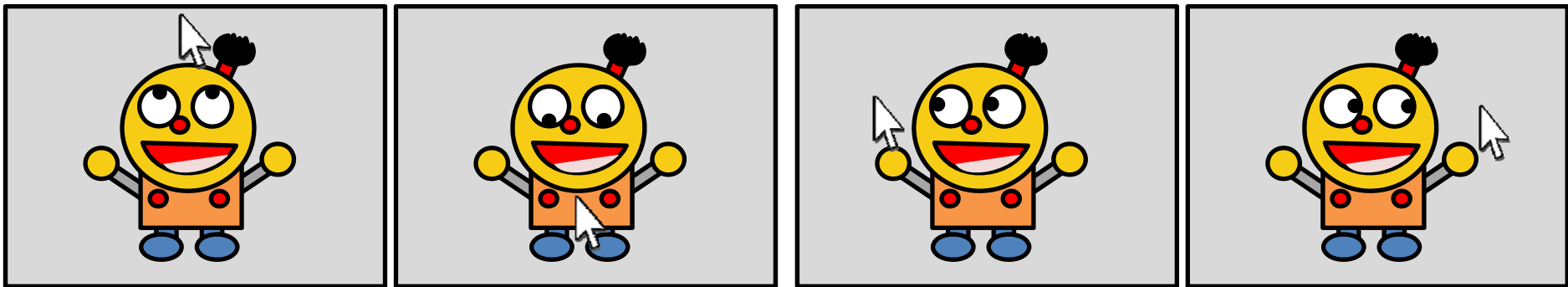


# プログラミング演習I（第4回）課題

## • 基本課題① スケッチ名: eye2

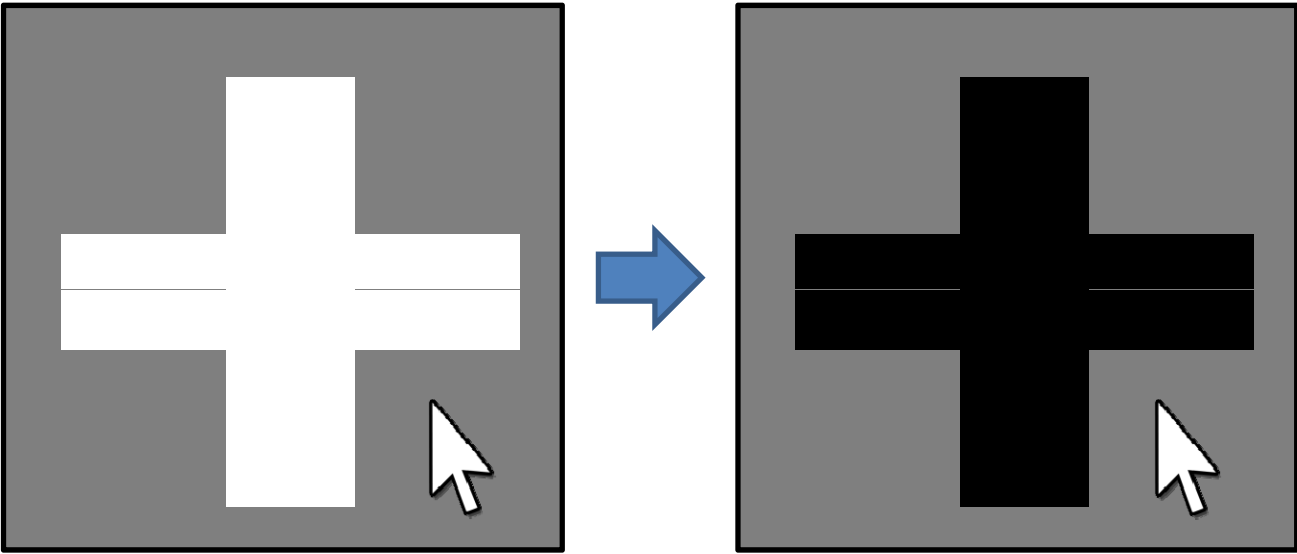
- カーソルの位置によってキャラクターの目の向きが変わるプログラムを作ってください。
- ただし、カーソルがキャラクターの顔に対して【上にある時】【下にある時】【左にある時】【右にある時】の4パターンで表現すること。
- カーソルが顔に対して斜め方向にある時は、目は中央にしてください。



