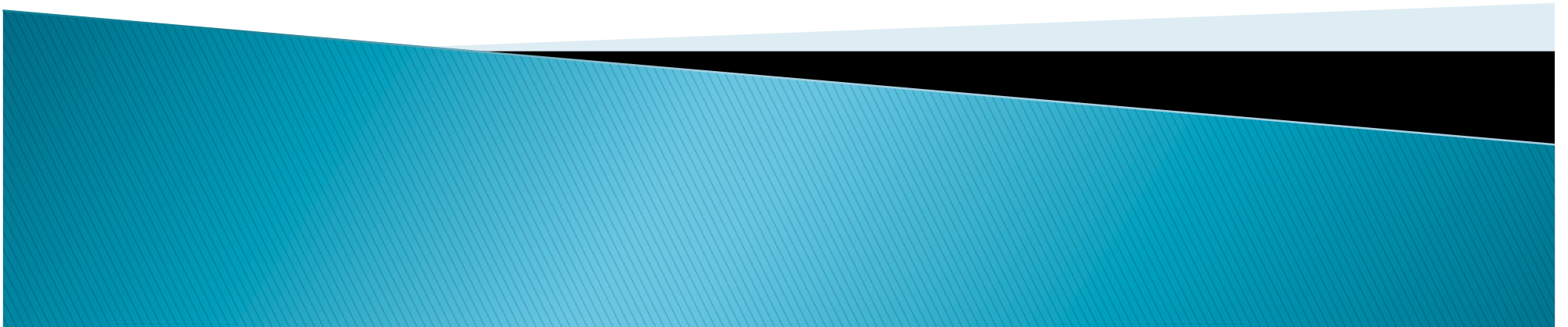


Scratch

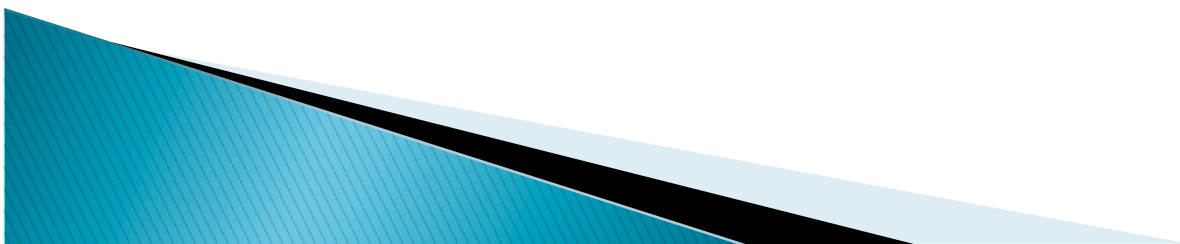
オブジェクト指向プログラミング



オブジェクト指向プログラミング

Object Oriented Programming(OOP)

- ▶ プログラミングの考え方
 - プログラム全体を「オブジェクト」の組み合わせで作成
- ▶ オブジェクト
 - データ+その操作プログラム(メソッド)
 - プログラム中で1つの役割を担う
- ▶ プログラム全体はオブジェクト間の“やりとり”で動作
 - やりとり=メソッドの呼び出し



ScratchにおけるOOP

- ▶ オブジェクト＝スプライト
 - データ＝ローカル変数
 - プログラム＝スクリプト
 - メソッド呼び出し?

「送る」と「受け取った時」を利用



サンプル(1)

- ▶ 猫が犬にぶつかると犬が移動を開始するプログラム
- ▶ 猫と犬の sprites を用意して、以下のように配置



サンプル(1)

- ▶ 以下のスクリプトを入力して実行

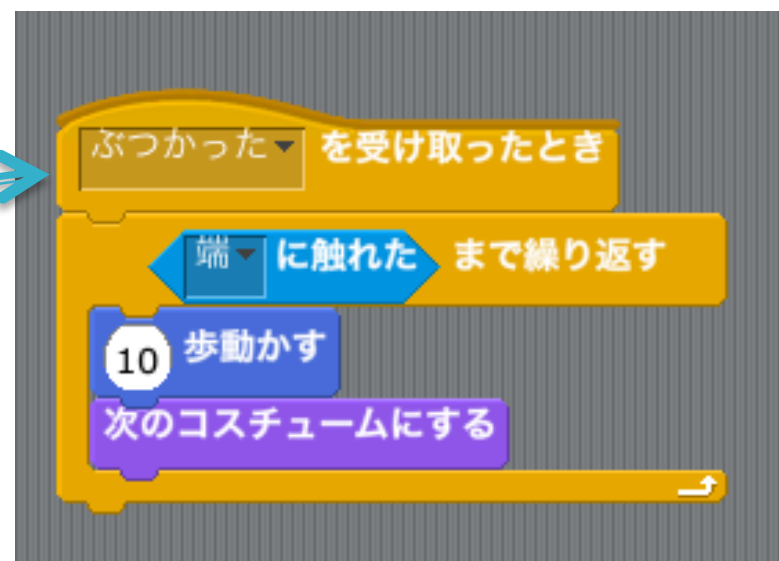
猫のスクリプト



The script for the cat character is as follows:

