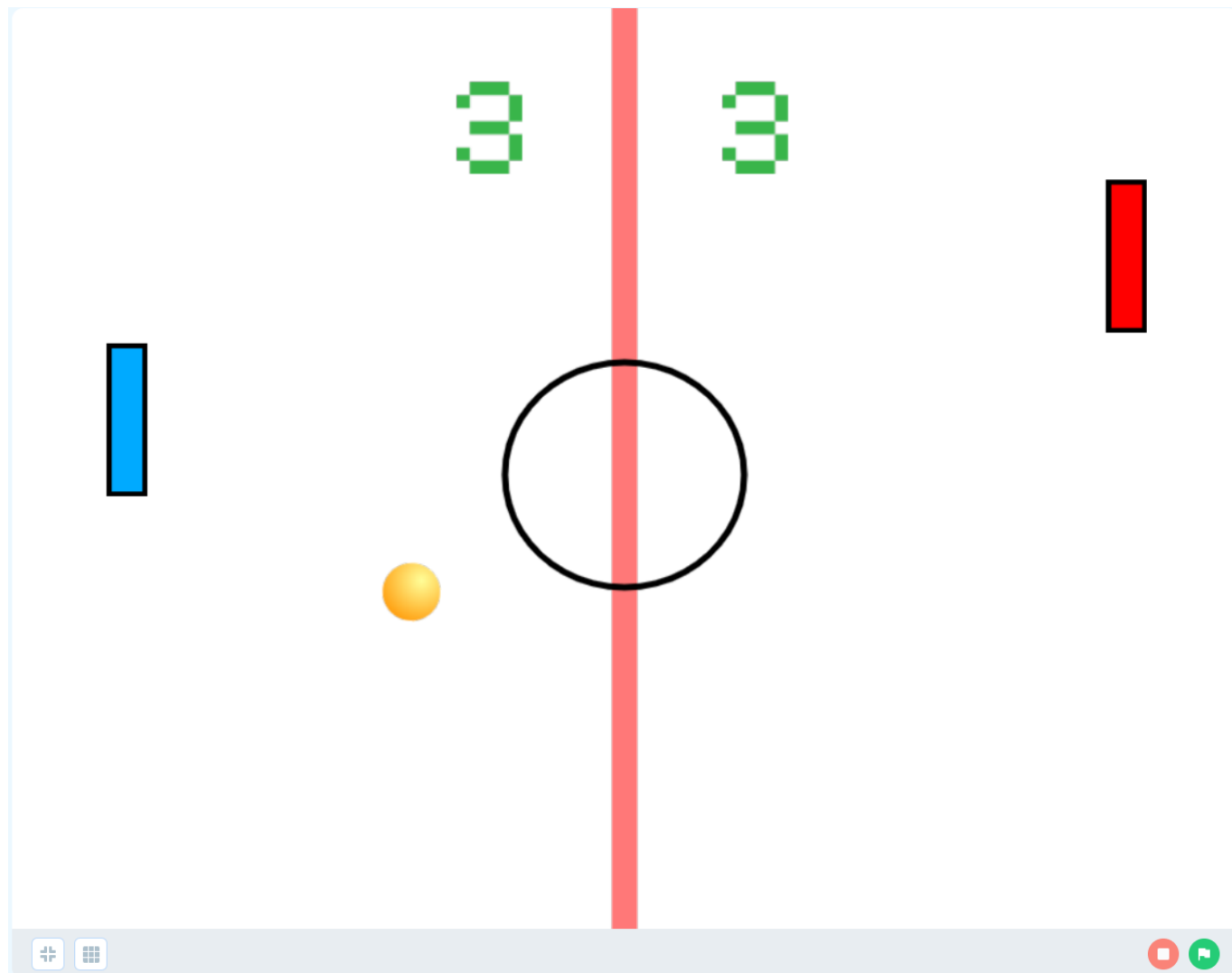


# 最終のテストにようこそ

---

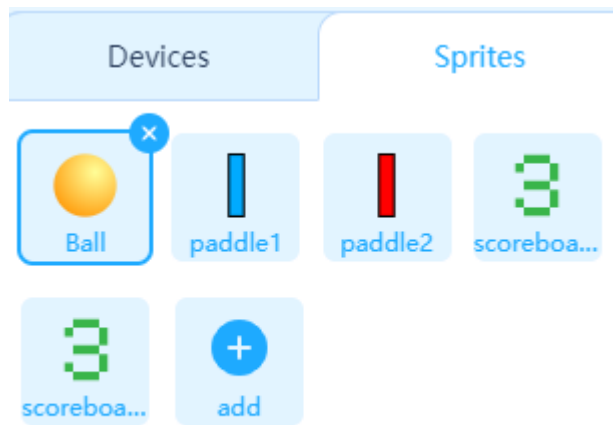
Python from Scratchの最終テストは**PONG**を作ることです。



