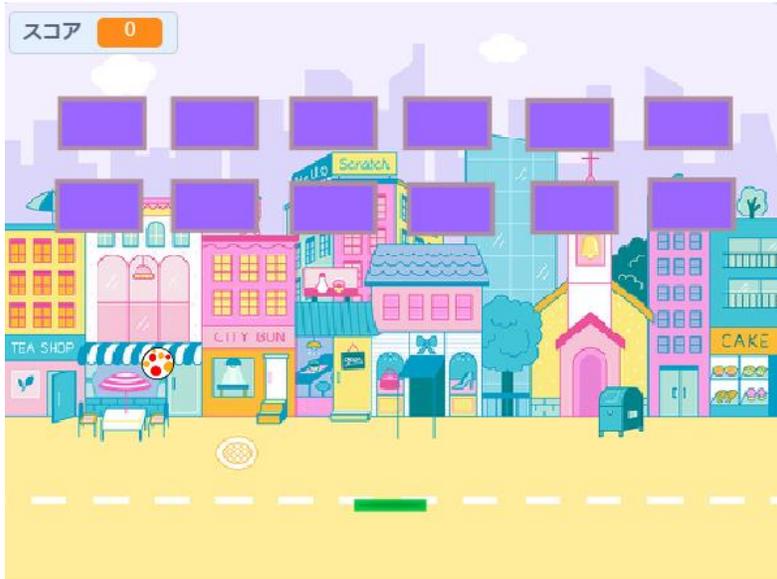


# ブロックくずし



## ○こんなゲームです

- ▶ 画面下のラケットを矢印キーで左右に動かしてボールを打ち返します
- ▶ ボールを並んでいるブロックに当てて消していきます
- ▶ すべてのブロックを消すと「勝ち」、ボールが画面の下の方に行くと即「負け」になります
- ▶ ボールの速度を変えたりして楽しんでください

## スプライトを用意しよう

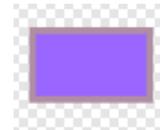
自分で描いてもライブラリから持ってきててもOKです



【ラケット】  
左右に動かす



【ボール】  
ブロックにぶつける



【ブロック】  
10~12個コピーして並べる



【勝ち】  
勝ちの表示



【負け】  
ゲームオーバーの表示

## ○スプライトをコピーするには

ブロックのスプライトは10~12個必要ですが、1つ作ってそれをコピーすると簡単です



# ラケットのプログラム



## ○左右に動かすには

Scratch code for racket movement. It starts with a 'when green flag clicked' event block, followed by a 'set x coordinate to 0, y coordinate to -130' block. Then there is a 'forever' loop containing two 'if' blocks. The first 'if' block checks 'if right arrow key is pressed' and contains a 'change x coordinate by 10' block. The second 'if' block checks 'if left arrow key is pressed' and contains a 'change x coordinate by -10' block.

最初の位置

Simplified Scratch code for racket movement. It starts with a 'when green flag clicked' event block, followed by a 'set x coordinate to 0, y coordinate to -130' block. Then there is a 'forever' loop containing two 'if' blocks. The first 'if' block checks 'if right arrow key is pressed' and contains a 'change x coordinate by 10' block. The second 'if' block checks 'if left arrow key is pressed' and contains a 'change x coordinate by -10' block.

ラケットの速度

## ○ボールにあてるとはね返る

Scratch code for ball bounce. It starts with a 'when green flag clicked' event block, followed by a 'forever' loop. Inside the loop, there is a 'when ball touches' block, followed by a 'send message to bound' block, and then a 'wait 1 seconds' block.

メッセージで「バウンド」を作る

Simplified Scratch code for ball bounce. It starts with a 'when green flag clicked' event block, followed by a 'forever' loop. Inside the loop, there is an 'if' block that checks 'if ball touches' and contains a 'send message to bound' block, followed by a 'wait 1 seconds' block.