- When clicked (がクリックされたとき)
- Repeat until (いぬ に触れた) until (まで繰り返す)
- Move 10 steps (10 歩動かす)
- Change to next costume (次のコスチュームにする)
- Wait 0.1 seconds (0.1 秒待つ)
- Send message (ぶつかった を送る)

犬のスクリプト



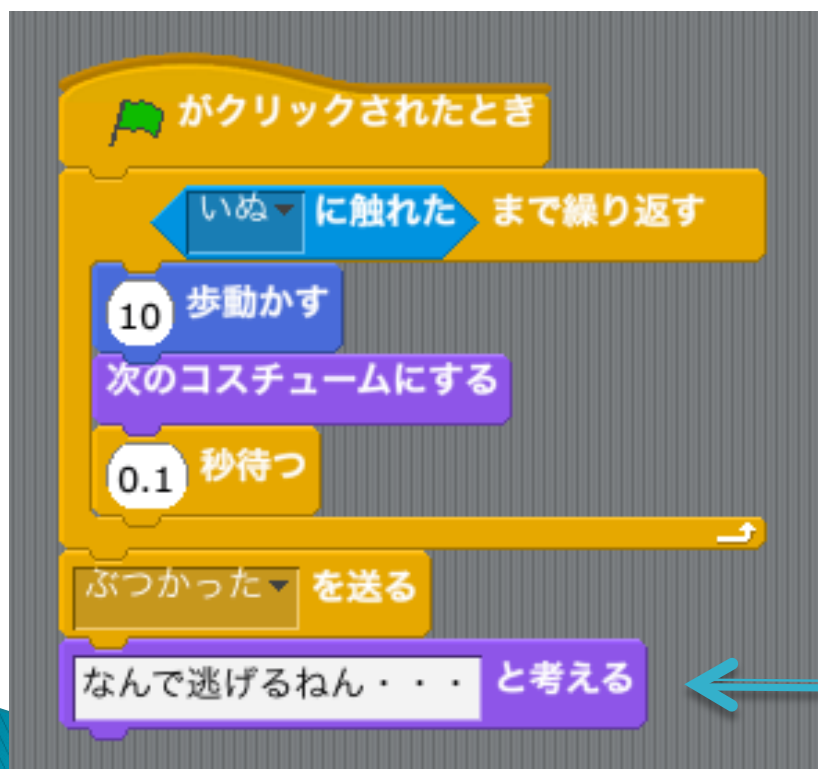
The script for the dog character is as follows:

- When received message (ぶつかった を受け取ったとき)
- Repeat until (端 に触れた) until (まで繰り返す)
- Move 10 steps (10 歩動かす)
- Change to next costume (次のコスチュームにする)

サンプル(1-a)

- ▶ 以下のように猫のスク립トを変更

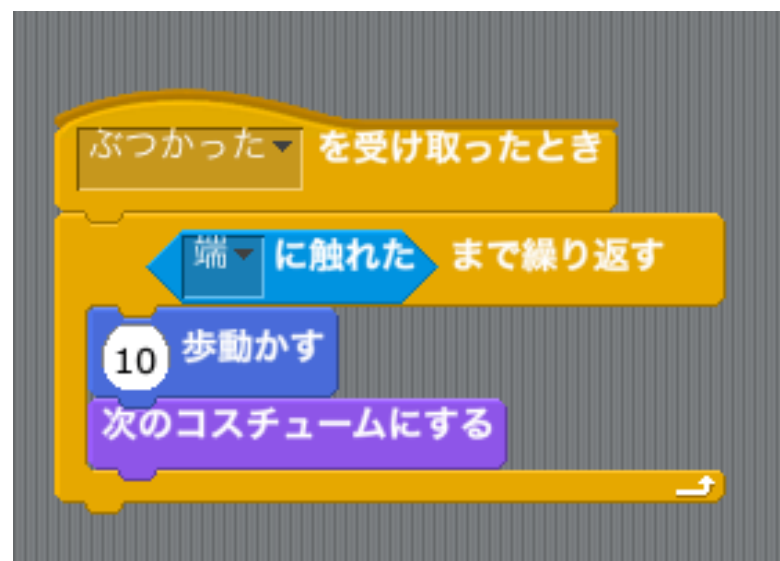
猫のスク립ト



The script for the cat character is as follows:

