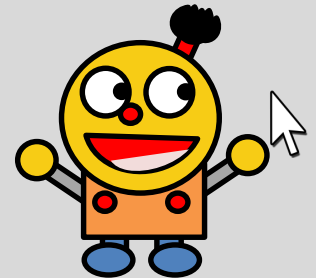
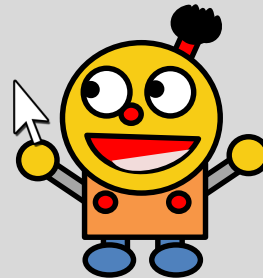
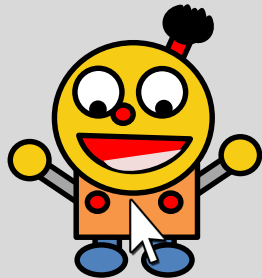
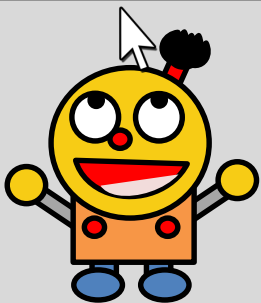


プログラミング演習I（第5回）課題

• 基本課題① スケッチ名：eye2

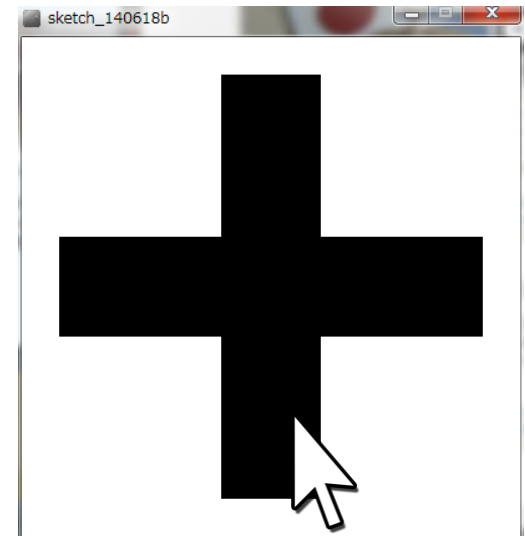
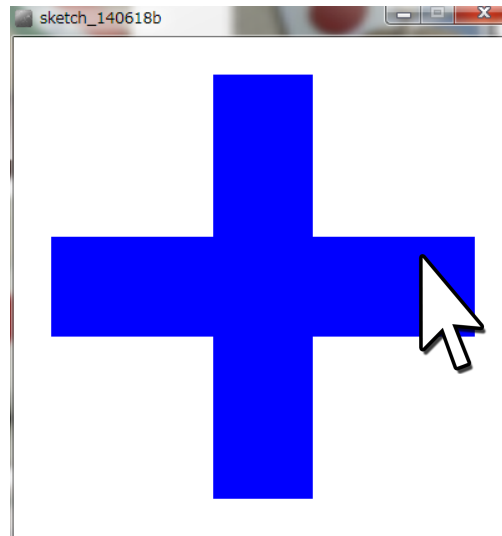
- カーソルの位置によってキャラクターの目の向きが変わるプログラムを作ってください。
- ただし、カーソルがキャラクターの顔に対して【上にある時】【下にある時】【左にある時】【右にある時】の4パターンで表現すること。



プログラミング演習I (第5回) 課題

• 基本課題② スケッチ名 : cross

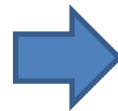
- 十字型の図形に対してマウスの当り判定をするプログラムを作ってください。
- 適当な十字型の図形を描き、図形の【内部】をクリックしたら図形の色が赤→青→黒→赤→青→黒と順に変わるようにせよ。
- 図形外部をクリックしても反応しないようにすること。



プログラミング演習I (第5回) 課題

基本課題③ スケッチ名: slot777

- 3つの数字を揃えるスロットのプログラムを作成せよ。
- 左と中央は7で固定し、右側だけ変化させよ。 右側の値は1から7まで順に増え、7までくると1に戻るようにせよ。STOPボタンをクリックするとそのタイミングでストップし、その時の数字を表示しなさい。
すべて7だった場合は黄色で塗りつぶしなさい
 - できれば楽しげな音も鳴らそう!
- STARTボタンを押すと、2回目にチャレンジできるようにすること。



ヒント

- 変数を整理しよう
 - 動いているかどうか:
 - 動いている最中に表示している値:
 - STOPを押された時に選ばれた値:
- 状況に応じて変数の値がどうなるかを整理
 - STARTボタンが押されるとどの変数の値がどう変化?
 - STOPボタンが押されるとどの変数の値がどう変化?
 - 画面で表示する値はどの変数の情報?

