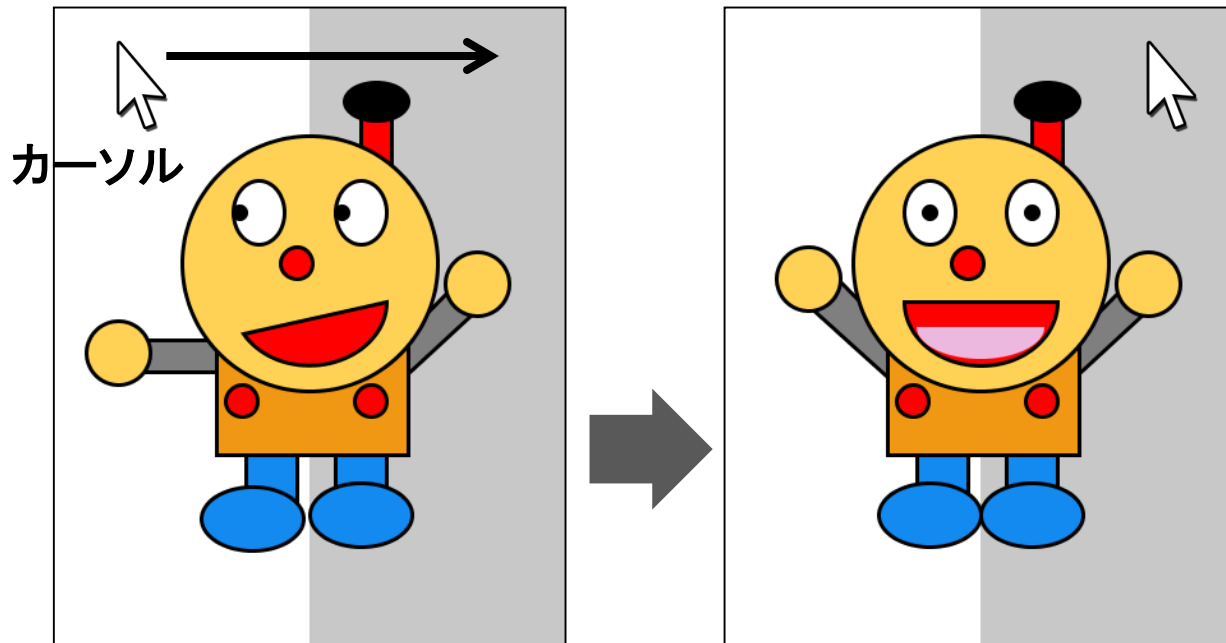


プログラミング演習I (第4回) 課題

• 基本課題① スケッチ名 : charapara

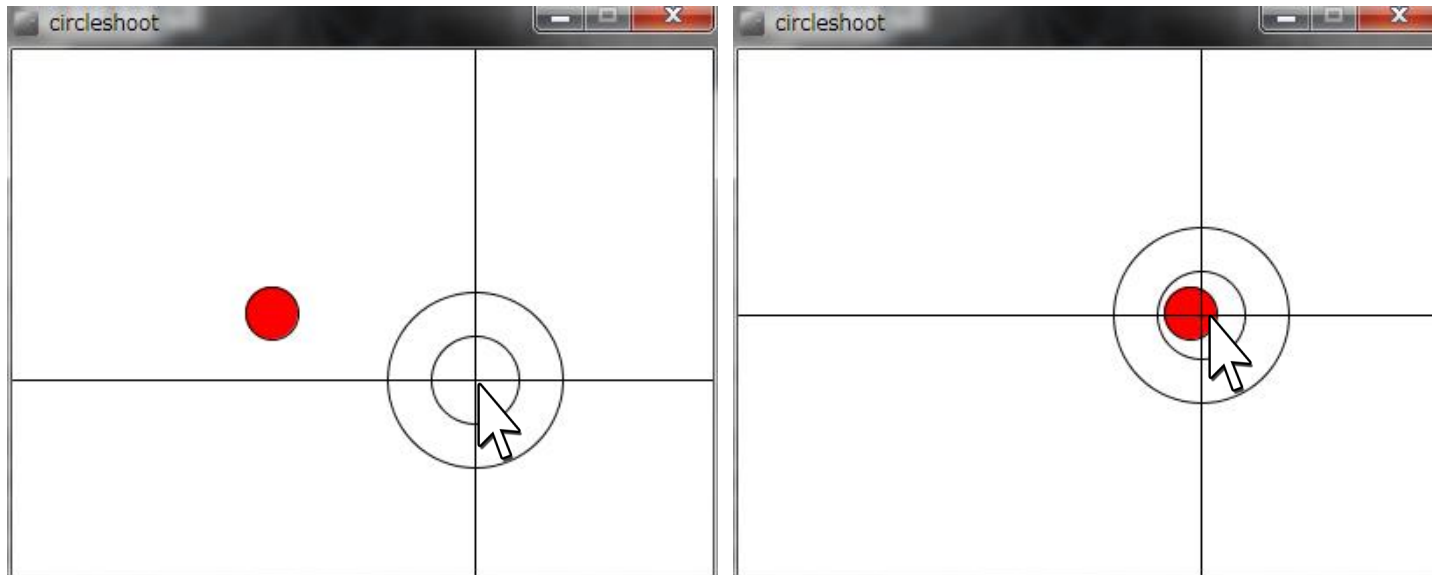
- キャラクタを表示し、カーソルの位置によって絵が変わるプログラムを作ってください。
- カーソルが 中央より左側 にある時と 中央より右側 にある時とでキャラクターの表情とポーズが変わるようにすること。
- 前々回のchara1とchara2のプログラムをうまく再利用しよう。
- 背景は灰色にする必要はありません。