- When clicked (がクリックされたとき)
- Repeat until (いぬに触れた) until (まで繰り返す)
- Move 10 steps (10 歩動かす)
- Change to the next costume (次のコスチュームにする)
- Wait 0.1 seconds (0.1 秒待つ)
- Send a message (ぶつかった) (ぶつかった を送る)
- Think (なんで逃げるねん・・・) (と考える)

犬のスク립ト



The script for the dog character is as follows:

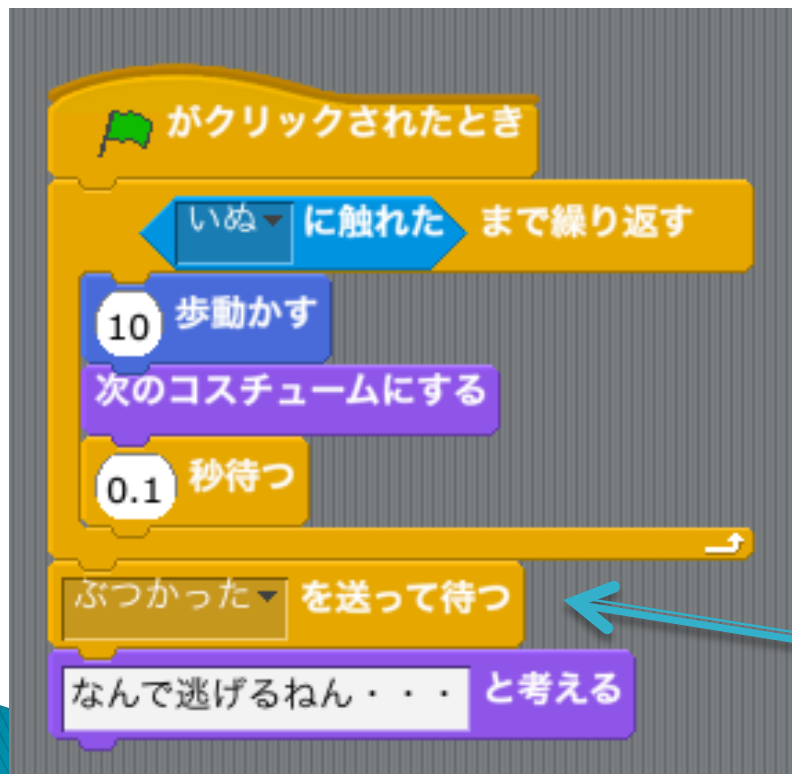
- When received a message (ぶつかった) (ぶつかった を受け取ったとき)
- Repeat until (端に触れた) until (まで繰り返す)
- Move 10 steps (10 歩動かす)
- Change to the next costume (次のコスチュームにする)

これを追加

サンプル(1-b)

- ▶ さらに変更して1-aのプログラムと比較してみる

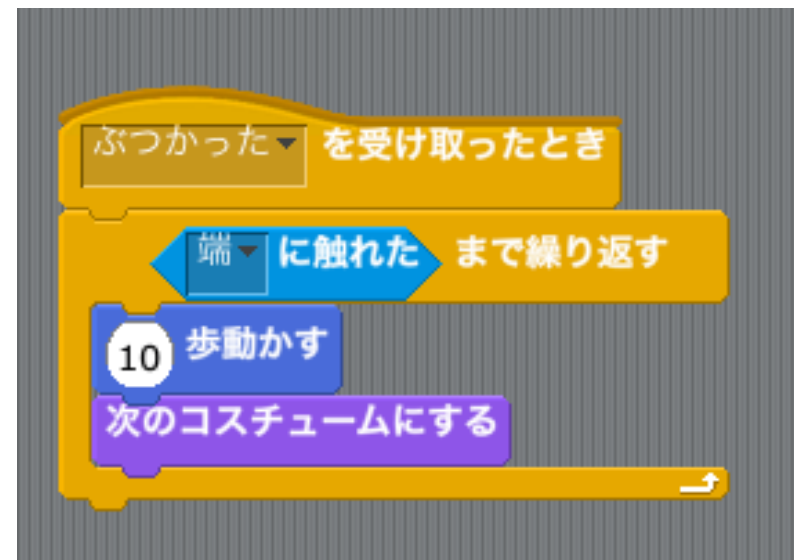
猫のスク립ト



The script for the cat character starts with a 'when clicked' event block. It contains a loop that repeats until it touches the 'dog' sprite. Inside the loop, the cat moves 10 steps, changes to the next costume, and waits for 0.1 seconds. After the loop, it sends a 'bumped' message and says 'Why are you running away...'.

```
がクリックされたとき  
いぬ に触れた まで繰り返す  
10 歩動かす  
次のコスチュームにする  
0.1 秒待つ  
ぶつかった を送って待つ  
なんで逃げるねん... と考える
```