プログラミング演習I (第5回) 課題

発展課題① スケッチ名: launch2

- 600x400のウィンドウ左下から砲台の向きにボール(半径15m)を発射するプログラムを作成せよ。ただし、砲台は上下キーによって角度を1度ずつ変更(初期角度は45度)し、右キーで投射されるようにすること。また、初速を100m/sとせよ。さらに、現在の角度を左上に表示せよ。
- ランダムに指定した距離(0~600mの間)に半径25mの赤丸のターゲットを描け。赤丸は上下に1フレームあたり2m動き、上端下端で折り返すようにせよ。なお、ボールがターゲットに衝突するとターゲットを消し、次のターゲットを登場させるようにせよ。1フレームは0.1秒とする。
- できれば当たった時に爆発音を鳴らせ。

sketch_140617c

45 degrees

砲台

ボール

ターゲット

$$x = v_0 t \cos \theta$$

$$y = v_0 t \sin \theta - \frac{1}{2} g t^2$$

sketch_140617c

スピードが早すぎる場合はすり抜けてもよい

あたらまでは何度でもトライ可能に

あたらたら別の場所に登場

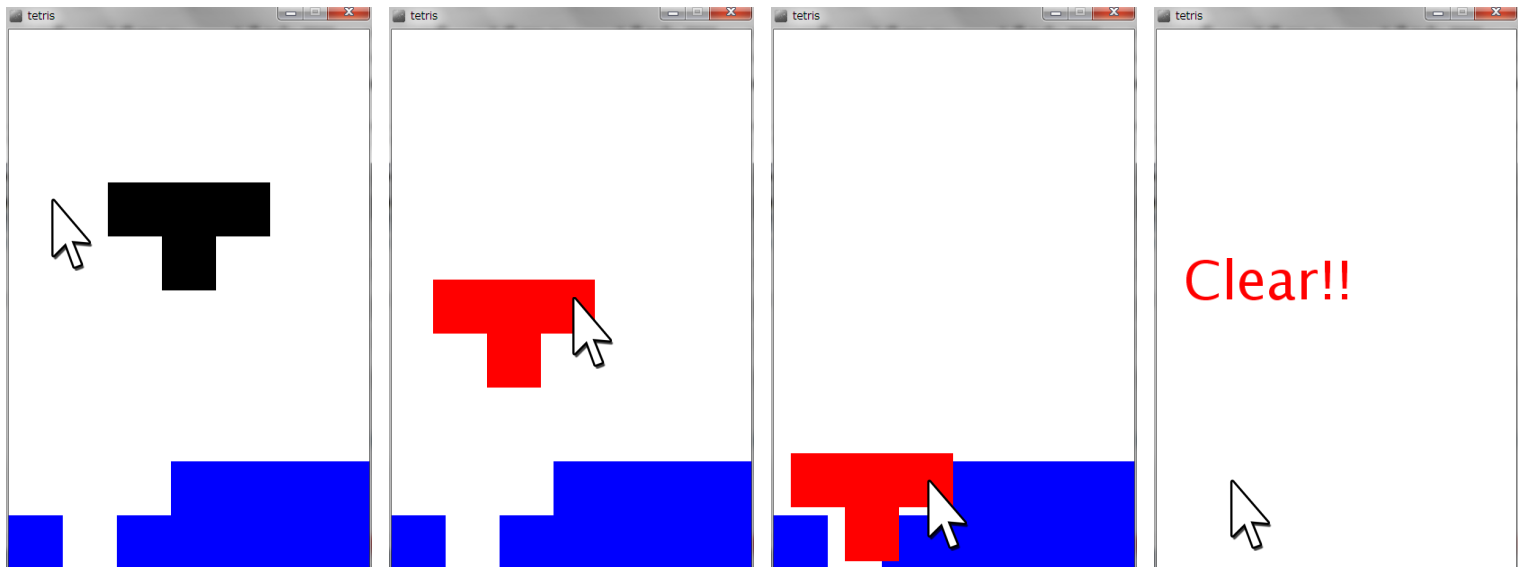
ヒント

- ランダムな値の取得には random を使おう
 - random(開始の数字, 終了の数字); で, 開始から終了までの実数値 (float) を取得することが出来る
 - 整数に変換するにはどうしたら良いか？