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 : circleshoot

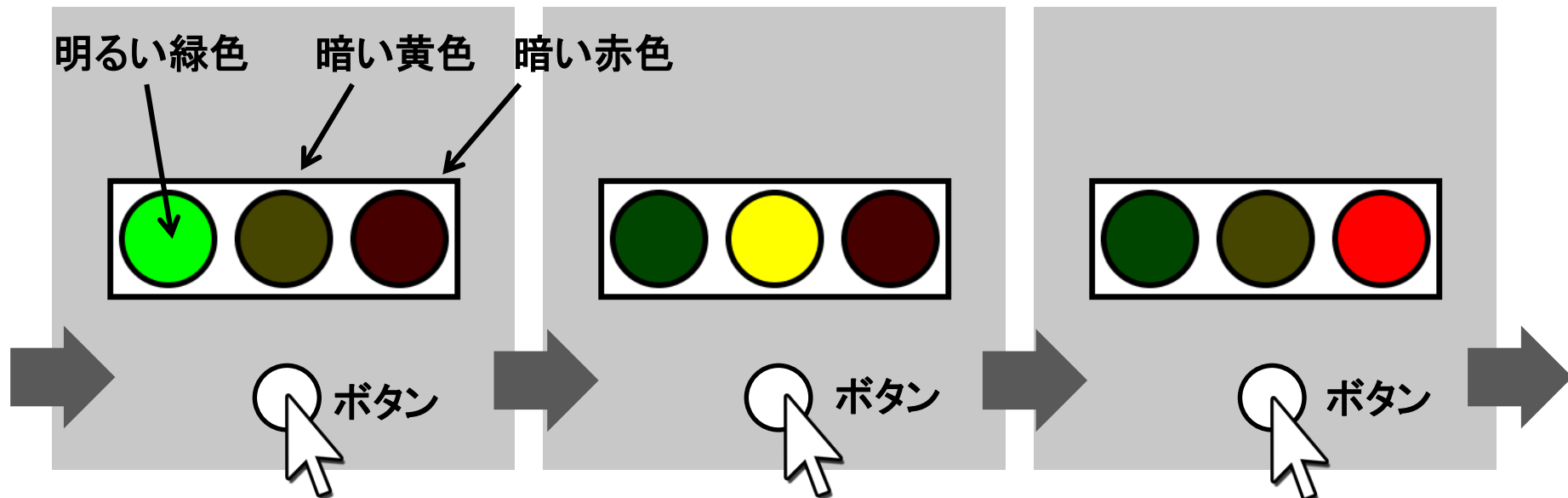
- 400x300の白背景のウィンドウ内に、中央から毎フレーム2ピクセルずつ右／左方向に移動する直径が30の赤丸を描け。また、両端に来ると跳ね返るようにせよ(跳ね返り判定は赤丸の中心でもよい)。
- 赤丸の上にカーソルを持って行くと動きを止め、カーソルが円から離れるとまた動くようにせよ。
- カーソルの周辺に照準(半径25と50からなる円とウィンドウ端まで突き抜けた十字線)を表示せよ。なお、照準は赤丸の手前に表示せよ。



