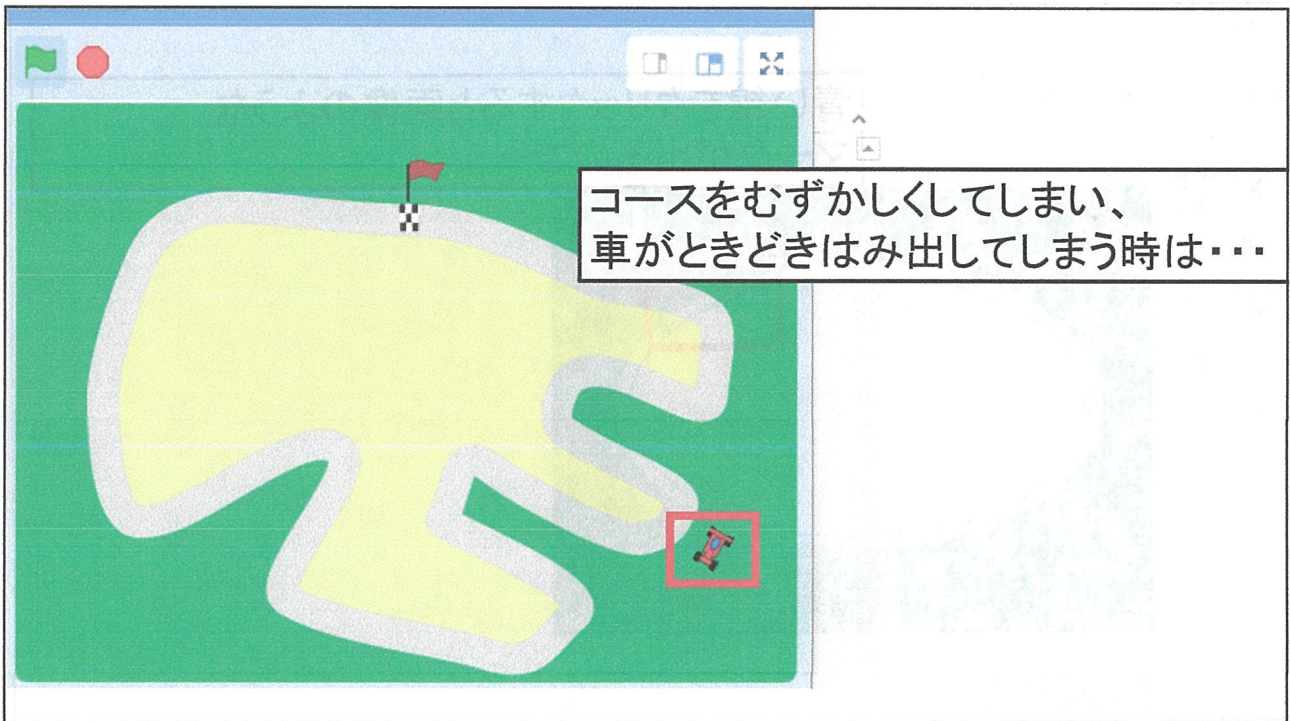
25



26



27



28

画面を車のコードにもどして  
動かす歩数を少なめにしたり、  
回る角度を変えてみよう！

The screenshot shows the Scratch interface with a code block for a car. The code block contains the following blocks: 'ステップ' (Steps) set to 5, '色に触れた' (Touched color) set to green, '戻す' (Reset) set to 15 degrees, and '角度' (Angle) set to -15 degrees. The car sprite is highlighted in the sprite area.

29





③

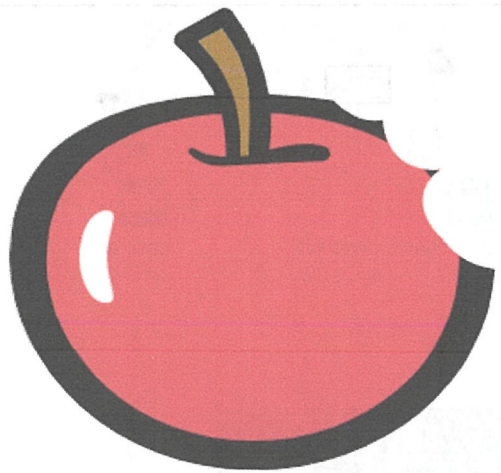
Micro bitを振ったときにリンゴがどんだんかじられるプログラムを組み立てよう

# Microbitと連動させてみよう

1



Micro bitを振ったときにリンゴがどんだんかじられるプログラムを組み立てよう！



2

画面の左下にある、拡張機能を追加をクリックしよう



3

Microbitをクリックしよう



4

# この画面が出たらいったん×をクリック



5

ブロックメニューの一番下に、Microbit用ブロックが追加される



6

8



「音」から「～の音を鳴らす」を持ってきて  
組み合わせよう

9

「制御」から「0秒待つ」を持ってきて  
組み合わせよう

秒数を「0.5秒待つ」に変更しよう

10

「見た目」から「次のコスチュームにする」を持ってきて組み合わせよう

The image shows the Scratch 'Looks' menu on the left. The 'next costume' block is highlighted with a red box. A red arrow points from this block to a custom block assembly on the right. The assembly consists of the following blocks from top to bottom: 'when clicked' (green), 'show sprite' (green), 'play bite sound' (purple), 'wait 0.5 seconds' (orange), and 'next costume' (purple).