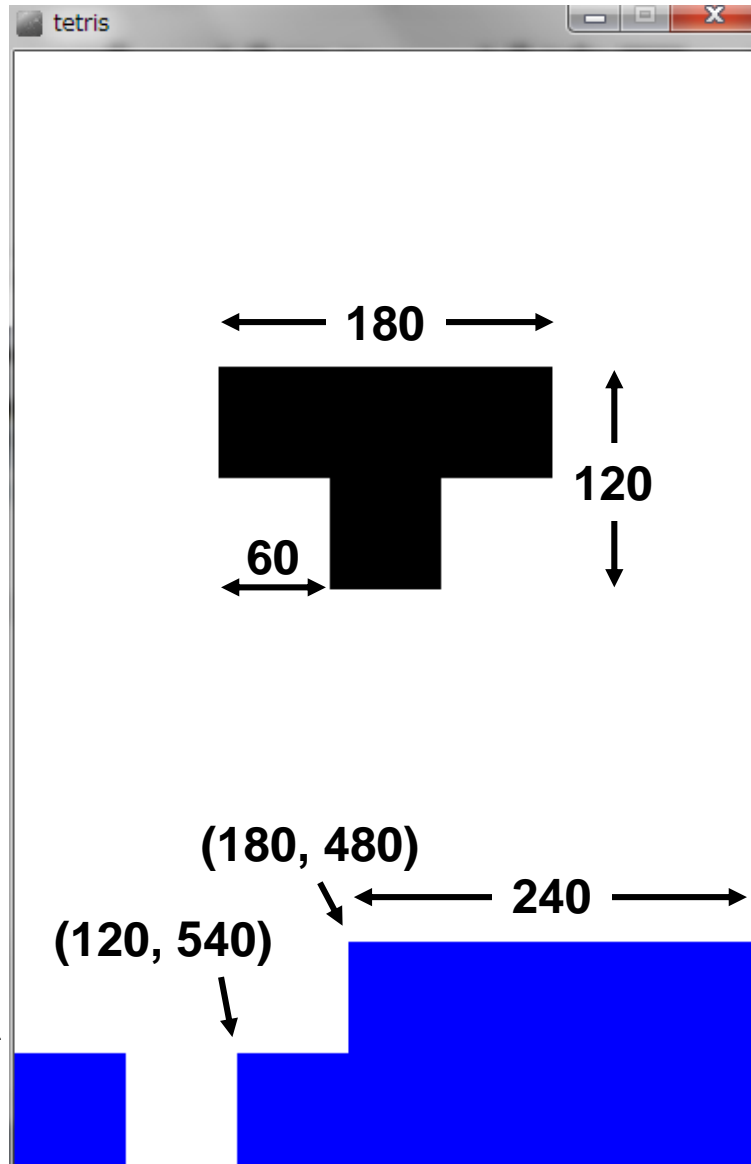
プログラミング演習I (第5回) 課題

• 発展課題② スケッチ名: Tetris

- 420x600のウィンドウ内に、下図に示す横180ピクセル縦12ピクセルの凸型の図形を描き(黒色)、図形内でマウスボタンを押すと選択状態(赤色)になり、ボタンを離すまで移動できるようにせよ。
- なお、移動においては相対位置をキープするようにして下さい。また、ドラッグ中は図形が赤色になるようにせよ。
- また、下図のように他のブロックを用意し、その隙間にピッタリハマった場合に「Clear!!」と表示するようにせよ(音を鳴らしても良い)



ヒント



- 必要な変数は何か？ 右に変数名を書き込もう！

マウスの座標(,)

ブロックの基本座標(,)

x方向の相対座標:

y方向の相対座標:

選択の有無:

クリアしたかどうか:

ヒント

- 相対座標はどのようにして保持し、その時の座標にどのようにして反映するか整理！

ドラッグ前

描画時の基本座標(blockX, blockY)

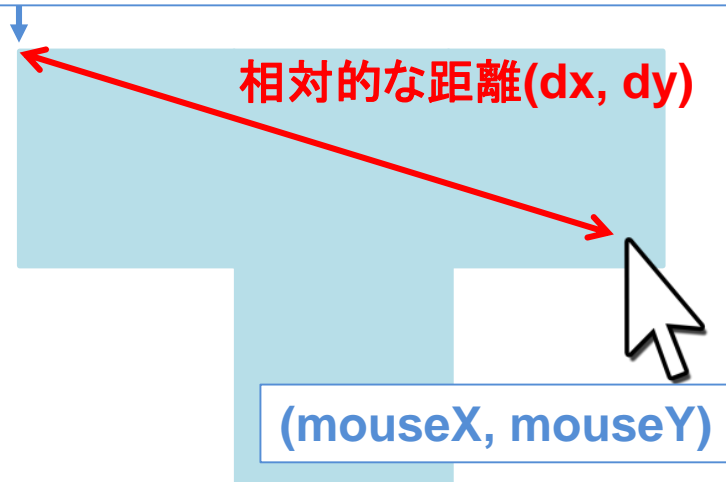


基本座標はどのように更新されるか？

ドラッグ中

基本座標(blockX, blockY)
はどのように更新されるか？

描画時の基本座標(blockX, blockY)



(mouseX, mouseY)