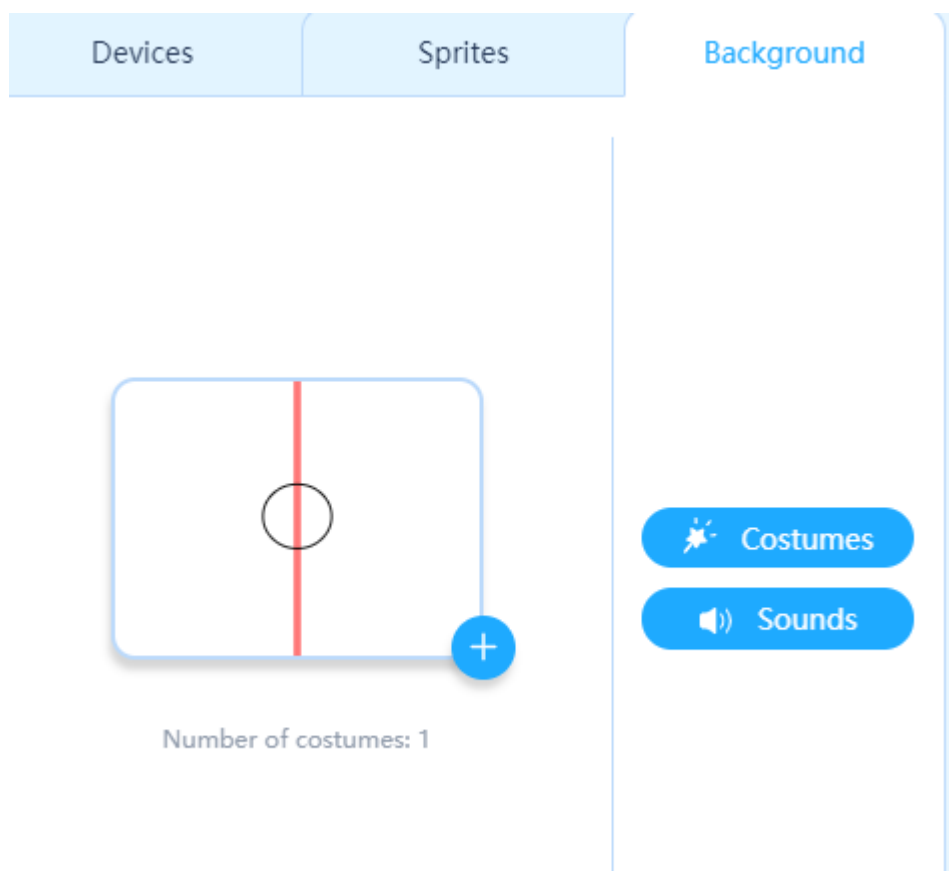
## 必要なもの

必要なものは以下のとおりです。

スプライト



## 背景




## 細かい部分

細かい部分も確認しましょう。

スコアのGUI(画面)