# プログラミング演習I (第4回) 課題

## • 基本課題② スケッチ名 : **cross**

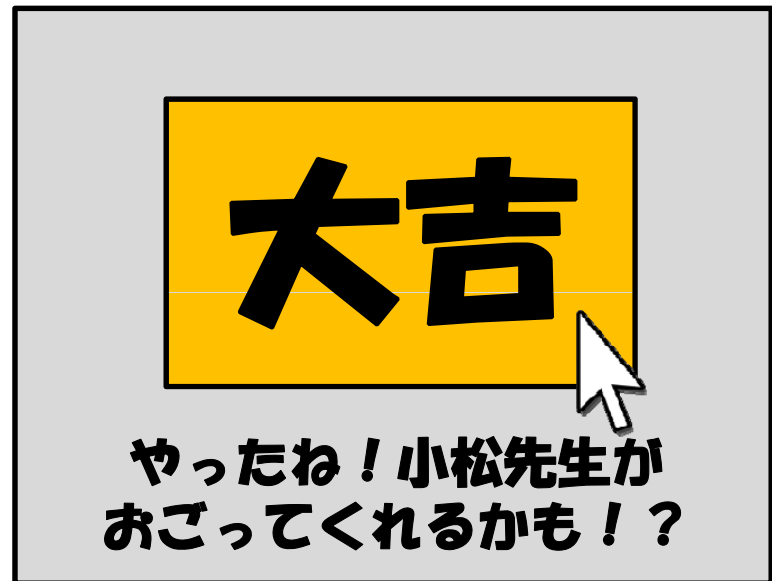
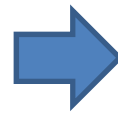
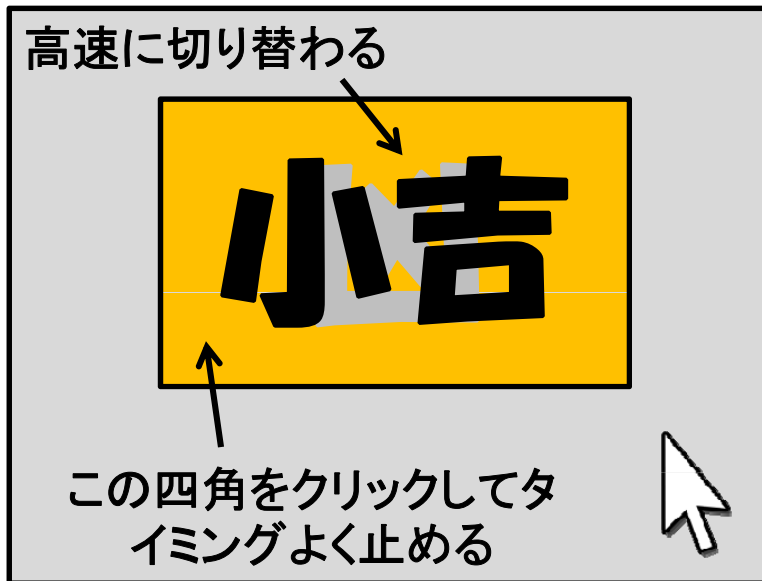
- 十字型の図形に対してマウスの当り判定をするプログラムを作ってください。
- 適当な十字型の図形を描き、図形の【外部】をクリックしたら図形の色が変わるようにしてください。
- 図形内部をクリックしても反応しないようにすること。