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
}

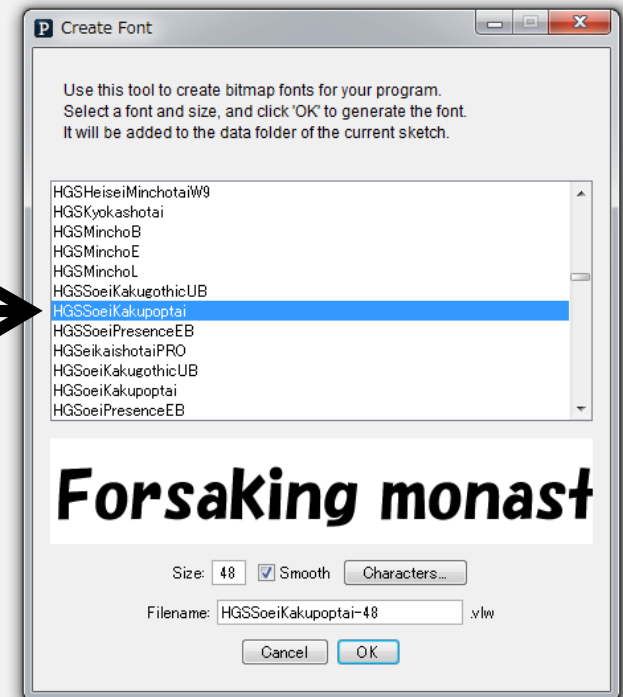
void draw() {
  fill(0);
  text( "Processing", 20, 90 ); // 文字を表示
}
```

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。

今日使うテクニック(音編)

- 音を鳴らすにはminimというものを利用する(詳しくは12回目で)
- 今日のテクニックを実現するにはおまじないが必要です
- 適当なwavやmp3ファイルをプログラムの上にドロップし下記のコードを実行

```
import ddf.minim.*;
Minim minim;
AudioSnippet crash;
void setup(){
  minim = new Minim( this );
  crash = minim.loadSnippet( "crash.mp3" );
}
void stop(){
  crash.close();
  minim.stop();
  super.stop();
}
```

赤文字は毎回同じおまじない
青文字は音に応じて利用

音を再生する部分に挿入

```
crash.rewind();
crash.play();
```

今日使うテクニック(音編)

マウスクリックの時に
クラッシュ音を鳴らす

crash.mp3 は事前に
ダウンロードし、
プログラムの上に
ドロップしておく！

```
import ddf.minim.*;
Minim minim;
AudioSnippet crash;
void setup(){
  size( 400, 400 );
  minim = new Minim( this );
  crash = minim.loadSnippet( "crash.mp3" );
}
void stop(){
  crash.close();
  minim.stop();
  super.stop();
}
void mousePressed(){
  crash.rewind();
  crash.play();
}
```

複数の音を使う場合
変数として定義を
繰り返すだけ！

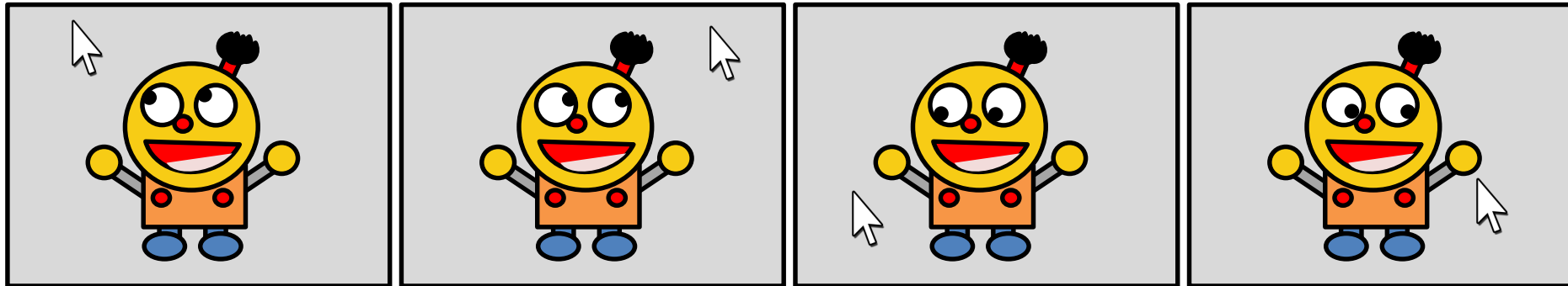
mp3やwavもOK!!