1



0

26 x 36


2



1

16 x 37


3



2

27 x 37


4



3

27 x 37


5



4

28 x 38


6



5

27 x 37


7



6

27 x 37


8




7

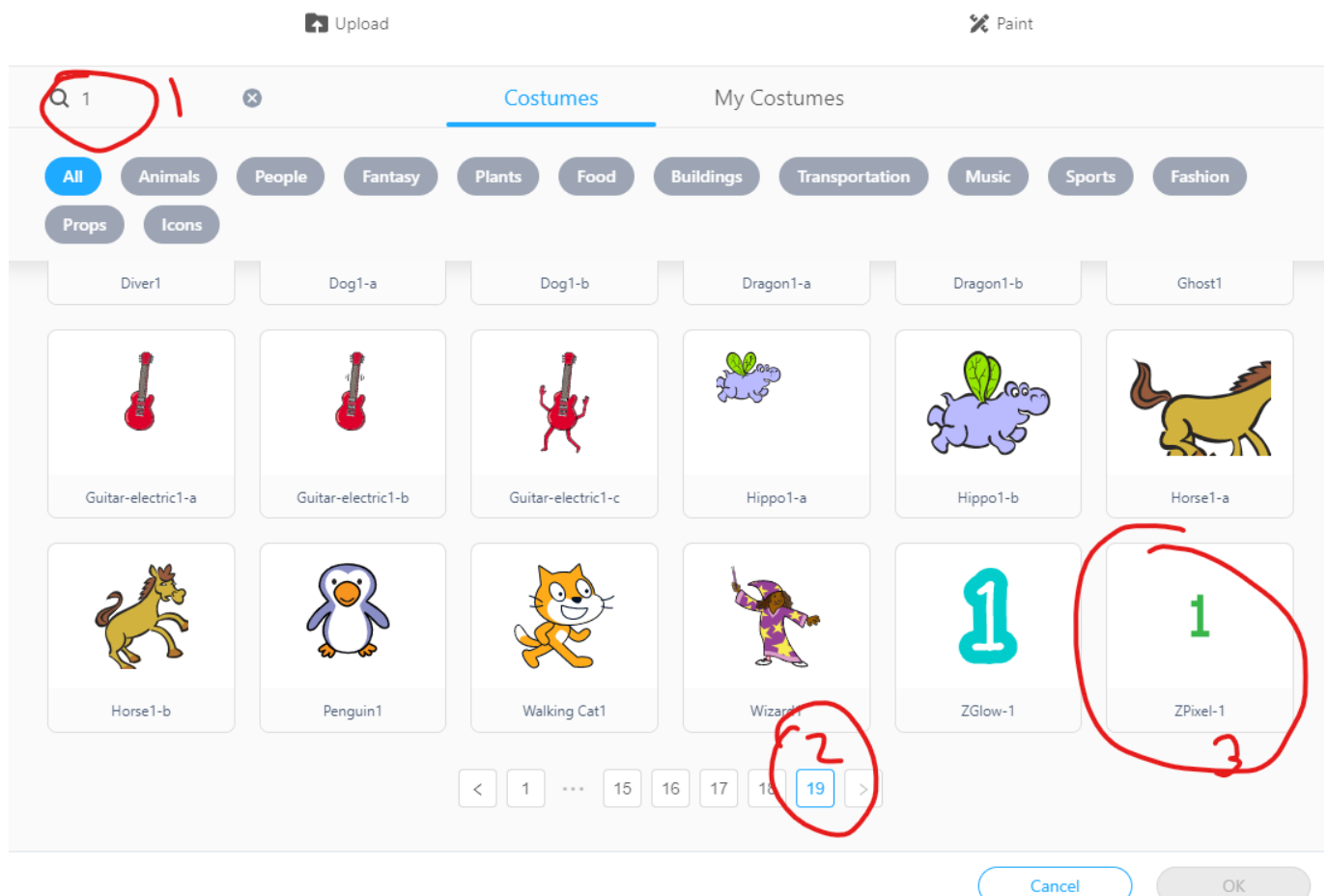
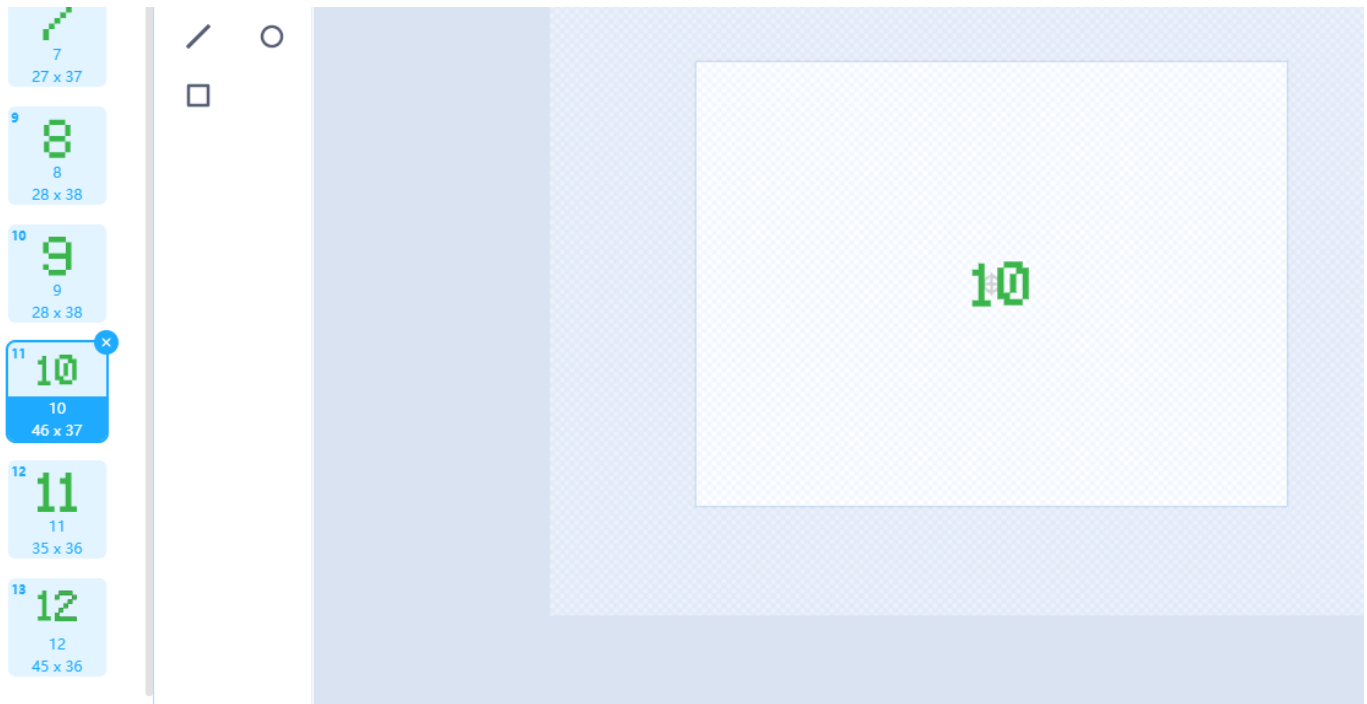
27 x 37