# プログラミング演習I (第4回) 課題

## • 基本課題③ スケッチ名 : slotkuji

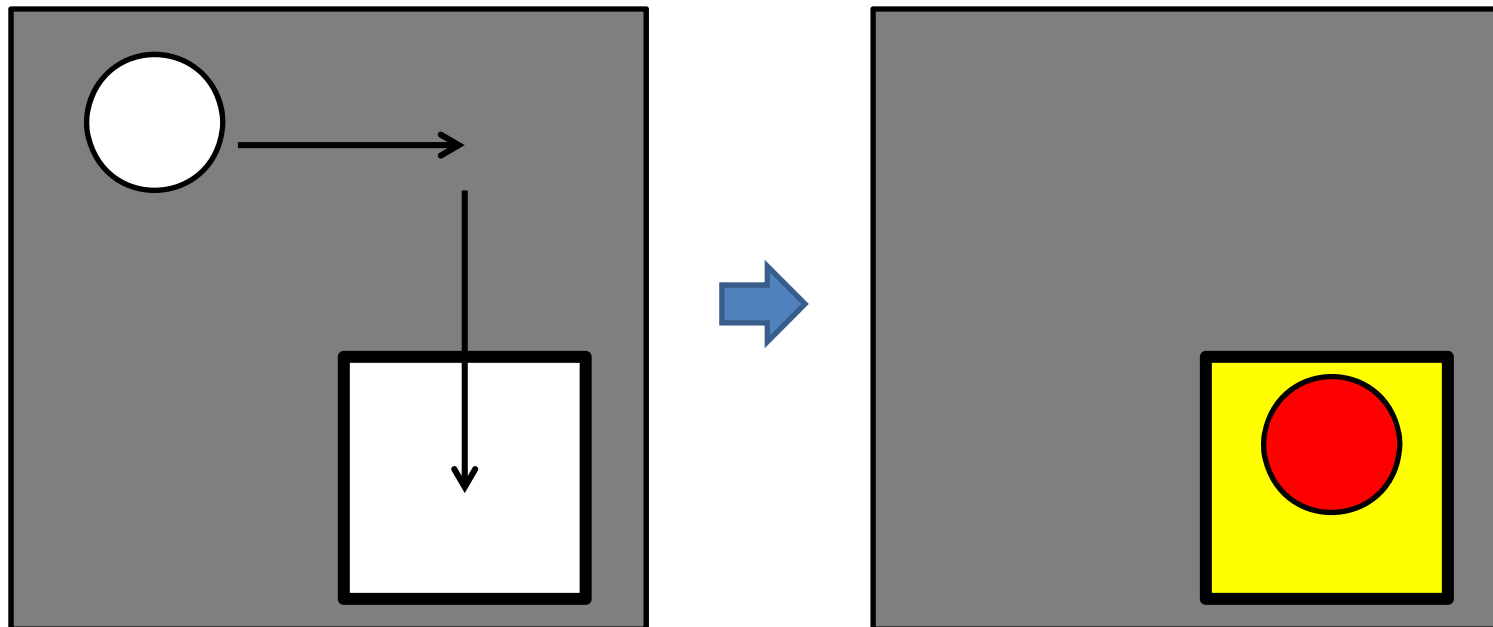
- おみくじのプログラムを作ってください。
- スロットのように大吉・中吉・小吉・凶の4種類が高速に表示され、四角いところをクリックするとストップして、「今日の運勢」に関するメッセージが表示されるようにしてください。
- もう一度クリックすると2回目にチャレンジできるようにすること。