```
import ddf.minim.*;
Minim minim;
AudioSnippet taiko;
AudioSnippet kane;
void setup(){
  size( 400, 400 );
  minim = new Minim( this );
  taiko = minim.loadSnippet( "taiko.mp3 " );
  kane = minim.loadSnippet( "kane.wav" );
}
void stop(){
  taiko.close();
  kane.close();
  minim.stop();
  super.stop();
}
void mousePressed(){
  if( mouseX < width/2 ){
    taiko.rewind();
    taiko.play();
  } else {
    kane.rewind();
    kane.play();
  }
}
```


プログラミング演習I (第5回) 課題

• 基本課題① スケッチ名: eye2

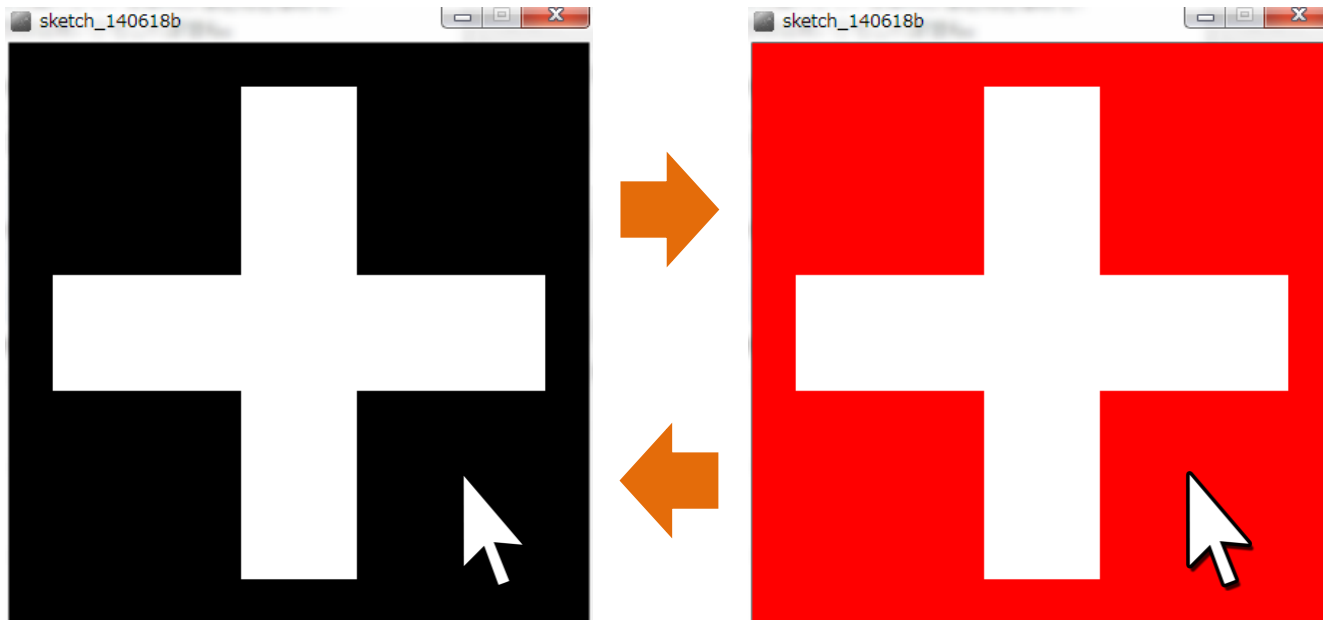
- カーソルの位置によってキャラクターの目の向きが変わるプログラムを作ってください。
- ただし、カーソルがキャラクターの顔に対して【左上にある時】【右上にある時】【左下にある時】【右下にある時】の4パターンで表現すること。



プログラミング演習I (第5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外部の色(背景色)が黒, 赤, 黒, 赤と交互に変わるようにせよ。
- 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5回) 課題

基本課題③ スケッチ名 : slot777

- 3つの数字を揃えるスロットのプログラムを作ってください。
- 左と中央の数字は7の状態、右の数字のみ動かさない。なお、draw()の度にランダムに5か6か7の数字を表示するようにせよ。