9



8





数字のスプライトの追加方法:

1. スプライトの中に新しいコスチュームを検索
2. 0～9を検索しましょう
3. 結果の最後のページをクリック
4. 最後のコスチューム4を選ぶ(ZPixel-#)
5. 自分で10～12を作る

## 全スプライトの必要なモジュール

各スプライトのスキプトの最初に書きましょう。



```
import random
from mBlock import event
import time
```



```
from mBlock import event
import time
```



```
from mBlock import event
import time
```

---

## 必要なプログラミングについて

### ボールの動き

1. 上と下のはしを触れたら跳ね返る
2. プレイヤーのパドルを通ったら、反対のプレイヤーのスコアは+ 1 点上がります！
3. 何秒一時停止後、ボールは真ん中に戻って、ランダムに右・左へ移動を始めます。
  1. `random.choice(list)` を使ってください
  2. `time.sleep(seconds)` も必要です
  3. 右左のはしに触れたら跳ね返らないように行動するべきです

### チャレンジ

2Pゲームを作りましょう

- 青は左、赤は右
- 青は `w` と `s` で上下に動く
- 赤は `up arrow` と `down arrow` で上下に動く
- どちらのプレイヤーは12点でできましたらゲームオーバーになります
- GUIでスコアを表示する

## アイディアマップ

効果音のプログラミング		
ゲームを実行して、「vs.2Pあるいはvs.CPU」のオプション	<b>ゲーム終了</b>	赤はCPU
BGMのプログラミング		

## 採点：合格点は80点です

**講師へ:** 下記のリストが採点基準です。1つ満たすごとに10点加点します。100点満点+ボーナスがあります。

- \_\_\_ 1. わからないところを自力で調べることができる
- \_\_\_ 2. プログラミングへの理解がある
- \_\_\_ 3. ゲームは全部の基本チャレンジに合致している
- \_\_\_ 4. プレイヤーの挙動に問題なく快適にプレイできる
- \_\_\_ 5. ゲームのタイミングが変ではない
- \_\_\_ 6. バグがない
- \_\_\_ 7. 発表のときにエラーがない
- \_\_\_ 8. 基本のPythonの文法を理解している
- \_\_\_ 9. スコアは問題なく表示されている
- \_\_\_ 10. ゲームオーバーの場合のプログラミングが問題ない
- \_\_\_ ボーナス 1 (アイディアマップ): 効果音を追加した(+5点)
- \_\_\_ ボーナス 2 (アイディアマップ): オプションがきちんと実装されている(+10点)
- \_\_\_ ボーナス 3 (アイディアマップ): BGMを追加した(+10点)

合計: \_\_\_\_\_ (最高得点: 125)

ボーナス点込みで80点が合格点です。