犬のスク립ト



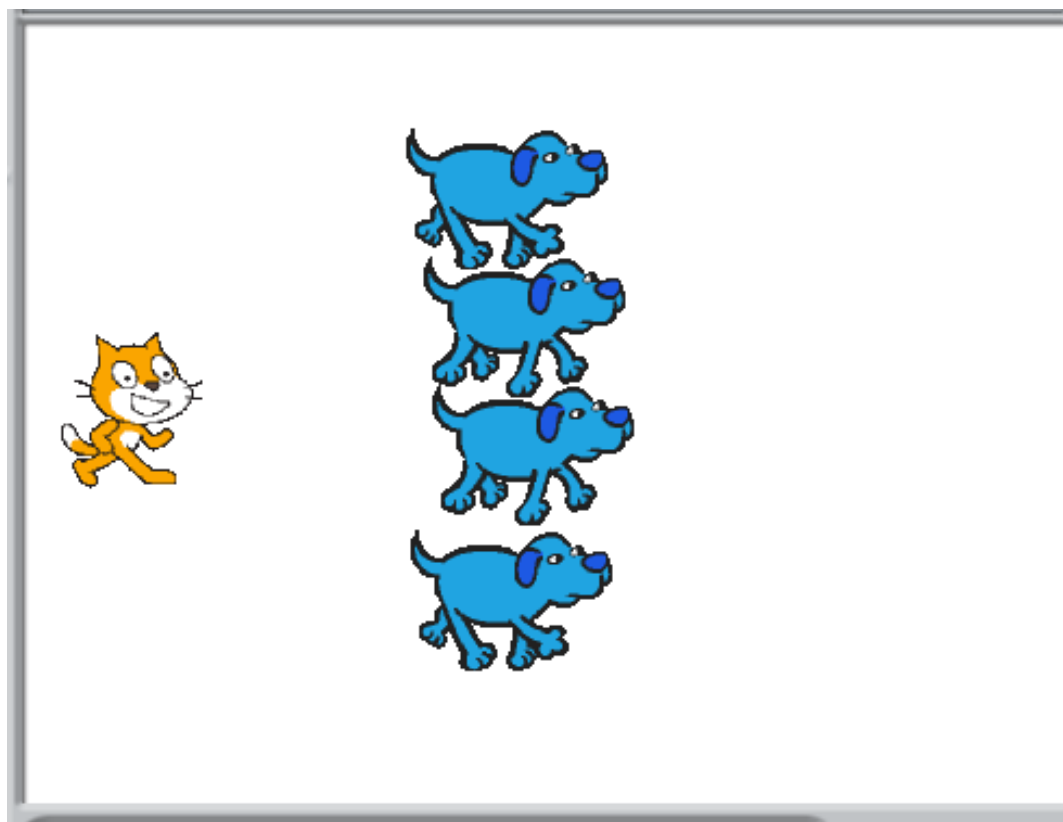
The script for the dog character starts with a 'when bumped' event block. It contains a loop that repeats until it touches the 'end' sprite. Inside the loop, the dog moves 10 steps and changes to the next costume.

```
ぶつかった を受け取ったとき  
端 に触れた まで繰り返す  
10 歩動かす  
次のコスチュームにする
```

ここを変更

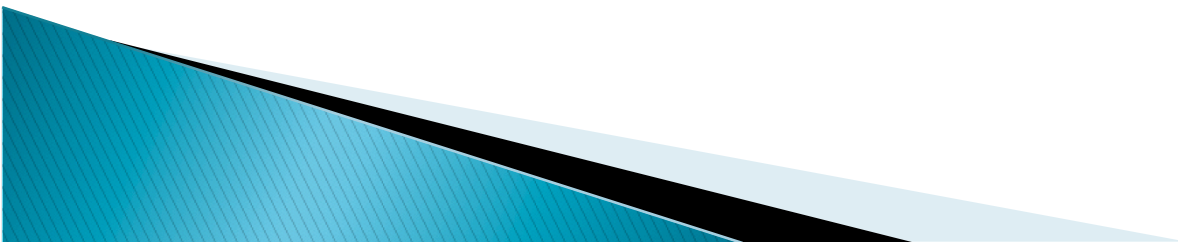
サンプル(1-c)

- ▶ 犬を複製してみる(スクリプトは変更しない)