- STOPボタンをクリックするとストップして、数字を表示しないで。
- 7が揃った場合は黄色で塗りつぶさない(可能なら音も鳴らそう)
- STARTボタンを押すと、何度でもチャレンジできるようにすること。



ヒント

- ランダムな値の取得には random を使おう
 - random(開始の数字, 終了の数字); で, 開始から終了までの実数値 (float) を取得することが出来る
 - 整数に変換するにはどうしたら良いか?
 - 開始の数字から終了の数字の間の実数の値が取れるということは, どういうことだろうか?

ヒント

- 変数を整理しよう
 - 動いているかどうか:
 - 動いている最中に表示している値:
 - STOPを押された時に選ばれた値:
- 状況に応じて変数の値がどうなるかを整理
 - STARTボタンが押されるとどの変数の値がどう変化?
 - STOPボタンが押されるとどの変数の値がどう変化?
 - 画面で表示する値はどの変数の情報?

プログラミング演習I (第5回) 課題

• 発展課題① スケッチ名: launch

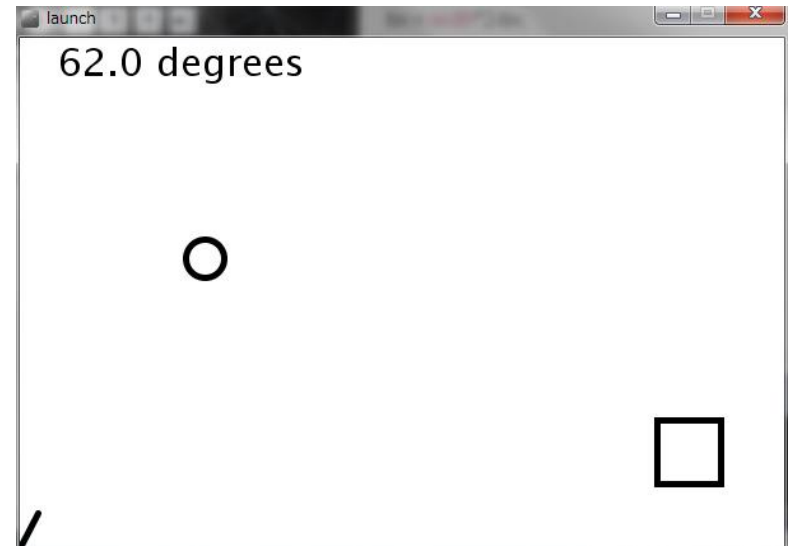
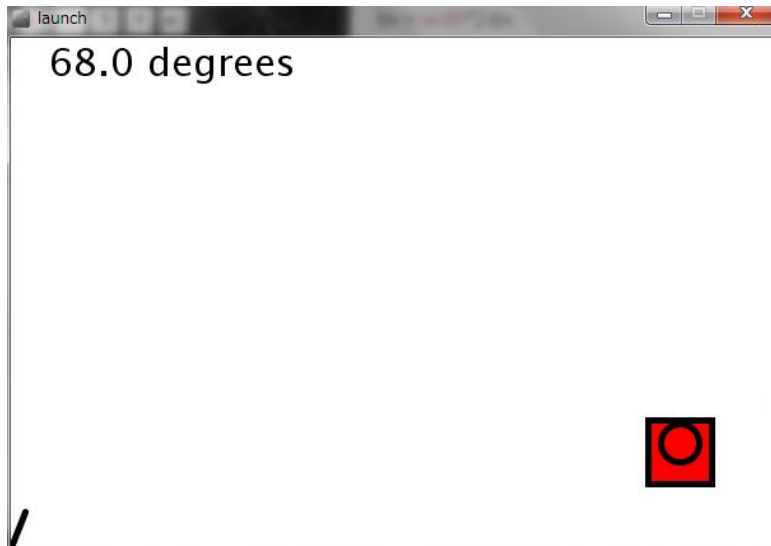
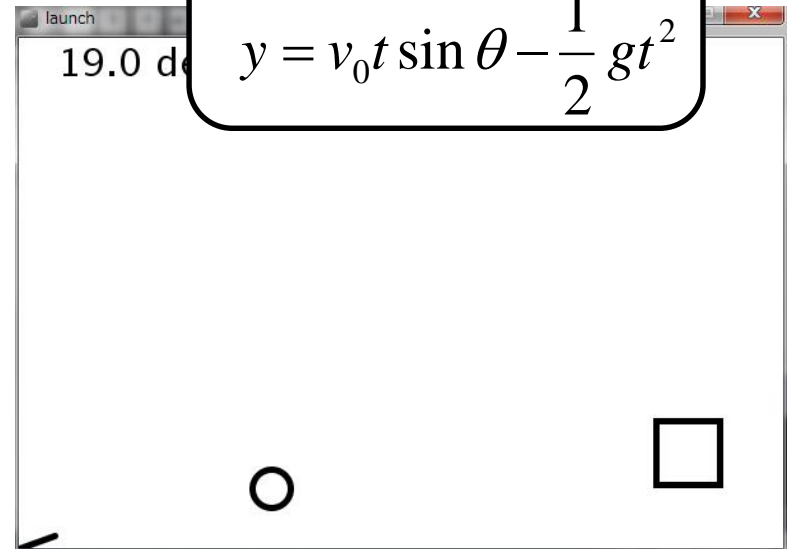
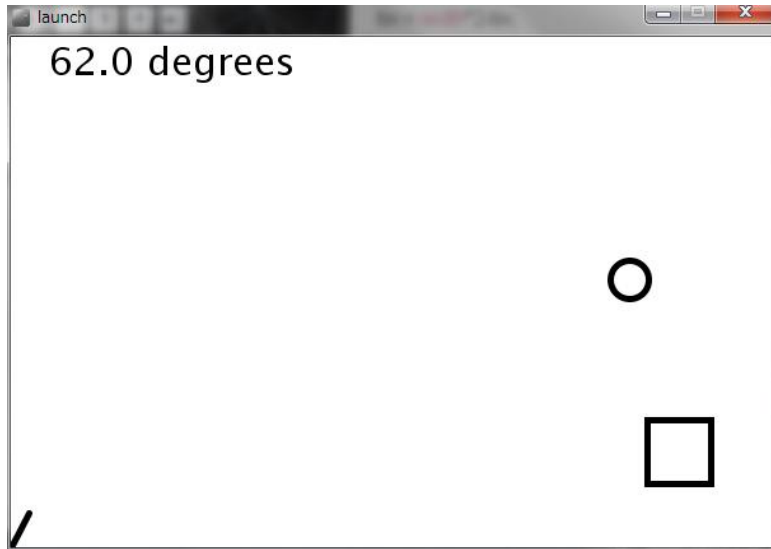
- 600x400の画面左下から砲台の向きにボール(直径30m)を発射するプログラムを作成せよ。
- 砲台の角度は上下キーで1度ずつ上下できるように(初期角度は45度とする)し、右キーで発射するようにすること。また、現在の角度を左上に表示せよ。
- 初速は100m/sとすること。さらに、画面の右端まで行くと左方向に跳ね返るようにせよ。draw1回あたり0.1秒ずつ進むものとする。
- 500m離れた場所の適当な高さに縦横50mの四角形の的を配置し、それに完全に入ると四角形とボールが赤色になってボールが停止するようにせよ(次頁の図参照)。
- 可能なら、当たった時に音がなるようにせよ。

