


# 今後の予定の変更

- 当初予定
    - オブジェクト指向プログラミング
    - イベント(キーボードとマウス)
    - インタラクティブアニメ
    - メッセージパッシング
  - 今後の予定
    - イベント(キーボードとマウス)
    - オブジェクト指向プログラミング + メッセージパッシング
    - インタラクティブアニメ
  - 以降の予定には変更無し
- 今日はこれ
- 

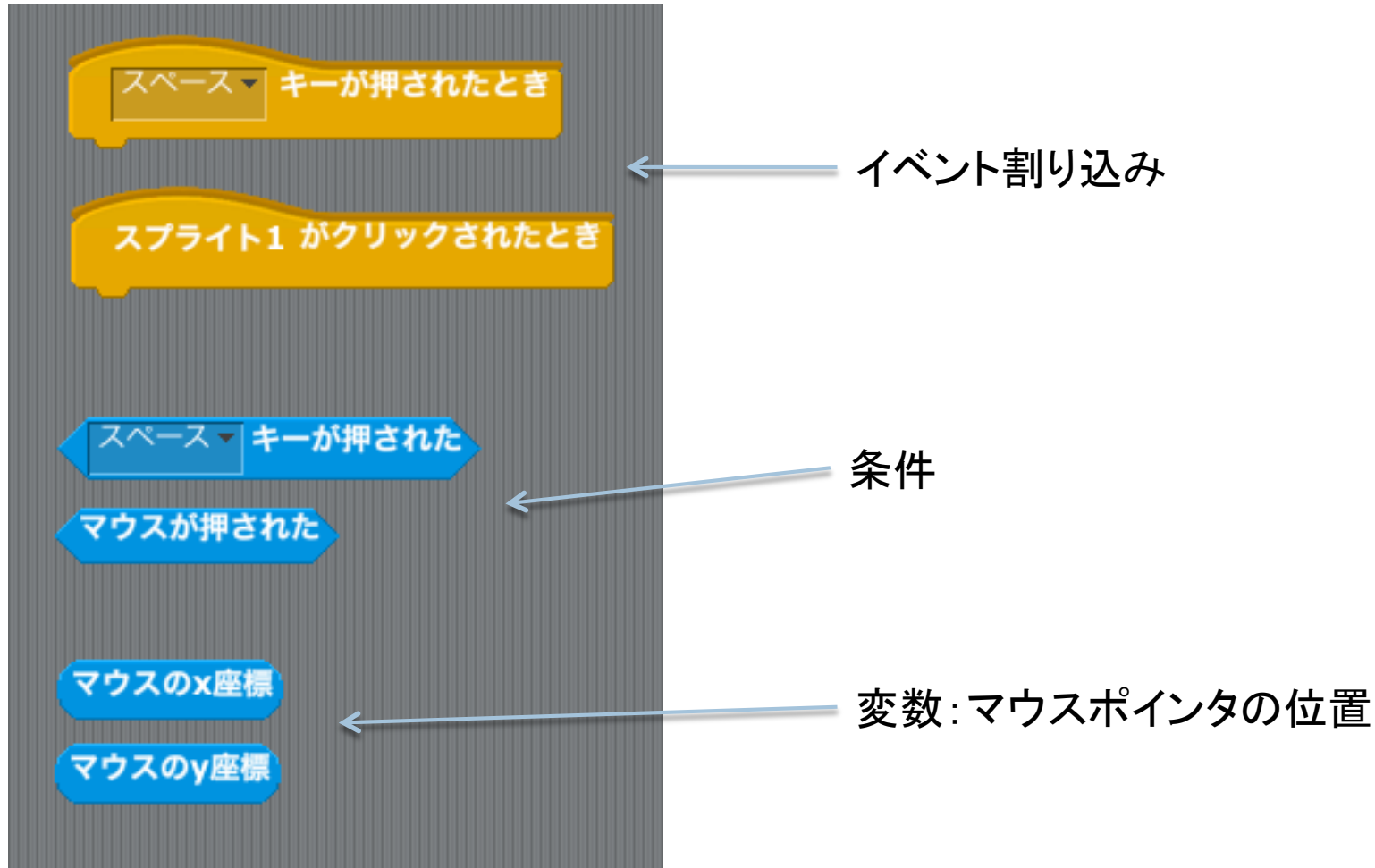
# SCRATCH

(3) イベント(キーボードとマウス)

# イベント

- 注意すべき事象
    - キーボード(何かキーが叩かれた)
    - マウス(マウスボタンが押された, マウスが移動した)
    - タイマー(時間になった)
    - エラー発生
- などなど