Scratchの「送る」の注意

- ▶ 「送る」と「送って待つ」の違い(1-aと 1-b)
 - 送る=送った後のスクリプトは送り先のスクリプトと平行実行
 - 送って待つ=送り先のスクリプトの終了を待つ
- ▶ 送り先の指定はできない(1-c)
 - 同じメッセージを複数のスプライトが受け取ることができる



サンプル(2)

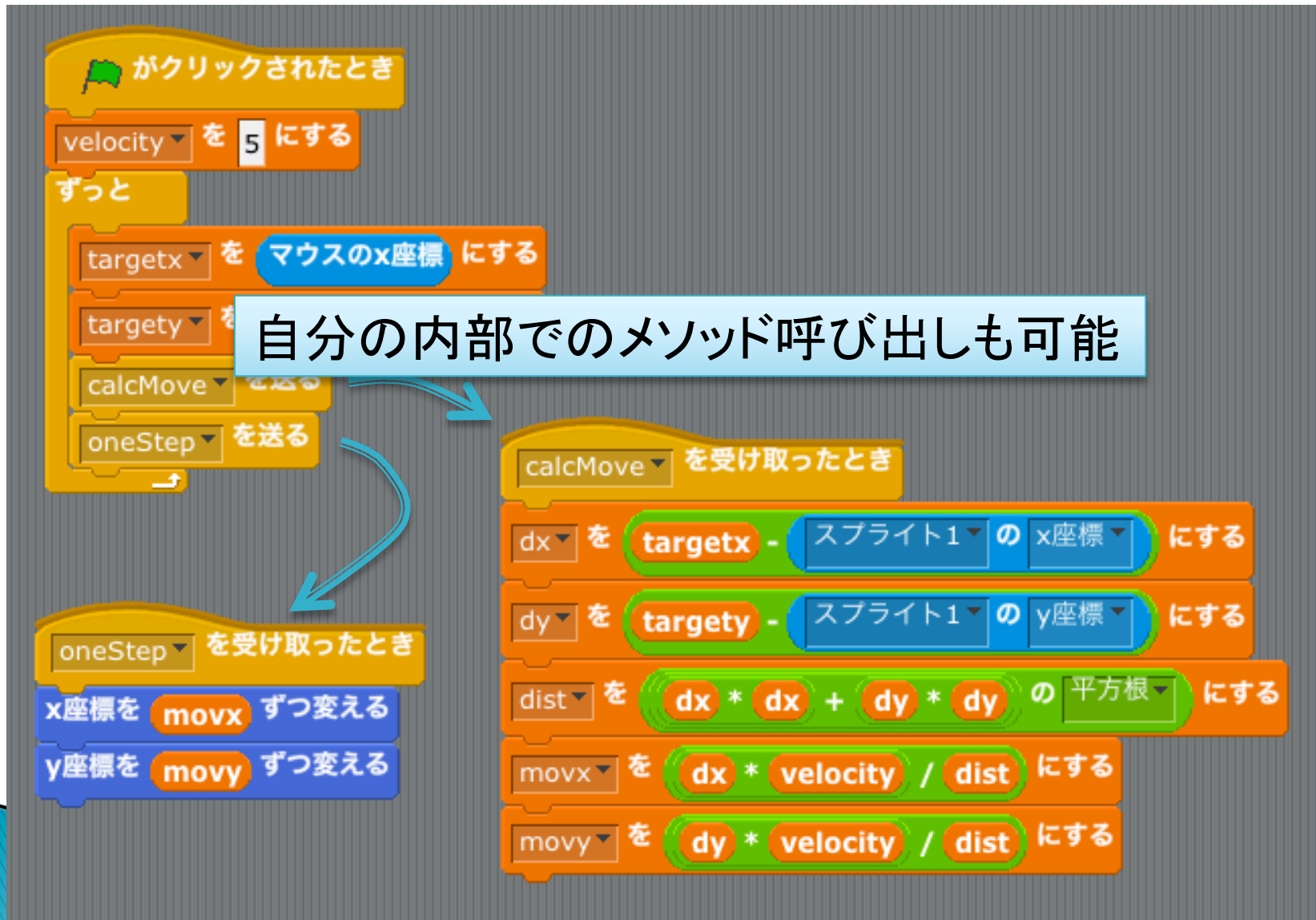
- ▶ マウスを追いかける猫(メソッド呼び出し版)



サンプル(2)