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 : signal

- 状態を示す3つの丸と、それを囲む四角形(土台)からなる信号機、信号機を操作するボタンからなるプログラムを作ってください。また、操作ボタン内をマウスクリックすることによって光る丸が緑→黄→赤→緑→... とシフトしていくようにしてください。ボタン外を操作しても反応しないようにしなさい。



丸ボタンの中をクリックしたら色が変わる！

赤の次はまた緑色に戻るように！

考え方

- この世界を実現する上で必要な変数をまず列挙！
 - 最初は多すぎるぐらい列挙してよい！
- 変数を定義！
- 変数を利用して世界を描画する！（適当に初期値を入れてみる）描画の準備をしっかりと考える！
- 必要な条件分岐を整理しどこに入れるかと、その条件分岐においてどんな変数をどう変化させるかを考える
- 条件分岐をプログラム内に入れていく
- 完成を目指して色々入れていく

プログラミング演習I (第4回) 課題

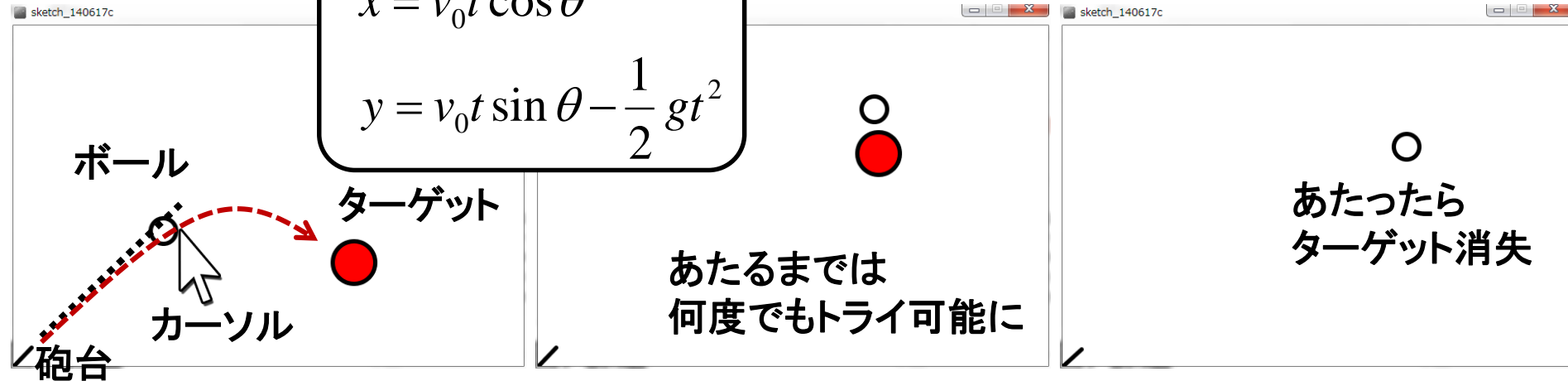
発展課題① スケッチ名: launch

- 600x400のウィンドウ左下からマウスカーソルがある方向にボール(半径15m)を発射する砲台のプログラムを作ってください。クリックされたタイミングで投射されるようにすること。また、クリック時の砲台からカーソルまでの距離を初速とせよ。さらに、400m先に半径25mの赤丸のターゲットを描け。赤丸は上下に2m/フレームで動き、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、ターゲットを消すようにせよ。1フレームは0.1秒とする。