11

「制御」から「もし~なら」を持ってきて組み合わせよう  
 同じ動きをくりかえす処理をつくるよ

The image shows the Scratch 'Control' menu on the left. The 'if loop' block is highlighted with a red box. A red arrow points from this block to a custom block assembly on the right. The assembly consists of the following blocks from top to bottom: 'show sprite' (green), 'play bite sound' (purple), 'wait 0.5 seconds' (orange), 'next costume' (purple), and 'if loop' (orange).

12

色 の効果を 0 にする

画像効果をなくす

表示する

隠す

最前面へ移動する

1 層 手前に出す

コスチュームの 番号

背景の

番号

名前

もし なら

コスチュームの 名前

「見た目」の「コスチュームの番号」を「コスチュームの名前」に変更して右側に持っていこう

13

21

もし なら

コスチュームの 名前

O = O

O = apple8

「演算」の「O=O」を右側に持っていこう  
内容を「O=apple8」に変更しよう

14

21

持ってきたブロックを赤い四角で囲んだ図のように組み合わせよう

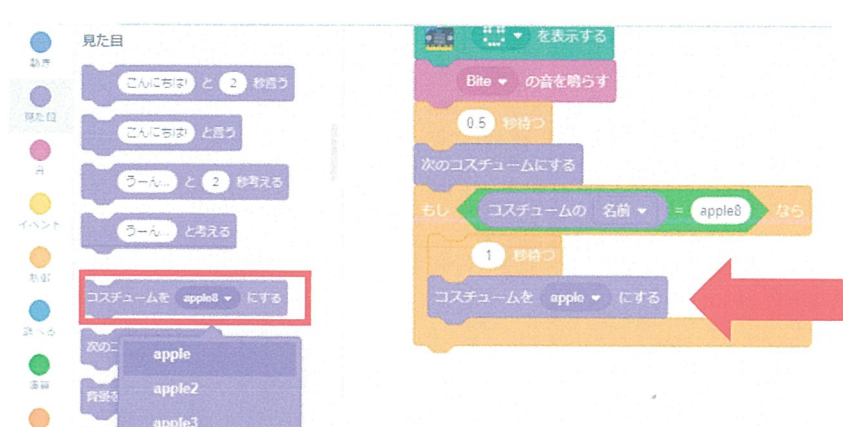
組み合わせたら、「もし～なら」ブロックに合体させよう

15

「制御」から「1秒待つ」を持ってきて図のように組み合わせよう

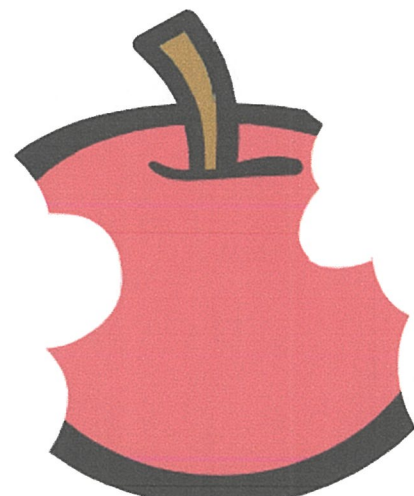
16