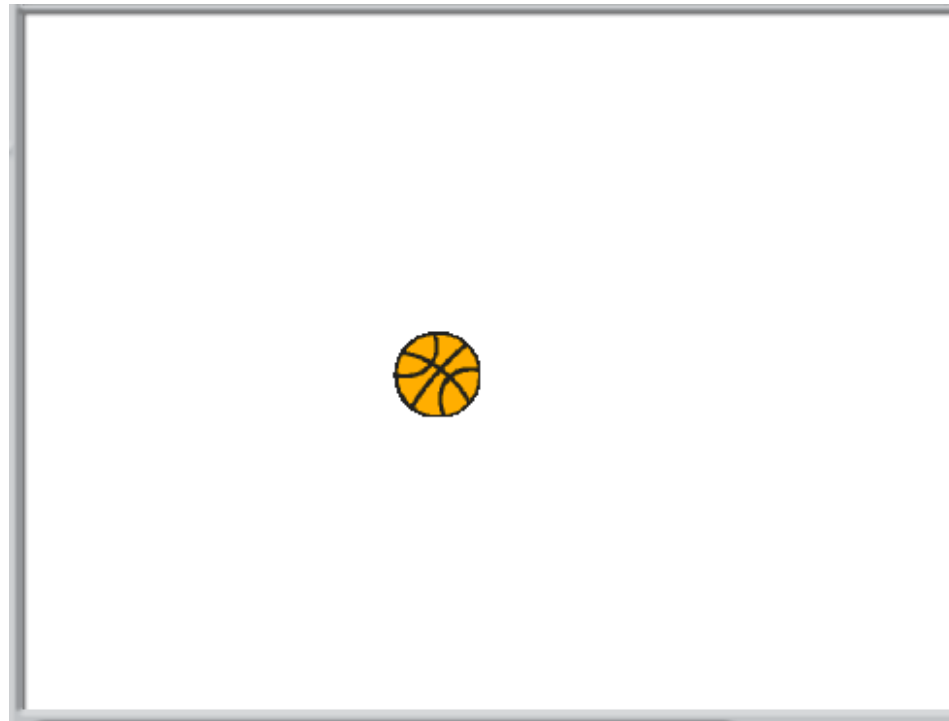


サンプル(2)