斜方投射の式

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

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



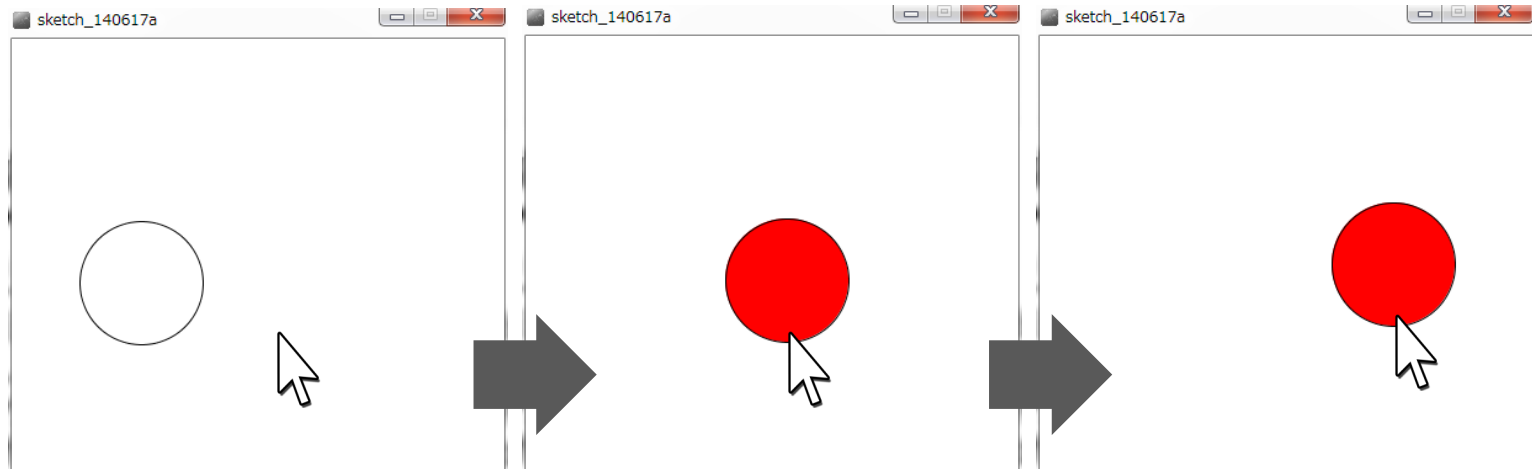
ヒント

- この世界を実現する上で必要な変数(変化するものを取りあえず片っ端から)をまず列挙！
 - マウスの座標
 - 砲台の角度
 - ボールの投射角度
 - ボールの初速
 - ボールの座標
 - ターゲットの座標
 - ターゲットの速度
 - ターゲットの存在フラグ
 - 時間経過を管理する変数
 - ほかにある？
- その変数によって世界を描画してみる！
- 変数に, 計算などによって値を入れてみる
- 跳ね返るといふことはどういう計算があればよい？

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 : dragdrop

- 400x400の画面上に半径が50の白色背景, 黒縁の円を画面中央に描画し, マウスボタンを円の中で押すと円の縁が赤色に変化し, マウスボタンを離すまでその色を維持するようにせよ. また, ボタンを離すと元の色に戻るようにせよ.
- また, マウスボタンを押した状態でマウスを移動すると, 円の位置がマウスに応じて移動するようにせよ. なお, 円のどの部分をマウスで掴んでいるのかということを維持するようにせよ(端で掴んだら, そのまま端を中心に平行移動)