# プログラミング演習I (第4回) 課題

## • 発展課題① スケッチ名: keycontrol

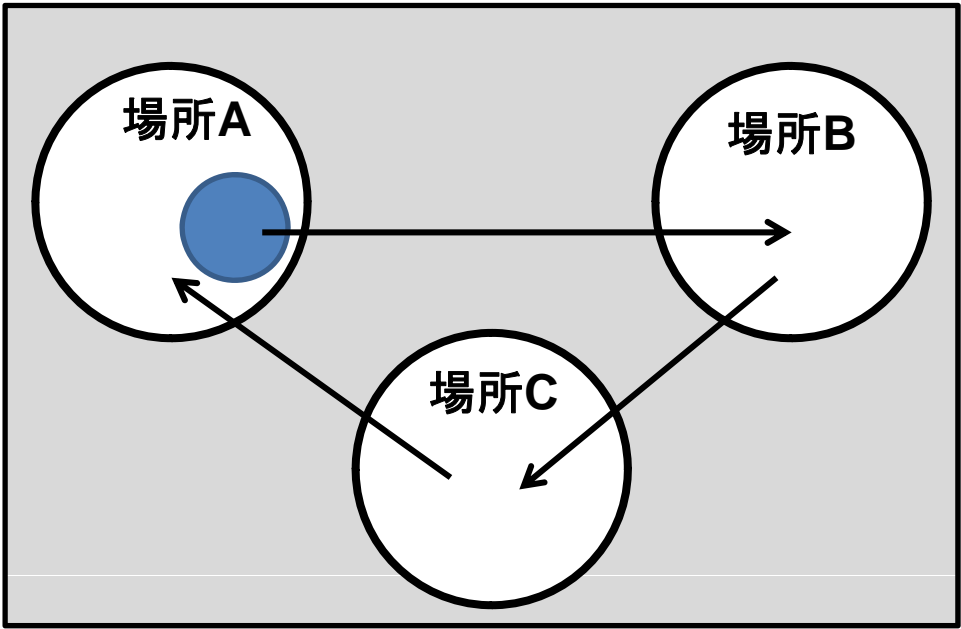
- 方向キーの操作で直径50の円を移動させ、円が目標となる正方形(一辺が60ピクセル)の中に完全に入ったら、正方形と円の色が違う色になるようにしてください。
- 実行するたびにランダムで正方形の位置が変わるようにすること。



# プログラミング演習I（第4回）課題

## ● 発展課題② スケッチ名: places

- 画面をクリックするごとに、円が場所A→場所B→場所C→場所A→…と順に移動していくプログラムを作成してください。
- 場所間の移動はアニメーションで自動的に動くものとし、また、円の移動中に画面をクリックしても反応しないようにしてください。



画面クリックごとに円が次の場所に移動していく。それぞれの場所の形はなんでも良い。

# 今日使うテクニック

## ① text()で表示する文字の大きさを変える方法

- 文字の大きさを変えるには `textSize(文字サイズ)` を使う。

```
void setup() {  
  size(300, 150);  
}  
  
void draw() {  
  fill(0);  
  textSize(50); // 文字の大きさを設定  
  text( "Processing", 20, 90 ); // 文字を表示  
}
```



Processing

- `textSize()`は、`fill()`や`stroke()`と同様に何回でもパラメータを変えて指定できるので、大きさの違う文字を混在させることができる。

# 今日使うテクニック

## ② text() で表示する文字の書体(フォント)を変える方法

- フォントを変えるには、PFont、createFont()、textFont() を使う。
- 日本語を使いたいときは日本語フォントの指定が必要
- 以下はHGS創英角ポップ体で「Processing」と書く例

```
PFont myFont; // フォント

void setup() {
  size(300, 150);
  myFont = createFont("HGSSoeiKakupoptai",10); // フォントを準備
  textFont(myFont); // フォントを設定
  textSize(50);      // 文字サイズを改めて変更することもできる
}

void draw() {
  fill(0);
  text( "Processing", 20, 90 ); // 文字を表示
}
```