「見た目」の「コスチュームをapple8にする」を「コスチュームをappleにする」に変更しよう  
変更したら図のように組み合わせよう

17



完成！  
Microbitとパソコンを繋いで、実際に動かしてみよう

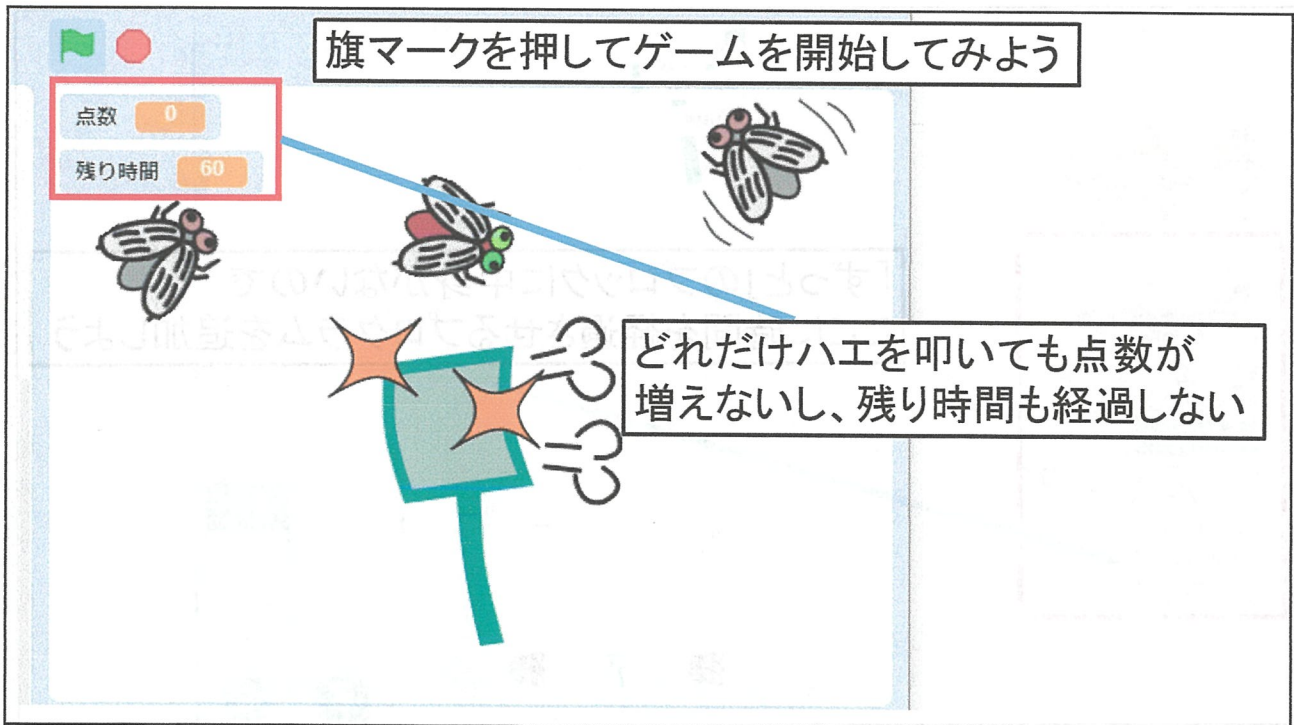
18



④

# ハエたたきゲームに得点と時間を追加しよう

1



2



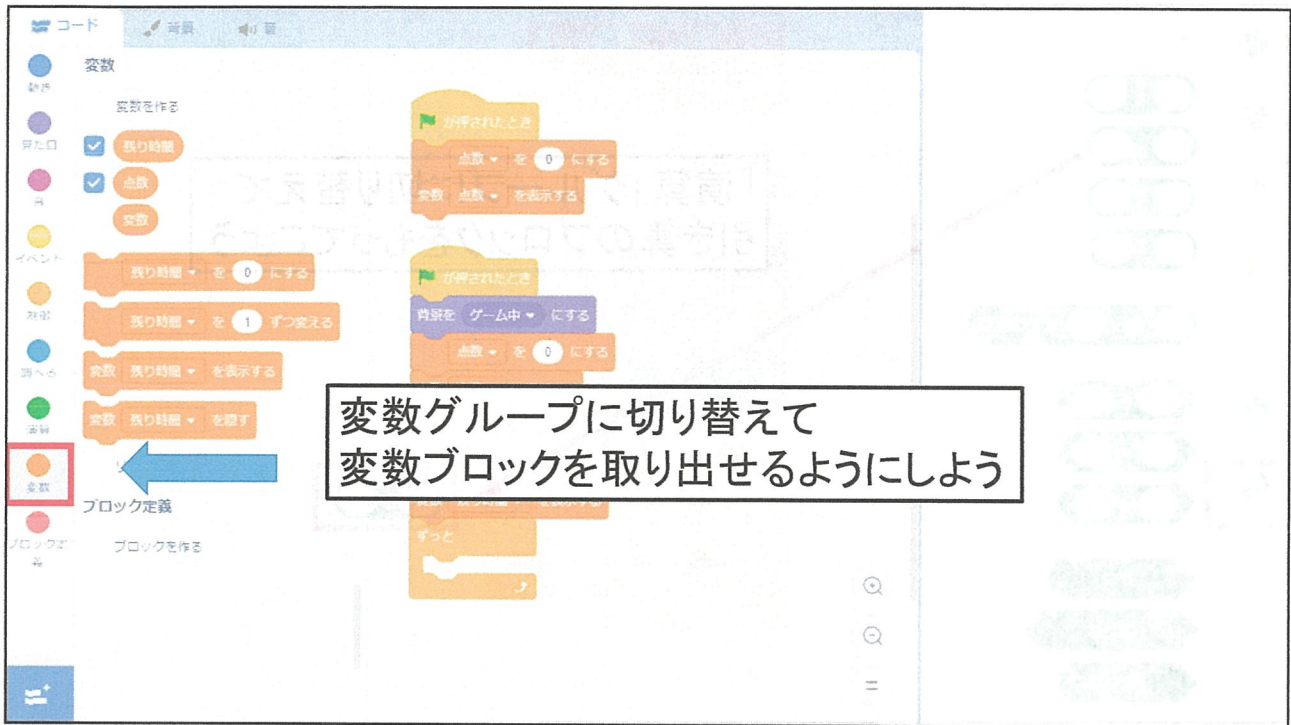
ステージをクリックして  
ゲーム画面のプログラムを開こう

3

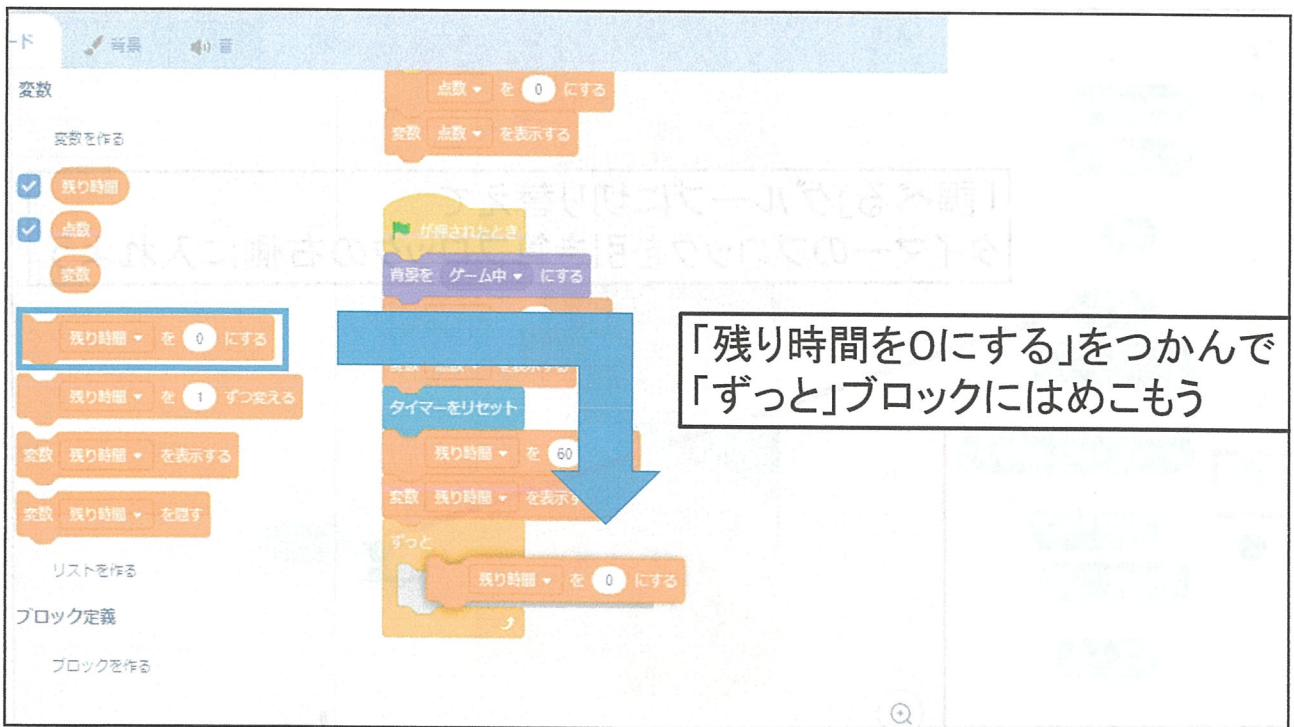


「ずっと」のブロックに中身がないので  
ここに時間を経過させるプログラムを追加しよう

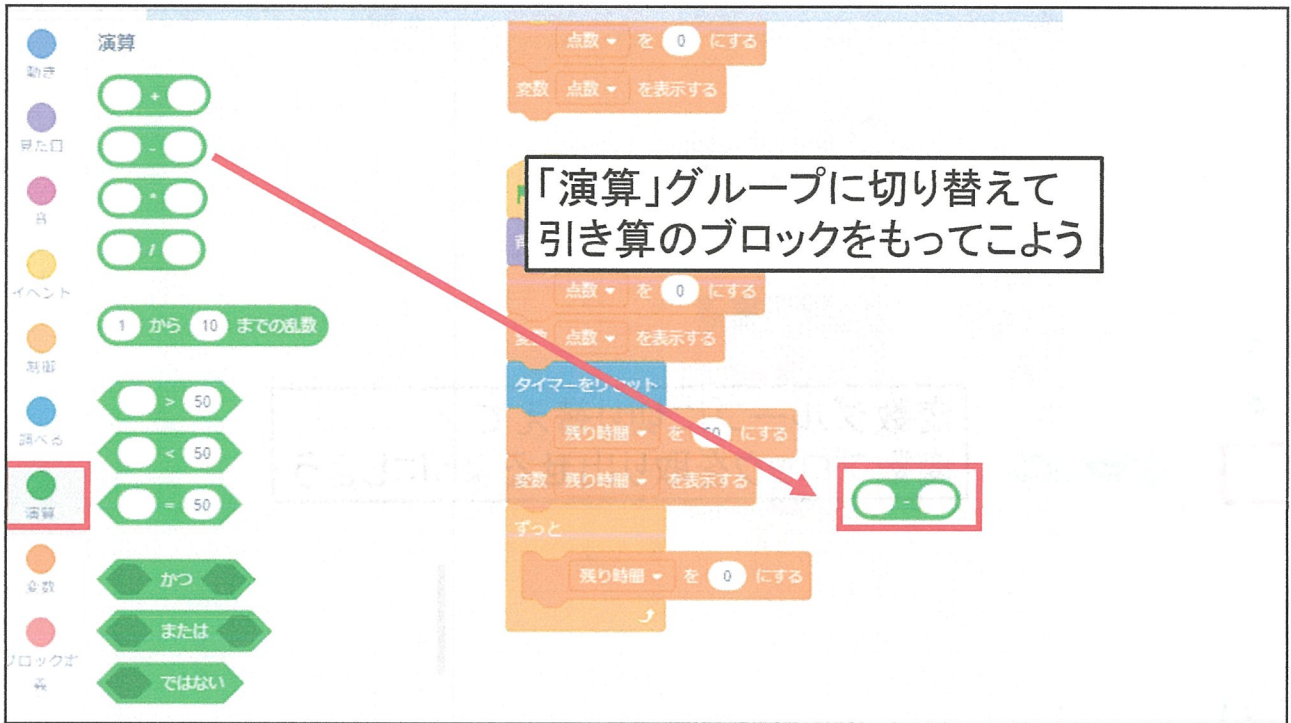
4



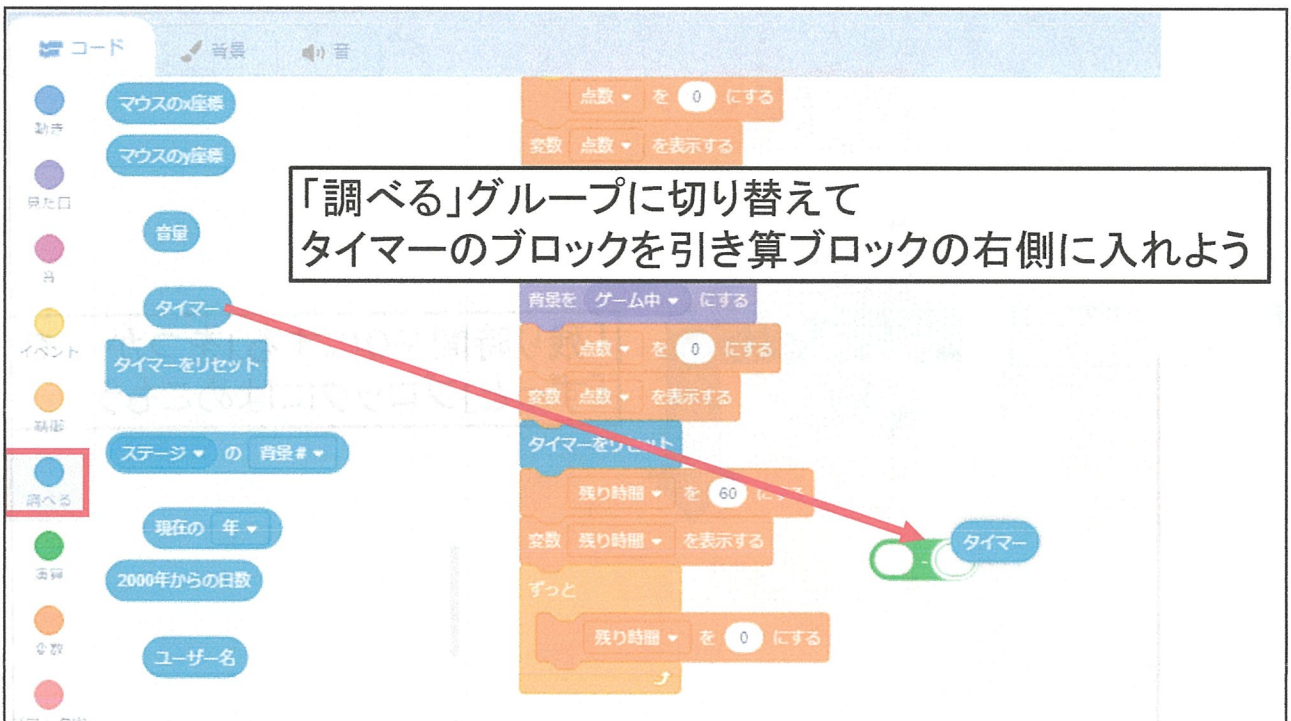