動作イメージ

斜方投射の式

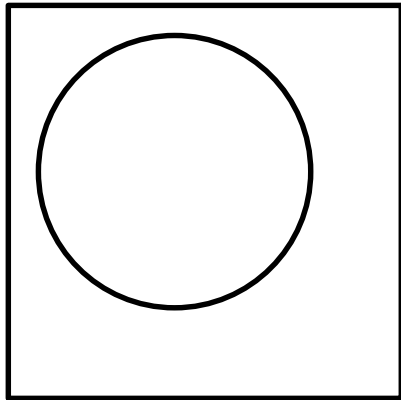
$$x = v_0 t \cos \theta$$

$$y = v_0 t \sin \theta - \frac{1}{2} g t^2$$

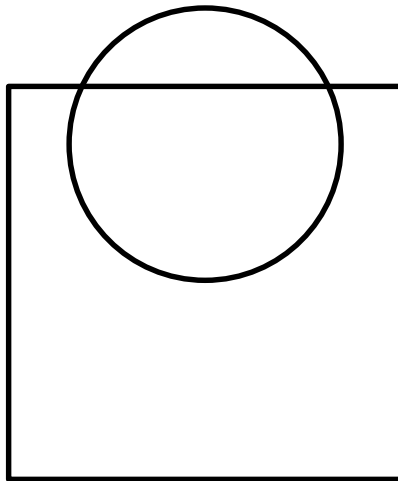


ヒント

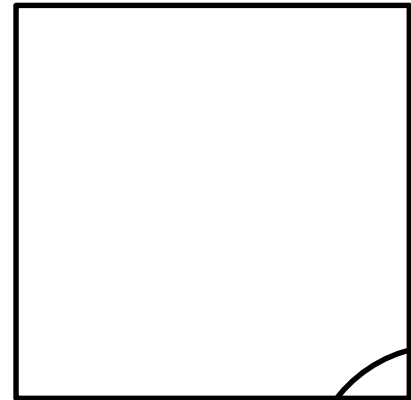
- 当たり判定はどうか？
 - 条件を整理して図示してみよう！
 - あたりとなる時のボールの中心座標は？



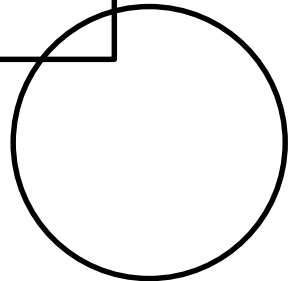
あたり



はずれ



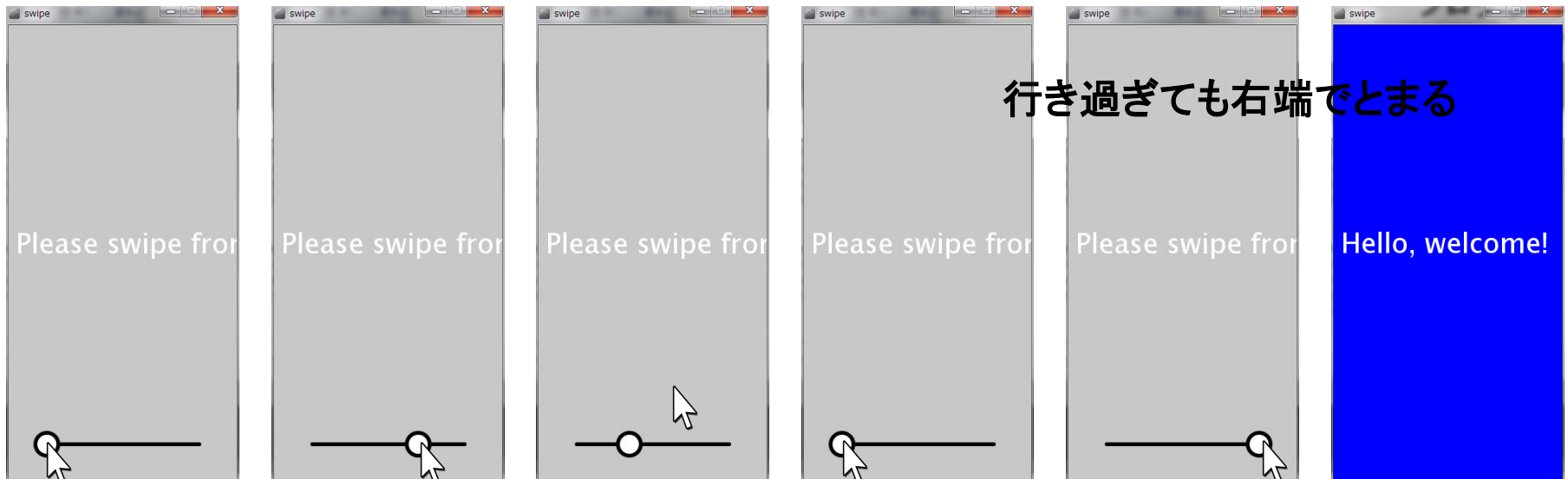
はずれ



プログラミング演習I (第5回) 課題

• 発展課題② スケッチ名: unlock

- 縦長のウインドウの下部に左の方に丸型のものを用意し、それを右側にスライド(スワイプ)させると、ロックが解除され何らかの画面に遷移する仕組みを実現せよ(例えば、作成したキャラクタが描画されている画面など)
- なお、右端まで移動せずに手を離れた場合は、5ピクセル/フレームの速度で最初の位置に戻るようにせよ。また、指定の位置より右や左にはみ出ないようにせよ。



ヒント

- 基本的には先週の仕組みと同じ. 必要な変数は?
 - マウスのボタンを離した時に元に戻ろうとする動きを実現するのに別の変数が必要となる. それは何か?

- マウスのボタンが離された時の動作を考える!
 - マウスのボタンが押されている時はどういう動作?
 - マウスのボタンが押されていない時はどういう動作?

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
}

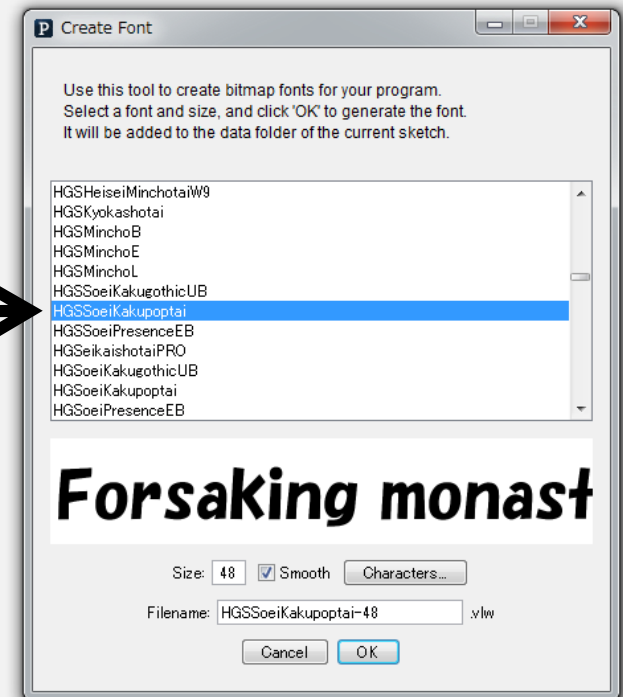
void draw() {
  fill(0);
  text( "Processing", 20, 90 ); // 文字を表示
}
```