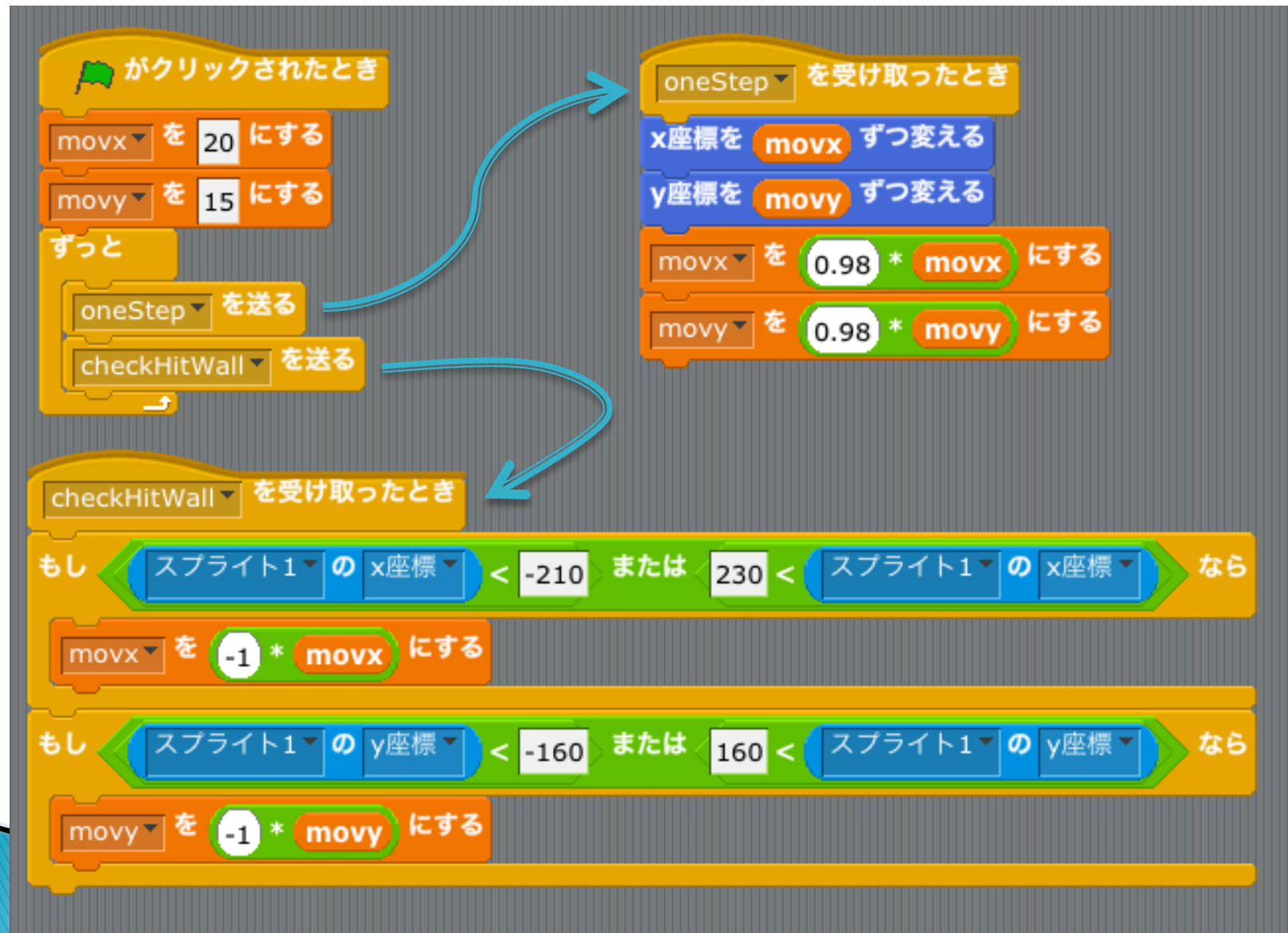


サンプル(3)

- ▶ 壁に当たると跳ね返るボール
- ▶ 以下のようにスプライトを用意

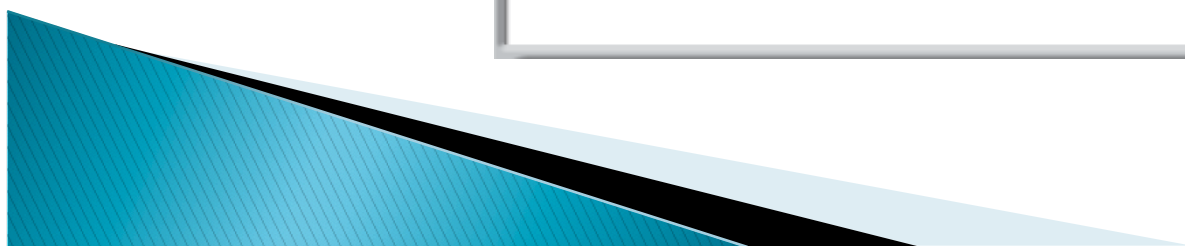
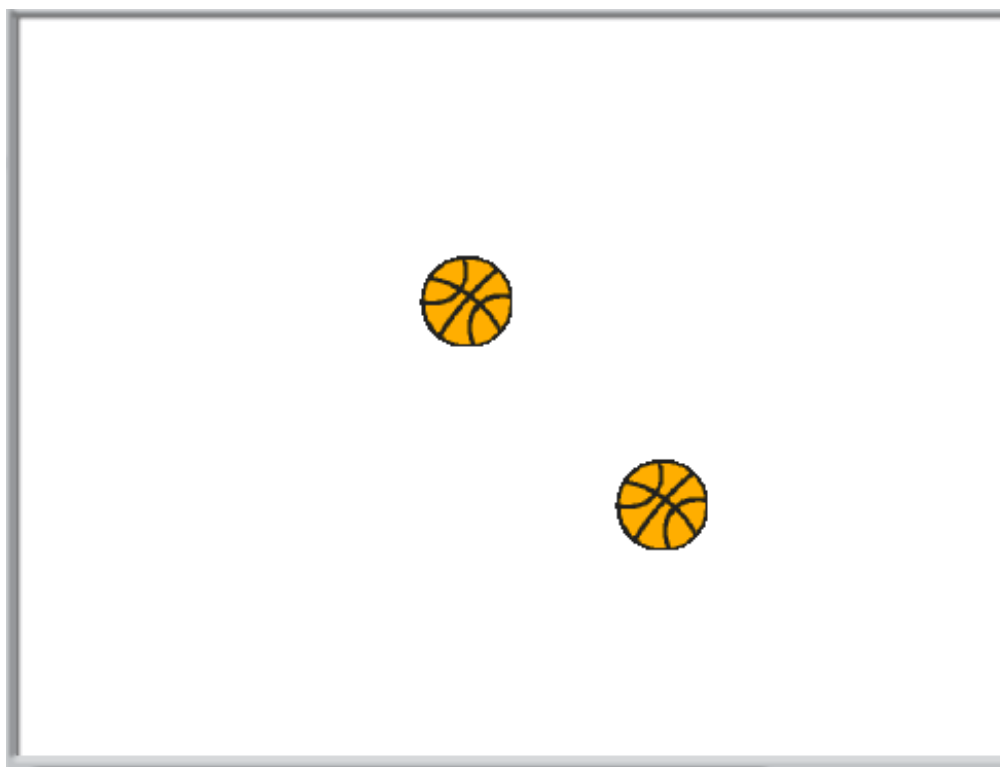