# 今回扱うイベント関係の部品



# イベント割り込み



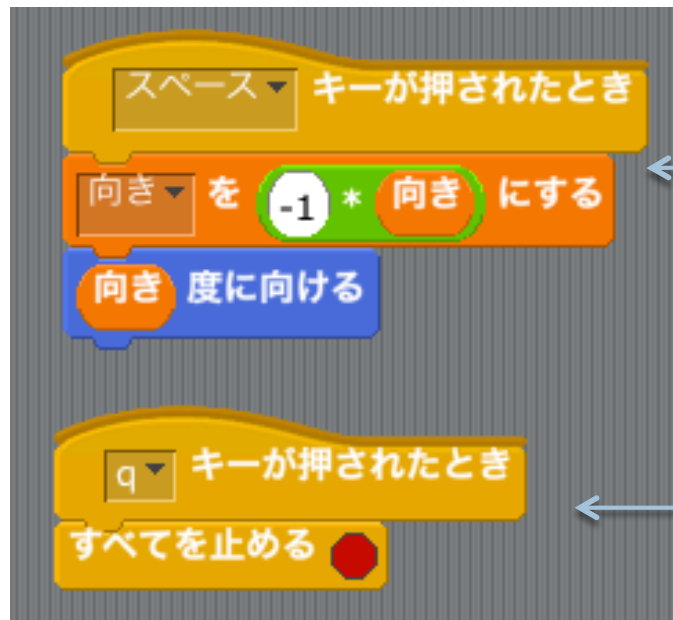
- イベント発生時に別のスクリプト部分を実行

# キーボード割り込みを使う

- 以下のプログラムを入力, 実行してみる
  - ▣ 注意) まず「向き」という変数を作る

The image shows two Scratch code snippets on a grey background. The left snippet is a 'when clicked' event block containing: 'set direction to 90', a 'forever' loop containing 'move 5 steps', 'go to next costume', and 'if reached edge, bounce back'. The right snippet is a 'when space key pressed' event block containing: 'set direction to -1 \* direction' and 'turn direction degrees', and a 'when q key pressed' event block containing: 'stop everything'.

# キーボード割り込み部分



← スペースキーが押されたら反転

← Qが押されたらプログラム終了

# マウス関係の割り込みも使う

- プログラムを以下のように変更してみる

The image displays four Scratch code blocks for mouse and keyboard events, arranged in a 2x2 grid. The top-left block is triggered by a mouse click and sets '向き' (direction) to 90, '考え中' (thinking) to 0, and includes a loop for moving 5 steps, changing costumes, and jumping back if at the edge. The top-right block is triggered by a click on 'Sprite 1' and toggles '考え中' (1 - thinking), with a conditional 'if thinking = 1' block containing 'うーん... と考える' (Uhn... thinking) and a 'do nothing' block. The bottom-left block is triggered by the space key and rotates the character by  $-1 \times$  '向き' (direction) and then '向き 度に向ける' (turn direction degrees). The bottom-right block is triggered by the 'q' key and uses a 'stop all' block.

```
Scratch Code Blocks:  
  
Block 1: がクリックされたとき  
  向き を 90 にする  
  考え中 を 0 にする  
  もし 考え中 = 0 なら、ずっと  
    5 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  
  →  
  
Block 2: スプライト1 がクリックされたとき  
  考え中 を 1 - 考え中 にする  
  もし 考え中 = 1 なら  
    うーん... と考える  
  でなければ  
    と考える  
  
Block 3: スペース キーが押されたとき  
  向き を -1 * 向き にする  
  向き 度に向ける  
  
Block 4: q キーが押されたとき  
  すべてを止める
```



# マウス座標を使う

- マウスポインタを追いかける猫を作成
  - 新規で別のプロジェクトを作成して下さい



# 使う部品

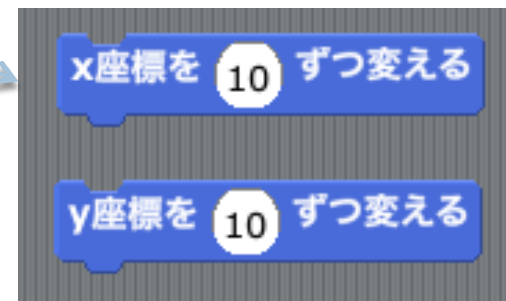
## □ 情報

- 自分の座標 (X位置, Y位置)
- マウスの座標 (X位置, Y位置)