# ボールのプログラム



○ボールが動く、ラケットより下にきたらゲームオーバー（負け）

Scratch code for ball movement and game over condition:

- が押されたとき (When green flag clicked)
- x座標を -200 、y座標を -100 にする (Set x coordinate to -200, y coordinate to -100)
- 45 度に向ける (Set direction to 45 degrees)
- ずっと (Forever loop)
- 5 歩動かす (Move 5 steps)
- もし端に着いたら、跳ね返る (If edge reached, bounce back)
- もし なら (If condition)
- y座標 < -160 (y coordinate < -160)
- 負け を送る (Send message: Lose)

メッセージで「負け」を作る

○ボールがはね返る

Scratch code for ball bounce:

- パウンド を受け取ったとき (When received message: Bounce)
- 0 度に向ける (Set direction to 0 degrees)
- 180 - 向き (180 - direction)

ラケットから送られてくるメッセージ

最初の位置と動く方向

Scratch code for ball movement and game over condition (repeated):

- が押されたとき (When green flag clicked)
- x座標を -200 、y座標を -100 にする (Set x coordinate to -200, y coordinate to -100)
- 45 度に向ける (Set direction to 45 degrees)
- ずっと (Forever loop)
- 5 歩動かす (Move 5 steps)
- もし端に着いたら、跳ね返る (If edge reached, bounce back)
- もし y座標 < -160 なら (If condition)
- 負け を送る (Send message: Lose)

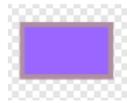
はね返る向き

Scratch code for ball bounce (repeated):

- パウンド を受け取ったとき (When received message: Bounce)
- 180 - 向き 度に向ける (180 - direction degrees)

ボールがラケットより下に来たとき

# ブロックのプログラム



✖ 10~12個

ブロックが変化の様子をよく観察しよう

○プログラムを組んだらコピーして増やします

ボールが当たるといろいろと変化します

```
が押されたとき
  大きさを 100 %にする
  90 度に向ける
  表示する
```



```
ずっと
  もし ボール に触れた なら
    パウンド を送る
```

```
10 回繰り返す
  色 の効果を 25 ずつ変える
  幽霊 の効果を 4 ずつ変える
  大きさを -4 ずつ変える
  15 度回す
  スコア を 1 ずつ変える
  隠す
```

変数で「スコア」を作ります

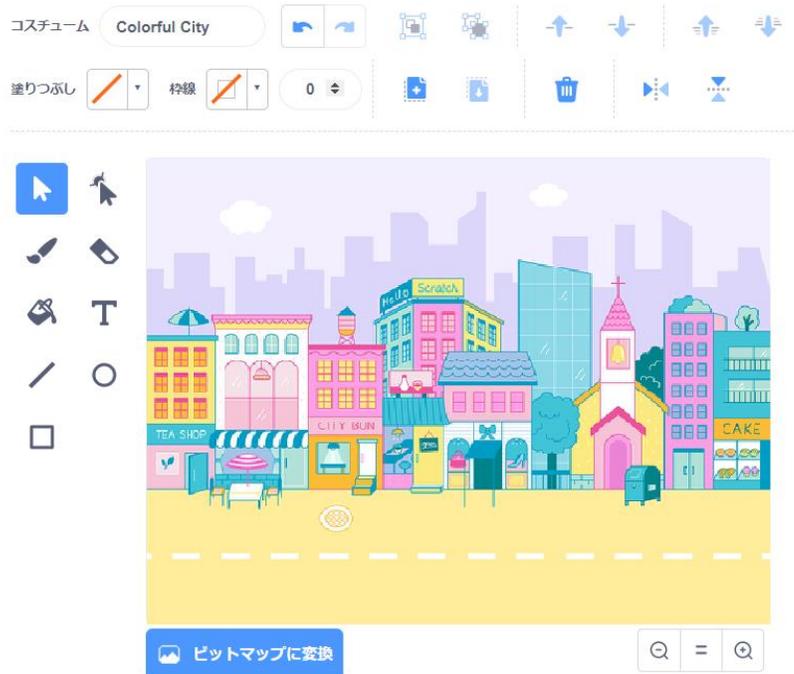
○コピー



```
が押されたとき
  大きさを 100 %にする
  90 度に向ける
  表示する
  ずっと
    もし ボール に触れた なら
      パウンド を送る
      10 回繰り返す
        色 の効果を 25 ずつ変える
        幽霊 の効果を 4 ずつ変える
        大きさを -4 ずつ変える
        15 度回す
      隠す
      スコア を 1 ずつ変える
```