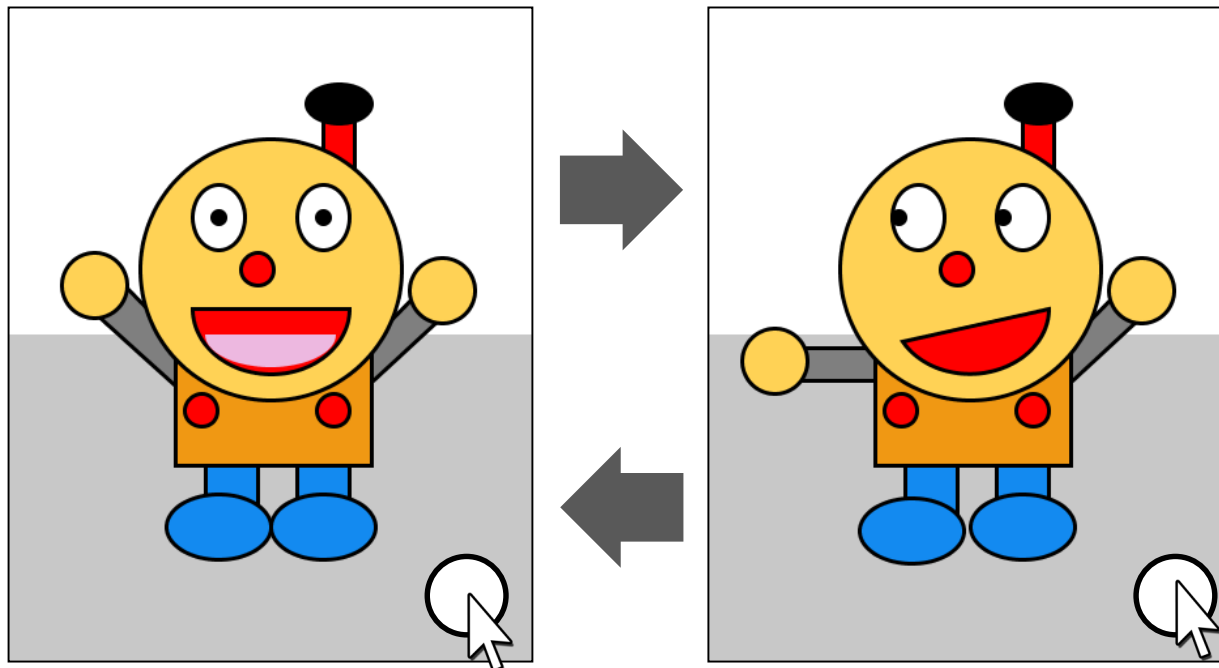
プログラミング演習I (第4回) 課題

- 必要な変数
 - 円の中心座標
 - 選択されているかどうかのフラグ
 - 選択時の円の中心からの相対座標
- 利用するメソッド
 - mousePressed() マウスのボタンを押す
 - 円の選択の判定とフラグ付与
 - 円の選択時の中心からの相対座標
 - mouseReleased() マウスのボタンを離す
 - 円の選択のフラグを離す

