

コンピュータ基礎演習

第13回

理工学部 情報科学科 隅田 康明

sumida@ip.kyusan-u.ac.jp

エコバッグデザイン募集について

- 令和3年度 入学式用エコバッグデザイン募集のお知らせ
 - <https://www.kyusan-u.ac.jp/eas/news/12375/>
 - <https://www.kyusan-u.ac.jp/eas/app/wp-content/uploads/2020/06/2021leaflet.pdf>
 - 応募期間：6月22日（月）～9月25日（金）
16：30まで
- （勝てるかどうかは別として）
第6回や第9回のアレンジで割と簡単にデザインを作れます
 - Processingでパーツ作成、Illustratorで加工修正
- 1人5点まで提出可能なので、
Processingで1点作って応募してみてくださいはどうか

対面/遠隔での質問受付体制

- 7/13から教室で授業時間に対面でも質問受付開始予定
 - 本来の時限の受講者のみ入室可
 - 入室前に手洗い、消毒、座席表への記入
 - 私語厳禁、座席は完全固定
 - 指示に従わない場合には減点や欠席措置
 - 質問は遠隔でも受け付ける
 - Zoomの画面を教室のプロジェクターに投影
 - Zoomでの個別対応は難しくなるかも知れません

詳細はK'sLifeで連絡の後に、Moodleに掲載

制作課題レポートの作成

- 開発したプログラムをレポートでまとめる
 - PowerPointで作成
 - その他でも形式があてれば認めることもあるが、教員側で開けないファイルは不可
 - 特に、プログラムをコピーしてそのまま動かさないものは×
 - つまり、指定に従わないなら自己責任
- スライドの構成
 - タイトル：レポートのタイトル、学籍番号、氏名等
 - 作品紹介：プログラムの実行画像と説明文
 - 工夫点：プログラムの工夫を説明
 - 何枚でも可、工夫点は最低3つ
 - 工夫点3つではいい評価は取れない（場合によっては単位も危ない）

レポートの締め切り (減点なしで受け付ける期間)

- 制作課題：最終回授業日の18:00まで
 - **7/24 (金), 18:00まで**
- 追加課題：7月24日 18:00まで
- 遅れ提出の締め切り：**7/24 18:00**まで
 - Moodleでの提出も打ち切り
 - 以降は特別に認めた場合を除いて受け付けない
 - 最終的なレポート締め切りと考えて問題ない
 - **最終講義日の翌日が最終締め切り、注意すること**

遠隔授業期間中の質問

- **困ったら早めに質問・相談！！**
- プログラムが完成していない人
 - 来れる人は対面教室に来ましょう。
 - この期に及んで質問しないのであれば、単位を落とすことも覚悟しましょう。
- もう質問を躊躇っている余裕は一切ありません
- 少しでも困ったことがあるなら、四の五の考える前にさっさとメールを送ること

授業についての質問メールについて

[授業名(曜日時限)]についての質問	} 件名
～先生	} 誰宛か
[授業名(曜日時限)]を受講しています、 20AA999の九産太郎です。	} 何者か
(質問内容)	} 用件
--	
20AA999 九産太郎 九州産業大学 芸術学部 ○○学科 1年	} 署名

細かいことはいいから、困っているならさっさと質問しなさい

今後、なんの役に立つのか？

- プログラミング自体は今後、必須の教養
 - 小学校でプログラミング科目が導入される
- PCで絵を描くときの道具が一つ増える
 - ちょっとしたイラストをProcessingで
 - 背景をProcessingで
- Webページ作成での理解が楽になる
 - javascriptアニメーションは、Processingとよく似ている
(来年以降の講義で使えるかも)
- ジェネラティブ、インタラクティブなアートが身近に
- ゲーム作りでもプログラミングは必須
 - Unity等のゲームエンジンで使われるC#も、書き方はProcessingによく似ている

最終回のレポート

- 制作課題レポートの提出
 - 第11回と12回で開発したプログラムの内容をレポートでまとめる
 - これまでのレポートの形式とは異なるので、注意すること
- 締め切りは7/24(金)18:00まで

レポート見本

コンピュータ基礎演習 制作課題レポート

学部 学科

学籍番号 氏名

作品タイトル：迷路



- 当たり判定を利用して迷路を作った
- 左上の黒い丸がスタート、青い丸をゴールにした
- その他の丸に触れるとゲームオーバー
- 迷路のコースを作るのが面倒だったので、コースは自動生成方式にした