# ステージのプログラム

## ○ステージの背景を決めます



自分で描いてもライブラリから持ってきててもOKです



メッセージで「勝ち」を作る

スコアをリセットします



すべてのブロックを消したら（数字はブロックの数です）

## 「勝ち」表示のプログラム

勝ち！

ステージから送られてくるメッセージ



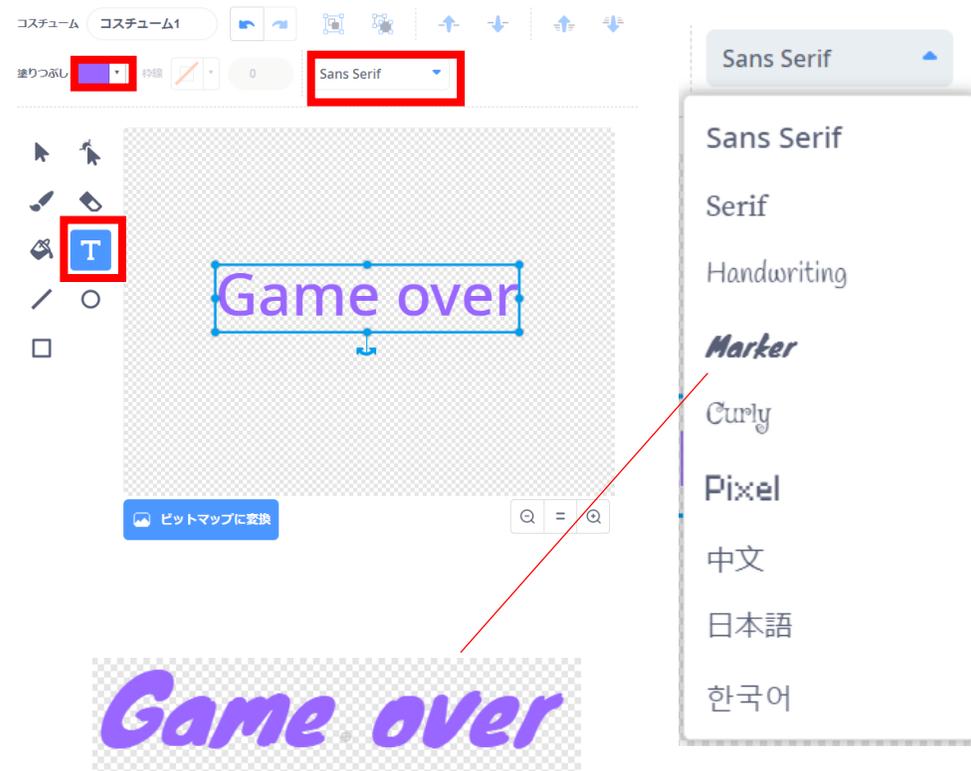
## 「ゲームオーバー」表示のプログラム

ゲームオーバー

ボールから送られてくるメッセージ



○ T (テキスト) で字体や色、大きさを変わります



## ○レベルチェック

- 必要なブロックを見つけられる
- 見本を見ながらブロックを並べられる
- 間違ったところを自分で見つけられる、直せる
- ブロックの意味（コマンド）がなんとなく分かる
- ペイントエディタが使える（初歩レベル）
- 制作途中で作品を上書き保存できる

## ○関連する学習

x座標、y座標

正の数、負の数

順次処理

衝突判定

条件分岐

メッセージ

コスチュームの切り替え

ステージの切り替え

ライブラリの活用

BGM・音の入れ方

デバッグ

キーボード操作