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 : charapara

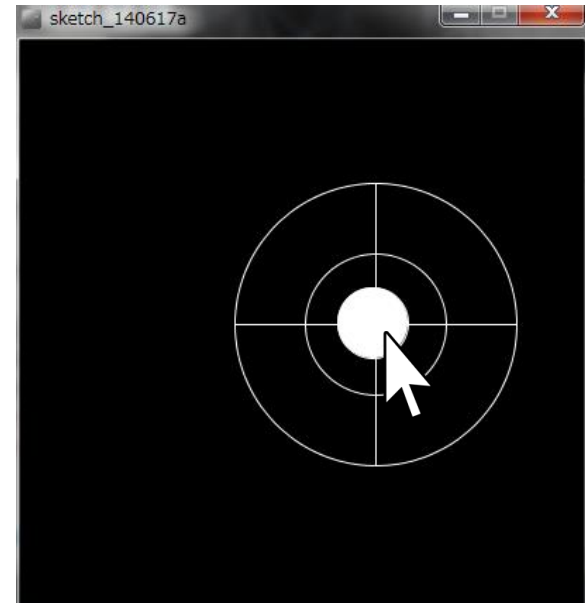
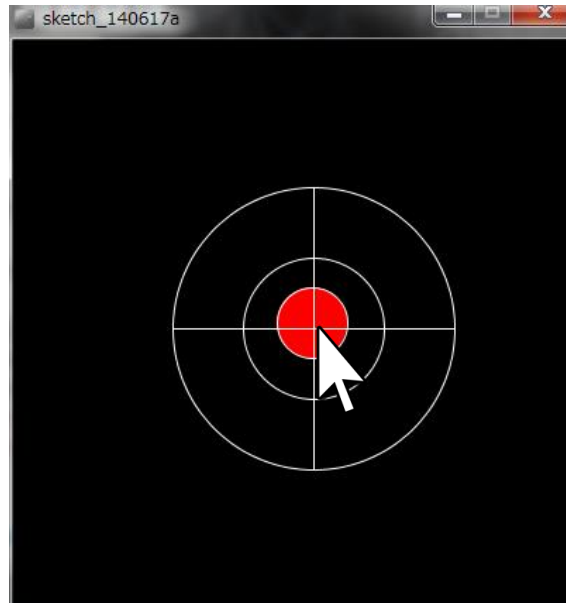
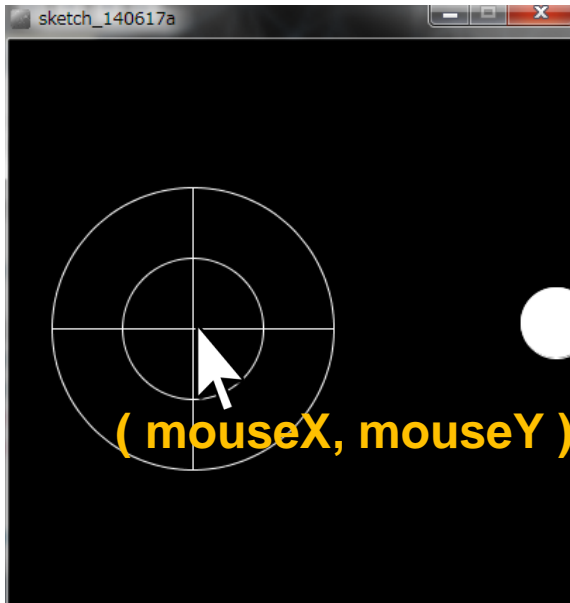
- キャラクタを表示し、画面右下に用意した白色の直径50以下の丸ボタンをクリックすることで絵がパラパラと切り変わるプログラムを作ってください。また丸ボタンの外をクリックした場合は反応しなくなるようにせよ。
- 前々回のchara1とchara2のプログラムをうまく再利用しよう。



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 : **circleshoot**

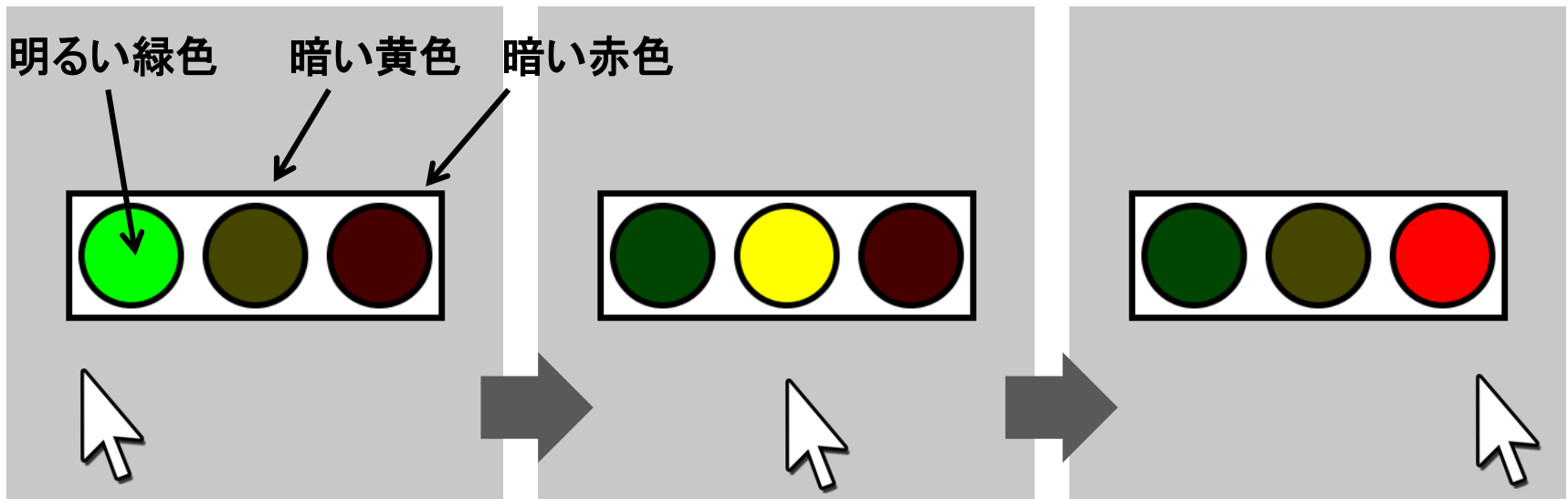
- 400x400のウィンドウ内で, 画面中央から毎フレーム2ピクセルずつ右方向に移動する直径が50の円が右端に來ると左方向に跳ね返り, 同じく左端に來ると右方向に跳ね返るようにせよ. また, マウスカーソルを中心に照準(半径50と100からなる円と十字線)を表示せよ. さらに, カーソルで円の内部をクリックすると赤色に, 再度クリックすると白色に戻るようにせよ