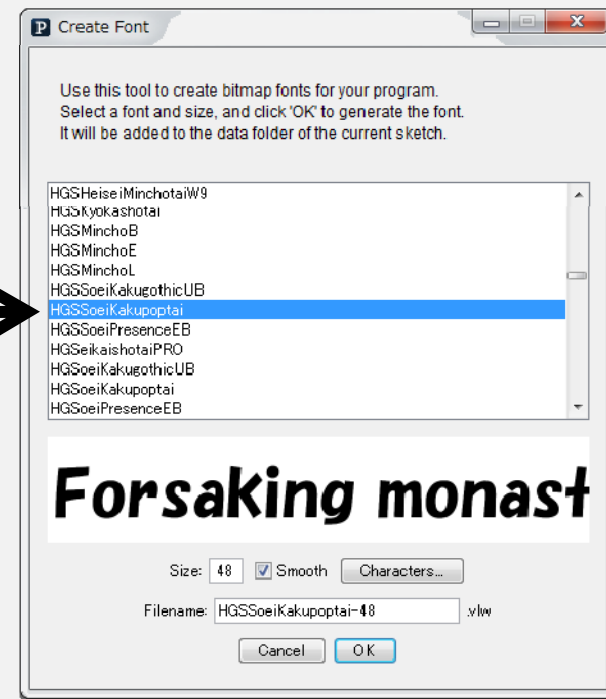
**Processing**

# 今日使うテクニック

- PFont はフォントを格納する変数につかうデータ型です。  
int や float などと同じような扱い。
- createFont( フォント名, 文字サイズ ) でフォントを準備する。  
フォント名は、Processingのメニューの  
Tools -> Create Font...  
で出てくるパネルで確認できる。

このリストにプログラム中で使える  
フォント名が表示される。

- 最後に、textFont( フォント ) で  
フォントを設定する。

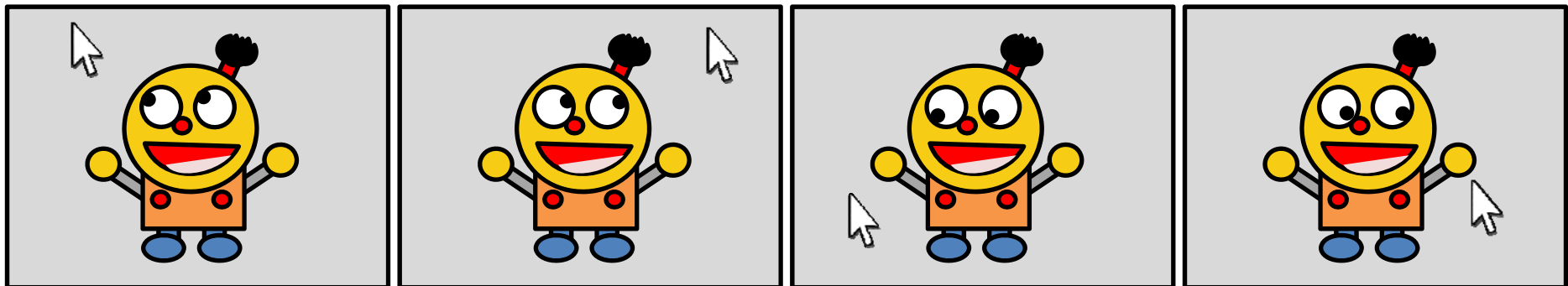




# プログラミング演習I（第4回）課題

## • 基本課題① スケッチ名：`eye2`

- カーソルの位置によってキャラクターの目の向きが変わるプログラムを作ってください。
- ただし、カーソルがキャラクターの顔に対して【左上にある時】【右上にある時】【左下にある時】【右下にある時】の4パターンで表現すること。
- カーソルが顔の真上や真横にある時は、目は中央にしてください。