5



6



7



8

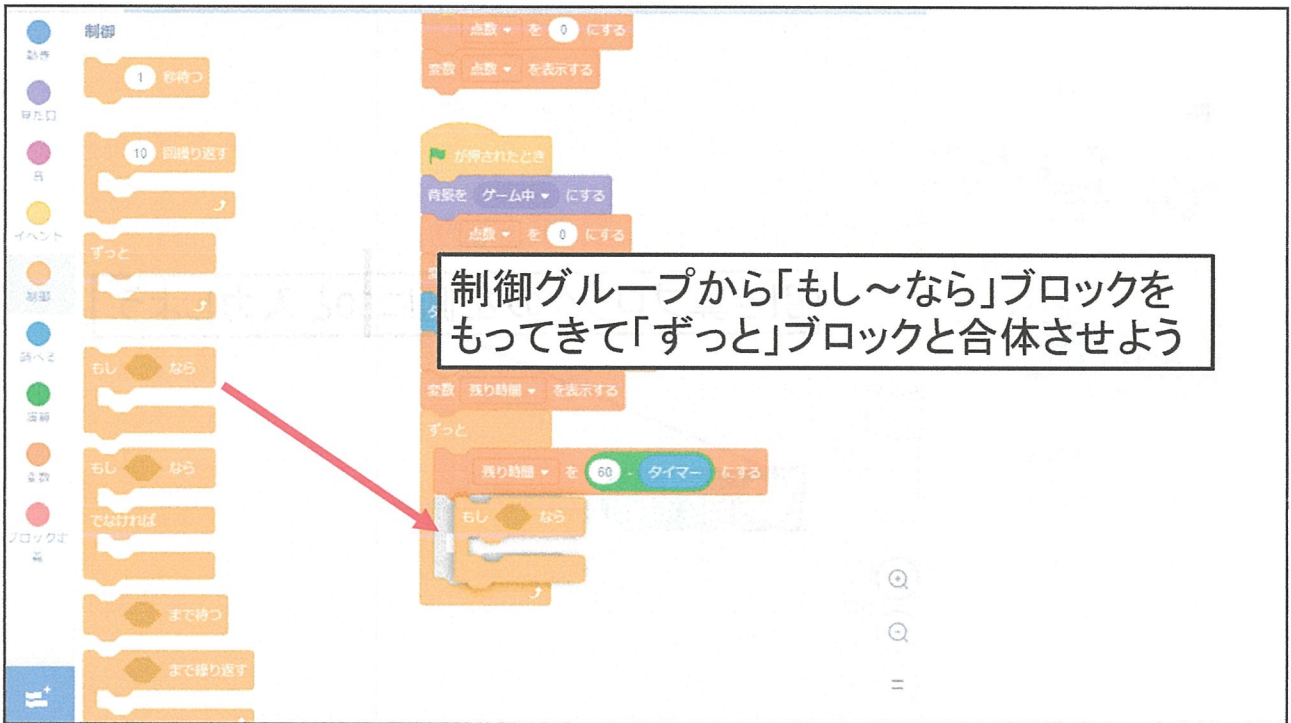
引き算ブロックの左側に60と入力しよう

9

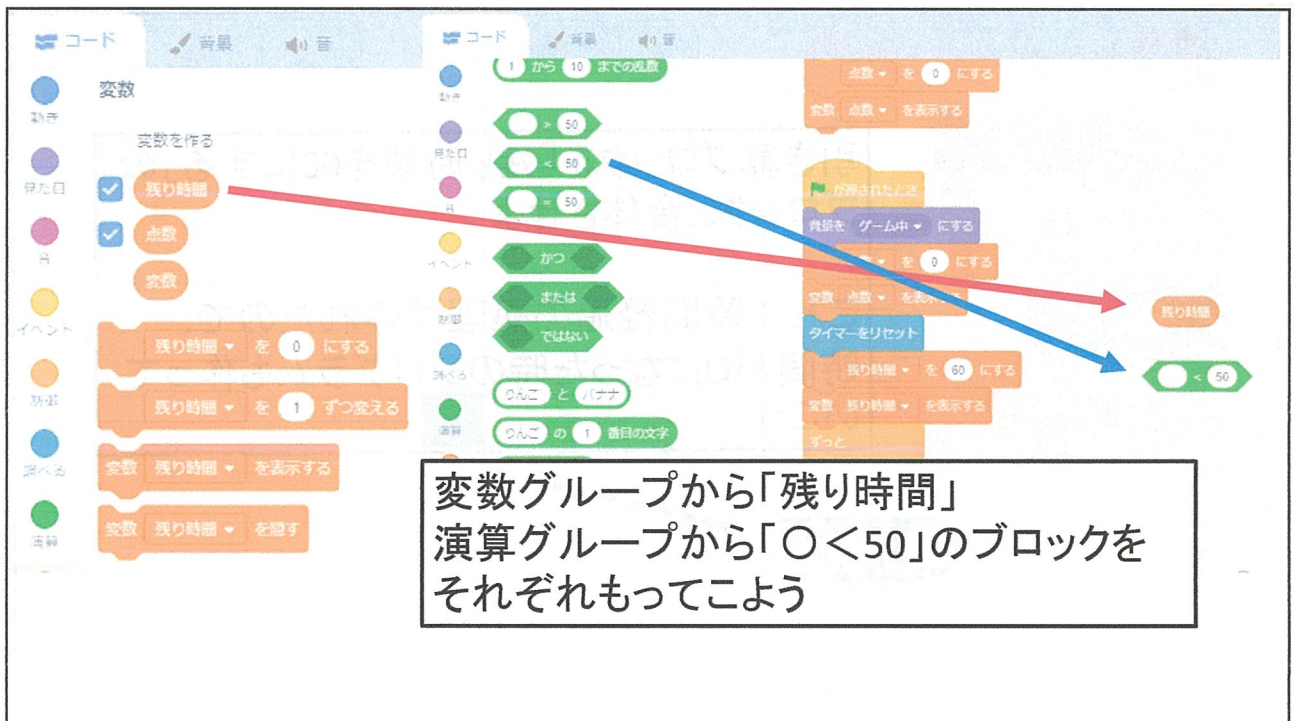
引き算ブロックを「残り時間を60にする」のブロックと合体させよう

これで時間経過の処理は作れたので、時間が0になった時のプログラムも作っていきましょう

10



11



12



「残り時間」と「0<50」のブロックを合体して「残り時間<0」に書きかえよう

13

「もし～なら」のブロックに演算ブロックを入れよう

14

残り時間が0より小さくなったらどうするかを追加する  
 イベントグループから「ゲームオーバーを送って待つ」をもってきてもしブロックに入れよう

タイマーをリセット  
 残り時間 を 60 にする  
 変数 残り時間 を表示する  
 ずっと  
 残り時間 を 60 - タイマー にする  
 もし 残り時間 < 0 なら  
 ゲームオーバーを送って待つ

15

見た目グループに切り替えて「背景をゲームオーバーにする」を入れよう

背景を ゲームオーバー にする  
 背景を ゲームオーバー にして待つ  
 次の背景にする  
 色 の効果を 25 ずつ変える  
 色 の効果を 0 にする  
 画像効果をなくす  
 背景の 番号  
 終わるまで ポップ の音を鳴らす  
 ポップ の音を鳴らす  
 すべての音を止める

背景を ゲーム中 にする  
 点数 を 0 にする  
 変数 点数 を表示する  
 タイマーをリセット  
 残り時間 を 60 にする  
 変数 残り時間 を 60 にする  
 ずっと  
 残り時間 を 60 - タイマー にする  
 もし 残り時間 < 0 なら  
 ゲームオーバーを送って待つ  
 背景を ゲームオーバー にする