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 : signal

- 信号機のプログラムを作成せよ。ただし、画面を左右に3等分し、マウスカーソルが左側にある場合は緑色、中央にある場合は黄色、右側にある場合は赤色が明るく灯るようにし、そうでない色は暗くなるようにせよ。



マウスカーソルの位置に応じて色が変わる！

考え方

- この世界を実現する上で必要な変数をまず列挙！
 - 最初は多すぎるぐらい列挙してよい！
- 変数を定義！
- 変数を利用して世界を描画する！（適当に初期値を入れてみる）描画の準備をしっかりと考える！
- 必要な条件分岐を整理しどこに入れるかと、その条件分岐においてどんな変数をどう変化させるかを考える
- 条件分岐をプログラム内に入れていく
- 完成を目指して色々入れていく

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: launch

- 画面左下からマウスカースルがある方向にボールを発射する砲台のプログラムを作ってください。
- クリックされたタイミングで投射されるようにすること。また、初速はクリック時の砲台からカーソルまでの距離にすること。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。跳ね返りの場合はそのまま左右方向を反転させるだけでよい。draw1回あたり0.1秒ずつ進むものとする

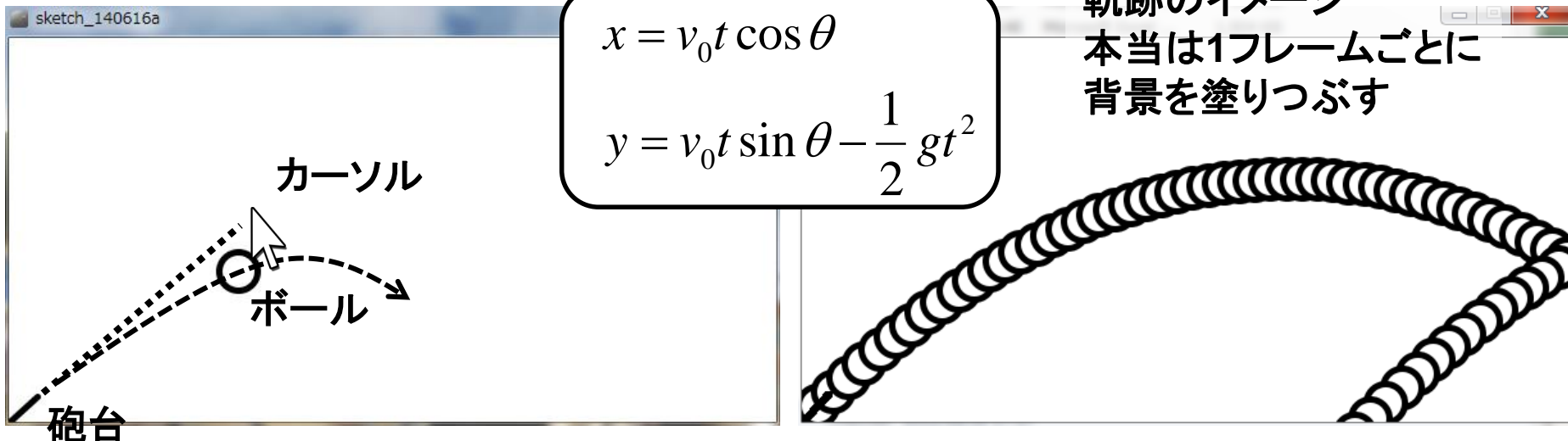
斜方投射の式

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