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。

今日使うテクニック(音編)

- 音を鳴らすにはminimというものを利用する(詳しくは12回目で)
- 今日のテクニックを実現するにはおまじないが必要です
- 適当なwavやmp3ファイルをプログラムの上にドロップし下記のコードを実行

```
import ddf.minim.*;
Minim minim;
AudioSnippet crash;
void setup(){
  minim = new Minim( this );
  crash = minim.loadSnippet( "crash.mp3" );
}
void stop(){
  crash.close();
  minim.stop();
  super.stop();
}
```

赤文字は毎回同じおまじない
青文字は音に応じて利用

音を再生する部分に挿入

```
crash.rewind();
crash.play();
```

今日使うテクニック(音編)

マウスクリックの時に
クラッシュ音を鳴らす

crash.mp3 は事前に
ダウンロードし、
プログラムの上に
ドロップしておく！

```
import ddf.minim.*;
Minim minim;
AudioSnippet crash;
void setup(){
  size( 400, 400 );
  minim = new Minim( this );
  crash = minim.loadSnippet( "crash.mp3" );
}
void stop(){
  crash.close();
  minim.stop();
  super.stop();
}
void mousePressed(){
  crash.rewind();
  crash.play();
}
```

複数の音を使う場合
変数として定義を
繰り返すだけ！

mp3やwavもOK!!

```
import ddf.minim.*;
Minim minim;
AudioSnippet taiko;
AudioSnippet kane;
void setup(){
  size( 400, 400 );
  minim = new Minim( this );
  taiko = minim.loadSnippet( "taiko.mp3 " );
  kane = minim.loadSnippet( "kane.wav" );
}
void stop(){
  taiko.close();
  kane.close();
  minim.stop();
  super.stop();
}
void mousePressed(){
  if( mouseX < width/2 ){
    taiko.rewind();
    taiko.play();
  } else {
    kane.rewind();
    kane.play();
  }
}
```