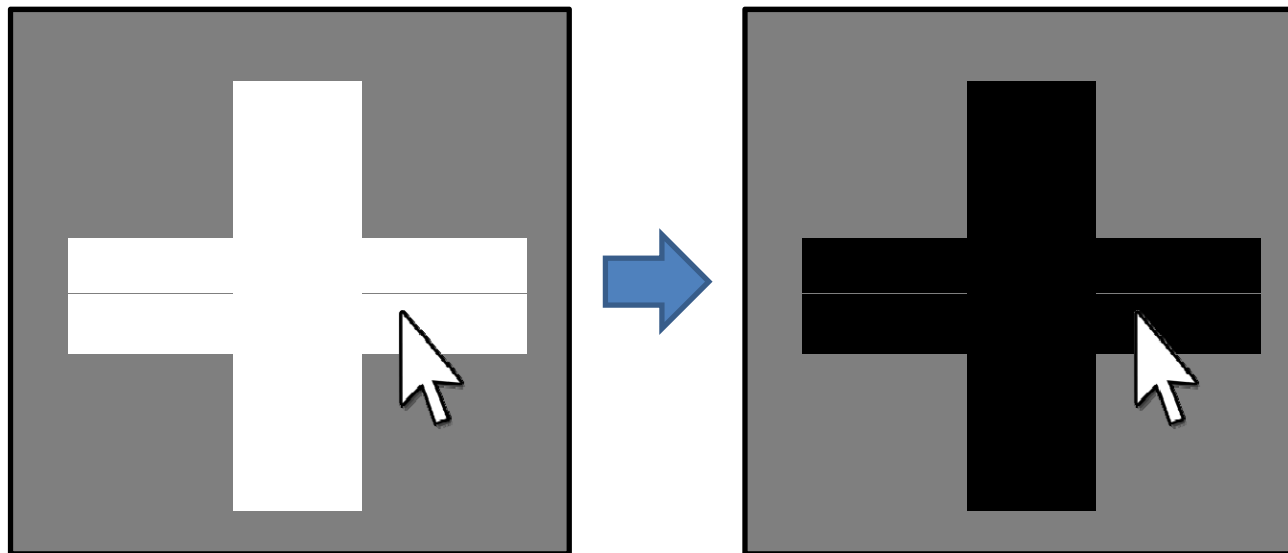


# プログラミング演習I (第4回) 課題

---

## • 基本課題② スケッチ名 : **cross**

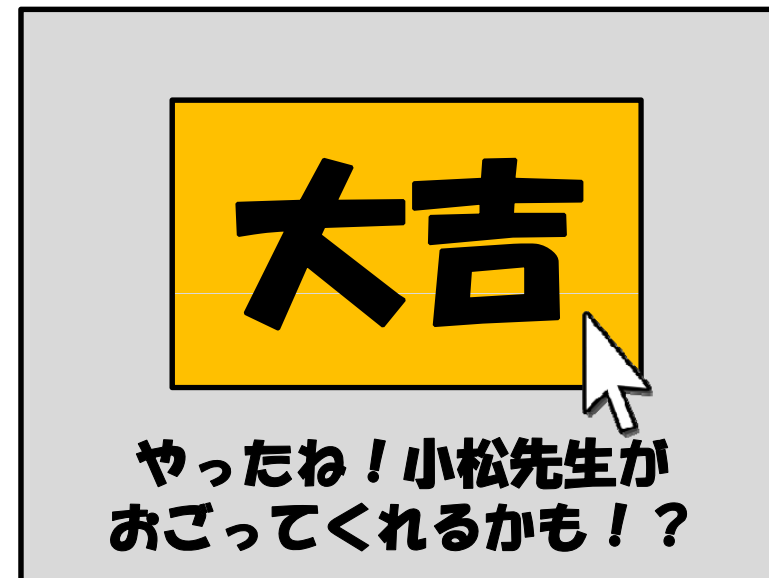
- 十字型の図形に対してマウスの当り判定をするプログラムを作ってください。
- 適当な十字型の図形を描き、図形の【内部】をクリックしたら図形の色が変わるようにしてください。
- 図形外部をクリックしても反応しないようにすること。