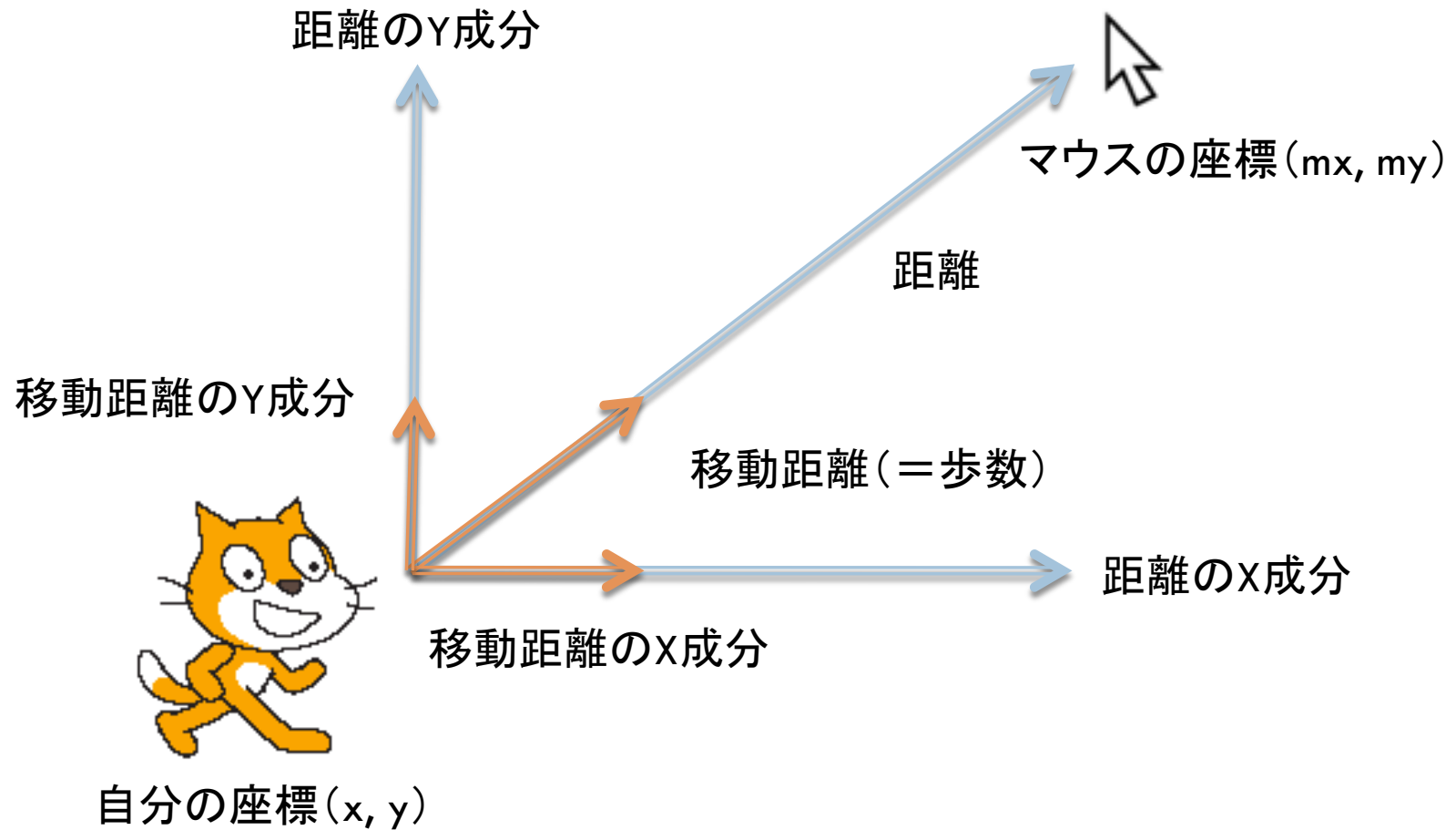


## □ 移動

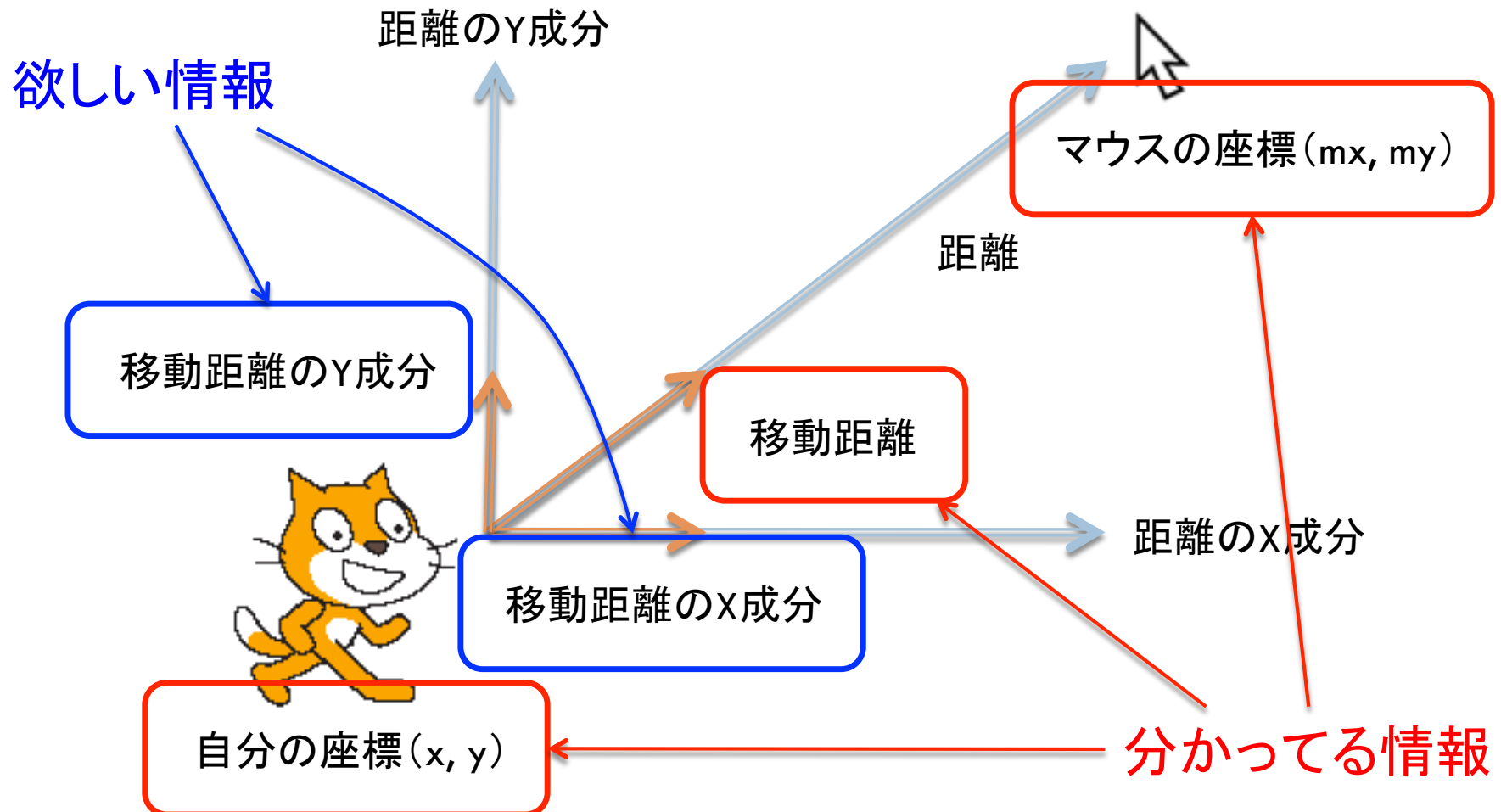
- 自分の座標を変更



# マウスの方向への移動(1)



# マウスの方向への移動(2)



# マウスの方向への移動(3)

距離 : 移動距離 = 距離のX成分 : 移動距離のX成分



距離 × 移動距離のX成分 = 移動距離 × 距離のX成分



移動距離のX成分 = 移動距離 × 距離のX成分 ÷ 距離

分かっている情報

距離のX成分とY成分から計算

「自分のX座標」と「マウスのX座標」から計算

※Y座標についても同様

# マウス方向への移動(4)

1. 距離のX成分 = マウスのX座標 - 自分のX座標
2. 距離のY成分 = マウスのY座標 - 自分のY座標
3. 距離 =  $\sqrt{\text{距離のX成分}^2 + \text{距離のY成分}^2}$
4. 移動距離のX成分 = 移動距離 × 距離のX成分 ÷ 距離
5. 移動距離のY成分 = 移動距離 × 距離のY成分 ÷ 距離

# プログラムに実現してみる

The image shows a Scratch script for calculating the distance from a mouse click to a character and moving the character towards it. The script is as follows:

```
がクリックされたとき
mov を 5 にする
ずっと
  dx を マウスのx座標 - x座標 にする
  dy を マウスのy座標 - y座標 にする
  dist を dx * dx + dy * dy の平方根 にする
  movx を mov * dx / dist にする
  movy を mov * dy / dist にする
  x座標を movx ずつ変える
  y座標を movy ずつ変える
  次のコスチュームにする
```

# 課題

「マウスを追いかける猫を追いかける犬」のプログラムを作成せよ  
(犬の sprites を追加し, 追加した犬の スクリプトを作成する)

※ sprites 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) Q キーを押すと犬は考えながら静止し, 再度 Q キーを押すと猫を追いかけるようになる

2) 猫を捕まえると犬は2秒間メッセージを表示する

※ 静止する必要はない

[提出期限]

2009年11月20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