16

変数グループに切り替えて「変数残り時間を隠す」を入れよう

変数 残り時間 を表示する

変数 残り時間 を 0 にする

変数 残り時間 を 1 ずつ変える

変数 残り時間 を表示する

変数 残り時間 を隠す

変数 残り時間 を 60 - タイマー にする

もし 残り時間 < 0 なら

ゲームオーバーを送って待つ

背景を ゲームオーバー にする

変数 残り時間 を隠す

17

制御グループに切り替えて「すべてを止める」を入れよう

変数 点数 を表示する

変数 残り時間 を表示する

タイマーをリセット

60 にする

ずっと

残り時間 を 60 にする

もし 残り時間 < 0 なら

ゲームオーバーを送って待つ

背景を ゲームオーバー にする

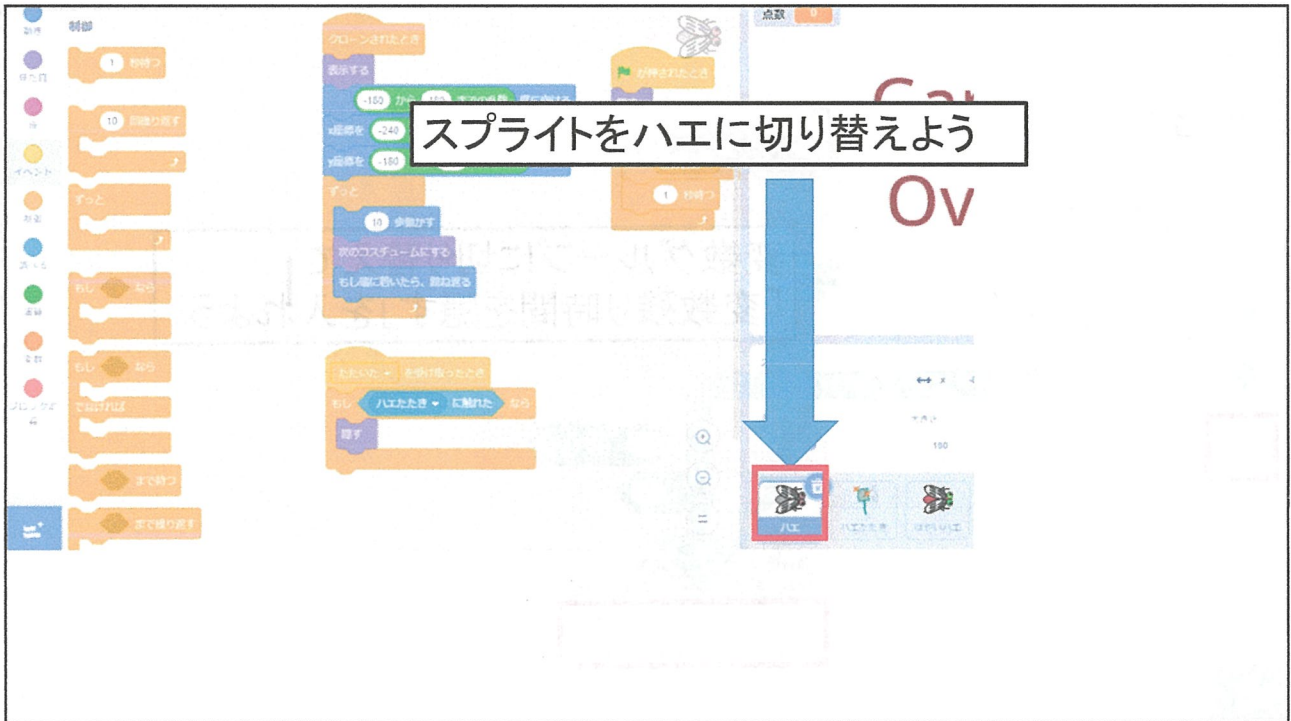
変数 残り時間 を隠す

すべてを止める

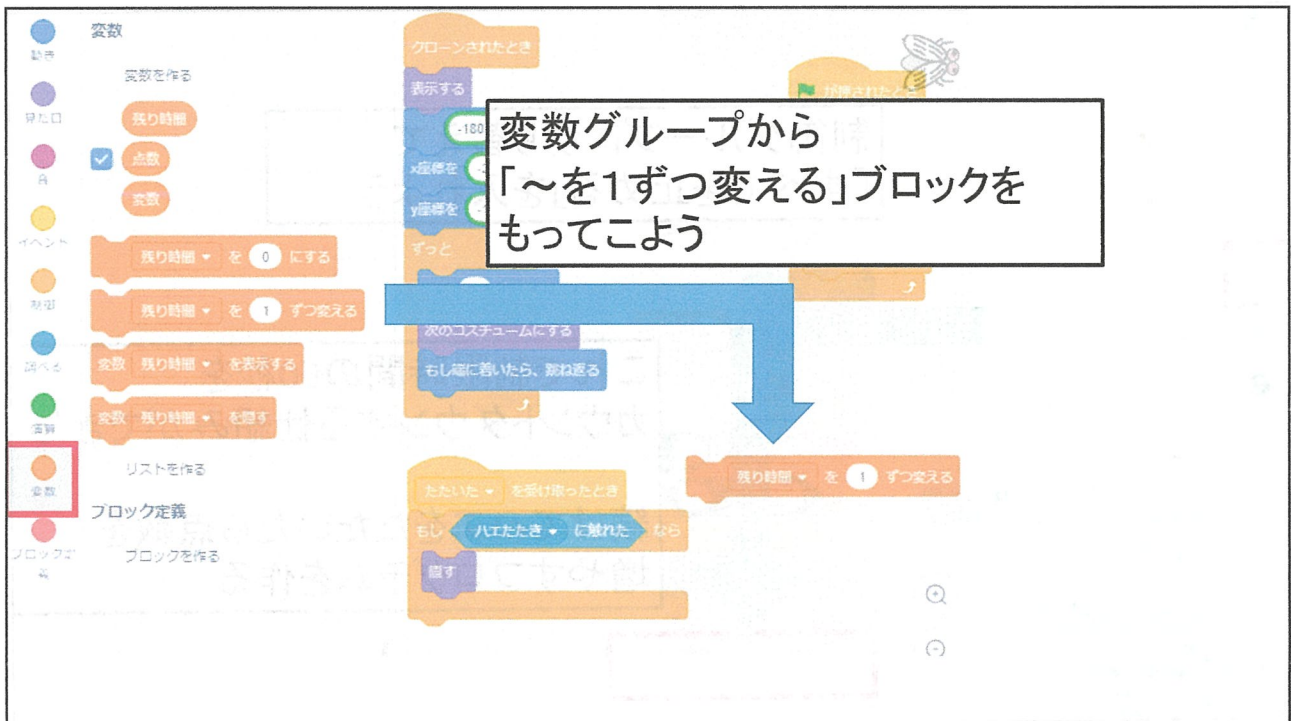
これで制限時間の60秒を  
カウントダウンする仕組みができた

続けてハエをたたいたら点数を  
増やすプログラムを作る

18



19



20



次のコスチュームにする

もし端に着いたら、跳ね返る