# プログラミング演習I (第4回) 課題

## • 基本課題③ スケッチ名 : slotkuji

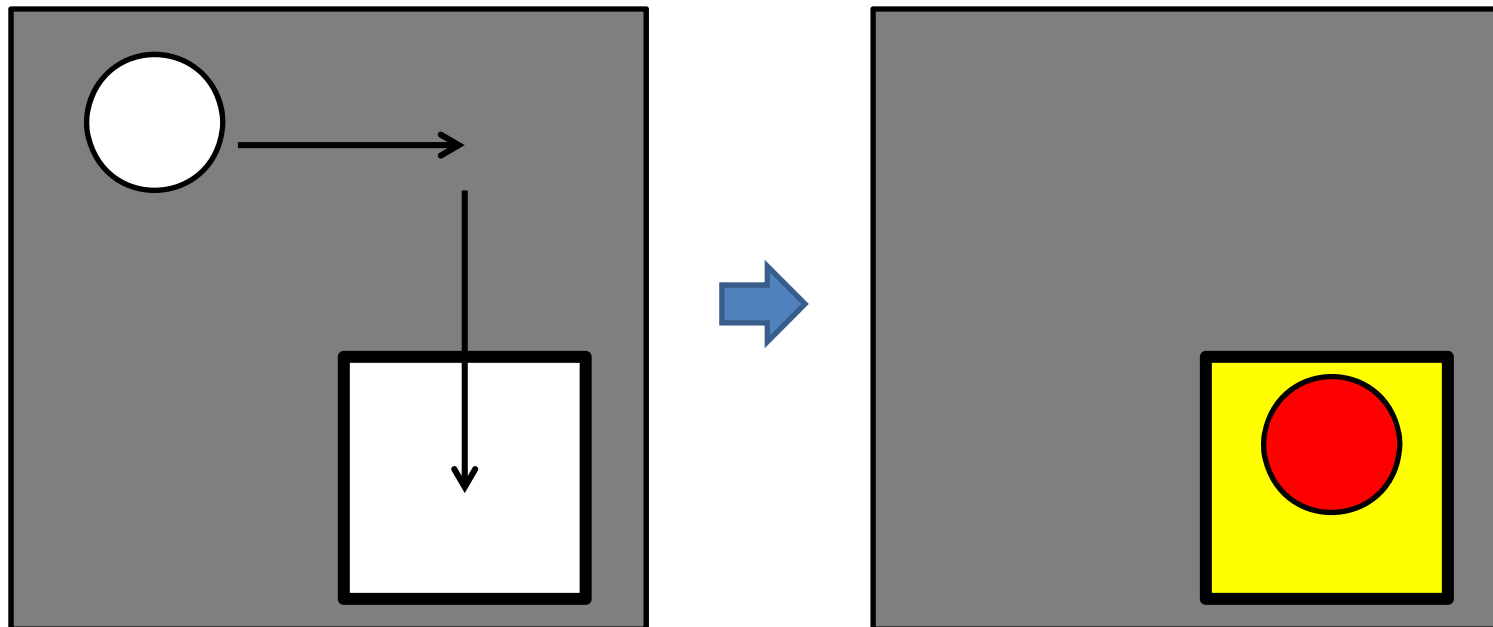
- おみくじのプログラムを作ってください。
- スロットのように大吉・中吉・小吉・凶の4種類が高速に表示され、四角いところをクリックするとストップして、「今日の運勢」に関するメッセージが表示されるようにしてください。
- もう一度クリックすると2回目にチャレンジできるようにすること。