サンプル(3)



サンプル(3-a)

- ▶ スプライトを複製
- ▶ 複製されたスプライトのスクリプトは次ページ



サンプル(3-a)

The image shows a Scratch script with four event-driven blocks:

- がクリックされたとき** (When clicked):
 - movx を 10 にする (Set movx to 10)
 - movy を 5 にする (Set movy to 5)
- oneStep を受け取ったとき** (When oneStep received):
 - x座標を movx ずつ変える (Change x coordinate by movx)
 - y座標を movy ずつ変える (Change y coordinate by movy)
 - movx を $0.98 * \text{movx}$ にする (Set movx to 0.98 * movx)
 - movy を $0.98 * \text{movy}$ にする (Set movy to 0.98 * movy)
- checkHitWall を受け取ったとき** (When checkHitWall received):
 - もし $\text{スプライト2のx座標} < -210$ または $230 < \text{スプライト2のx座標}$ なら (If x coordinate is less than -210 or 230 is less than x coordinate, then...)
 - movx を $-1 * \text{movx}$ にする (Set movx to -1 * movx)
 - もし $\text{スプライト2のy座標} < -160$ または $160 < \text{スプライト2のy座標}$ なら (If y coordinate is less than -160 or 160 is less than y coordinate, then...)
 - movy を $-1 * \text{movy}$ にする (Set movy to -1 * movy)

サンプル(3-a)

The image shows a Scratch script with the following blocks:

- がクリックされたとき** (When clicked):
 - movx を 10 にする (Set x to 10)
 - movy を 5 にする (Set y to 5)
- oneStep を受け取ったとき** (When oneStep is received):
 - x座標を movx ずつ変える (Change x by movx)
 - y座標を movy ずつ変える (Change y by movy)
 - movx を 1 にする (Set movx to 1)
 - movy を 0.98 にする (Set movy to 0.98)
- checkHitWall を受け取ったとき** (When checkHitWall is received):
 - もし < スプライト2 の x座標 < -210 または 230 < スプライト2 の x座標 > > なら (If x is less than -210 or 230 is less than x):
 - movx を $-1 * \text{movx}$ にする (Set movx to -1 times movx)
 - もし < スプライト2 の y座標 < -160 または 160 < スプライト2 の y座標 > > なら (If y is less than -160 or 160 is less than y):
 - movy を $-1 * \text{movy}$ にする (Set movy to -1 times movy)

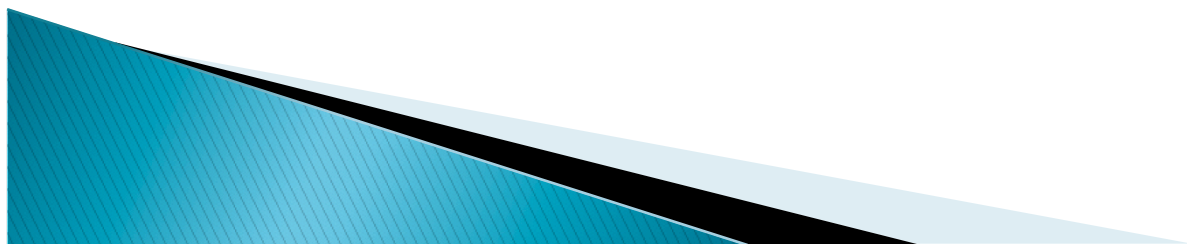
A blue dashed box highlights the initial 'movx' and 'movy' blocks. A blue callout box with the text 'ここが違う' (This is different) has arrows pointing to the 'movx' and 'movy' blocks in the 'oneStep' event, the 'movx' block in the 'checkHitWall' event, and the 'movy' block in the 'checkHitWall' event.

サンプル(3-a)

- ▶ 複製されたスプライトには「繰り返し」や「メソッドの呼び出し(送る)」が無くても良い



- ▶ オリジナルのスプライトが行うから



課題

「マウスを追いかける猫を追いかける**複数**の犬」プログラムを作成せよ

※ スプライトは「猫」と「犬」でなくても、動作が実現されていれば良い。

[要件]

1) サンプル(2)を参考に「送る」と「受け取る」をどこかで用いること

※ 猫についてはサンプル(2)をそのまま使っても良い

2) どれかの犬が猫を捕まえたら全ての犬は止まるようにする

※ 何かのキーを押したら再び動くようになるとなお良い

[提出期限]

2009年12月 4日(金)17:00

[提出先]

情報科学演習III ホームページの課題提出フォームより提出