たたいたを受け取ったとき

もし ハエたたき に触れた なら

点数 を 10 ずつ変える

隠す

ふつうのハエは1匹たたくごとに10点ずつ増やしたいので、「点数を10ずつ変える」に変更しよう

23

クローンされたとき

表示する

x座標を -180 から 180 までの乱数 度に向ける

y座標を -240 から 340 までの乱数 にする

y座標を -180 から 180 までの乱数 にする

ずっと

50 歩動かす

次のコスチュームにする

もし端に着いたら、跳ね返る

たたいたを受け取ったとき

もし ハエたたき に触れた なら

隠す

スプライトを「はやいハエ」に切り替えよう

はやいハエもたたかれた時に点数を増やしたいので、同じプログラムを追加する

24

変数グループから「～を1ずつ変える」ブロックをもってきて「ハエたたきに触れたなら」のブロックと合体

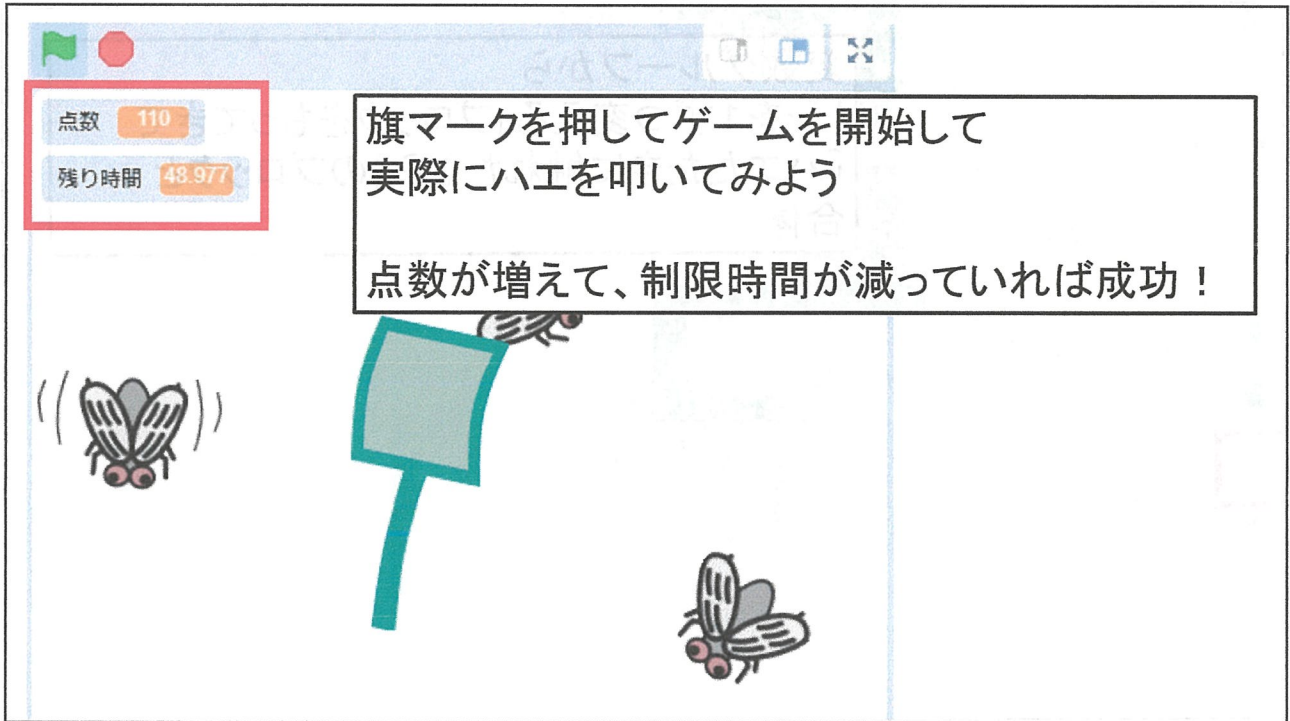
変数グループから「～を1ずつ変える」ブロックをもってきて「ハエたたきに触れたなら」のブロックと合体

25

▼から「点数」を選択して30ずつ増えるように変更しよう

▼から「点数」を選択して30ずつ増えるように変更しよう

26



27