# プログラミング演習I (第4回) 課題

## • 発展課題① スケッチ名: keycontrol

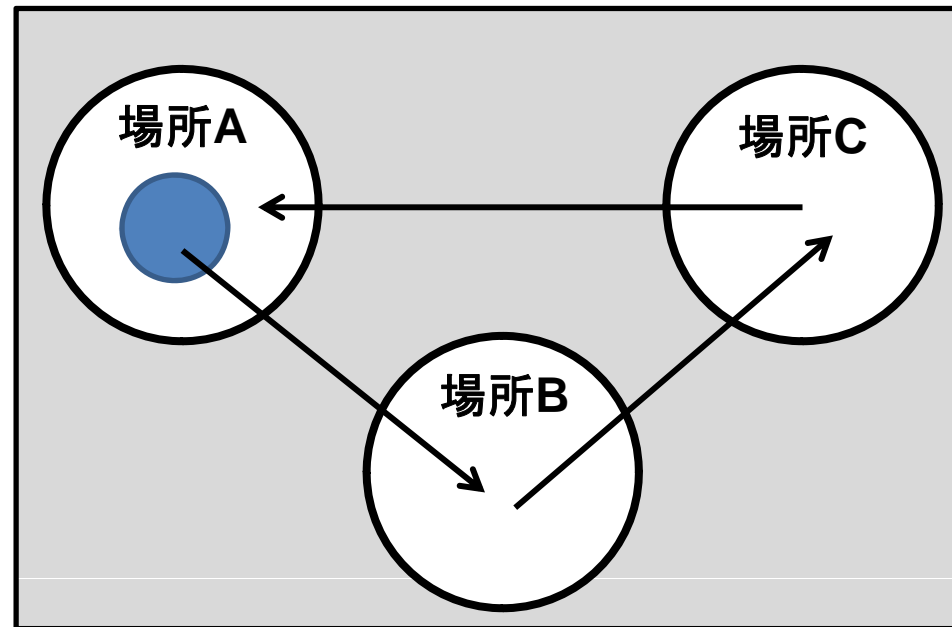
- 方向キーの操作で直径50の円を移動させ、円が目標となる正方形(一辺が60ピクセル)の中に完全に入ったら、正方形と円の色が違う色になるようにしてください。
- 実行するたびにランダムで正方形の位置が変わるようにすること。



# プログラミング演習I (第4回) 課題

## • 発展課題② スケッチ名: places

- 画面をクリックするごとに、円が場所A→場所B→場所C→場所A→…と順に移動していくプログラムを作成してください。
- 場所間の移動はアニメーションで自動的に動くものとし、また、円の移動中に画面をクリックしても反応しないようにしてください。



画面クリックごとに円が次の場所に移動していく。それぞれの場所の形はなんでも良い。

# 今日使うテクニック

## ① text()で表示する文字の大きさを変える方法

- 文字の大きさを変えるには `textSize(文字サイズ)` を使う。

```
void setup() {  
  size(300, 150);  
}  
  
void draw() {  
  fill(0);  
  textSize(50); // 文字の大きさを設定  
  text( "Processing", 20, 90 ); // 文字を表示  
}
```



Processing

- `textSize()`は、`fill()`や`stroke()`と同様に何回でもパラメータを変えて指定できるので、大きさの違う文字を混在させることができる。

# 今日使うテクニック

## ② text() で表示する文字の書体(フォント)を変える方法

- フォントを変えるには、PFont、createFont()、textFont() を使う。
- 日本語を使いたいときは日本語フォントの指定が必要
- 以下はHGS創英角ポップ体で「Processing」と書く例

```
PFont myFont; // フォント

void setup() {
  size(300, 150);
  myFont = createFont("HGSSoeiKakupoptai",10); // フォントを準備
  textFont(myFont); // フォントを設定
  textSize(50);      // 文字サイズを改めて変更することもできる
}

void draw() {
  fill(0);
  text( "Processing", 20, 90 ); // 文字を表示
}
```

**Processing**

# 今日使うテクニック

- PFont はフォントを格納する変数につかうデータ型です。  
int や float などと同じような扱い。
- createFont( フォント名, 文字サイズ ) でフォントを準備する。  
フォント名は、Processingのメニューの  
Tools -> Create Font...  
で出てくるパネルで確認できる。

このリストにプログラム中で使える  
フォント名が表示される。

- 最後に、textFont( フォント ) で  
フォントを設定する。