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

軌跡のイメージ

本当は1フレームごとに背景を塗りつぶす



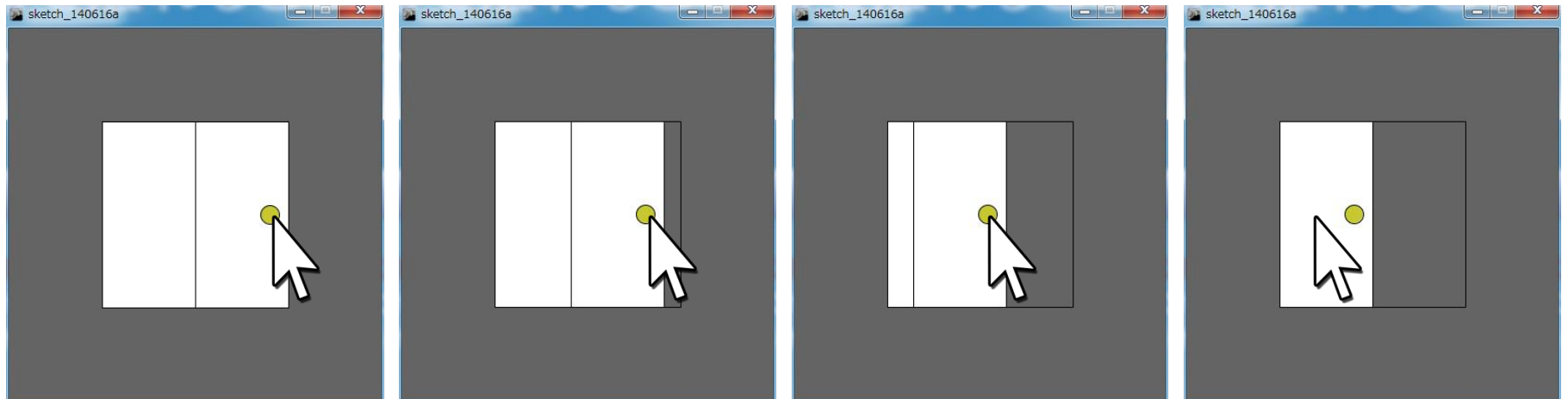
ヒント

- この世界を実現する上で必要な変数(変化するものを取りあえず片っ端から)をまず列挙！
 - マウスの座標
 - 砲台の角度
 - ボールの投射角度
 - ボールの初速
 - ボールの座標
 - 時間経過を管理する変数
 - ほかにある？
- その変数によって世界を描画してみる！
- 変数に, 計算などによって値を入れてみる
- 跳ね返るということはどういう計算があればよい？

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: dragdrop

- 400 × 400 の灰色のウインドウに100 × 200の白色の扉を2枚並べて書き、片方の扉に取っ手を描き(半径30程度)、取っ手をカーソルで掴みながら動かすと下図のように扉をスライドして開ける事のできるプログラムを作成せよ。なお、取っ手を掴んだらそれ以降は取っ手からずれても良いものとする。また動かす扉を最初的位置から右へや左の扉を超えて動かないようにする事(課題案: 豊崎富心)



行き過ぎても左端でとまる

プログラミング演習I (第4回) 課題

- 必要な変数

- ドアの取っ手の中心座標

- ドアの取っ手が選択されているかどうかのフラグ

- 他に何かある？

- 利用するメソッド

- `mousePressed()` マウスのボタンを押す

- ドアの取手の選択の判定とフラグ付与

- `mouseReleased()` マウスのボタンを離す

- ドアの取手の選択のフラグを離す