実行画像と作品の概要を簡単に説明

5個(5セット)以上は書かなくてもいい

変数説明

- ゲーム画面を切り替えを管理する変数
 - `int gameMode = 0;`
 - スタート画面=0,ゲーム画面=1,ゲームオーバー画面=2,クリア画面=3
- ゲームの得点を管理する変数
 - `int score = 0;`
- 自キャラの座標を管理する変数
 - `int mx=10,my=10;` //自キャラのxy座標
 - `int mvx,mvy;` //自キャラの動く速度

最低5個または5セットの変数を説明
(1つの図形用の変数はまとめて1セット)

配列説明

- 障害物になる丸の座標を管理する配列
 - `int[] ellipseX = new int[1000];`
 - `int[] ellipseY = new int[1000];`
- 障害物になる丸のサイズを管理する配列
 - `int[] ellipseR = new int[1000];`
- 障害物になる丸を描画するかを管理する配列
 - `boolean[] drawEllipse = new boolean[1000];`
- 障害物になる丸を描画するかを管理する配列
 - `color[] ellipseColor = new color[1000];`

変数がなく、配列の場合(配列の場合は5個なくても可)

メソッド説明

- drawStart
 - スタート画面を描画する処理をまとめたメソッド
- drawGame
 - ゲーム画面の処理をまとめたメソッド
- makeEllipses
 - 障害物になる丸の座標、サイズ、色を設定するメソッド
- drawMe
 - 自分で動かすキャラクターを描くメソッド

作ったメソッドの機能を簡単に説明

障害物を左右に動かす工夫

- 左右に動く障害物を作った
 - 障害物の変数は $t1x, t1y, t1Vx, t1Vy$
 - $t1x$ が400より大きくなったら、 $t1Vx$ に-1を代入して左方向に動くようにした
 - $t1x$ が0より小さくなったら、 $t1Vx$ に1を代入して右方向に動くようにした

工夫点の内容を文章で説明


```
if(t1x > 400){  
    t1vx = -1;  
} else if(t1x < 0){  
    t1vx = -1;  
}
```

工夫点のプログラムを貼り付ける
(工夫点が短ければ、前のスライドに貼ってもいい)

流れ星を斜めに動かす工夫

- 流れ星に見立てた丸を右上から左下に動かした
 - 流れ星の変数は $t2x, t2y$
 - $t2x$ を -1 することで左に動かした
 - $t2y$ を $+1$ することで下に動かした

$$t2x = t2x - 1;$$
$$t2y = t2y + 1;$$

短い工夫点でも書いて良い

マウスに追従して自キャラを動かす工夫

- 条件分岐を利用して、マウス座標に追従して動くようにした
 - 自キャラのx座標(mx)がマウスのx座標よりも大きければ、 mx を-1にし、マウスのx座標より小さければ、 mx を+1にした
 - 自キャラのy座標(my)がマウスのy座標よりも大きければ、 my を-1にし、マウスのy座標より小さければ、 my を+1にした
 - 自キャラの移動はdrawMeメソッドにまとめて分かりやすくした

```
void moveMe() {
    if(mouseX > mx){ //もしマウスの座標が自
        キャラより右なら
            mvx = 1; //速度を1にする
        } else if(mouseX < mx){ //もしマウスの座標
        が自キャラより左なら
            mvx = -1; //速度を-1にする
        } else { //そうでないなら(同じ座標なら)
            mvx = 0; //速度を0にする
        }

        //y座標についても同様
        if(mouseY > my){
            mvy = 1;
        } else if(mouseY < my){
            mvy = -1;
        } else {
            mvy = 0;
        }
        mx += mvx; //速度を自キャラの座標に足す
        my += mvy;
    }
```

前のページの工夫点のプログラムを貼り付ける

自キャラを描画するメソッド

- 2つの丸を組み合わせた、自キャラを描画するメソッド drawMe を作った
 - キャラクターの中心が x, y になるように配置して、当たり判定で利用できるように工夫した。
 - （中心点を中心にするように配置した）

プログラム貼り付け
（長くなるなら次のページに）

感想(感想や反省点など)

- 制作課題のプログラミングをしてみた感想を自由に
- 反省点
 - もっとこうしておけば良かった、こんな機能を入れてみたかったなど、何かあれば

プログラム

```

    ellipse(0,0,200,200);
    fill(0,0,255); //障害物の色は真っ白と真っ
青以外
    ellipse(width,height,200,200);
    //得点を計算し、表示する
    dispScore();
    score++;

```

いつものように、プログラム全文を貼り付ける

```

} はみ出しても気にしない
//ゲーム画面を描画
void drawGame() {
// 見た目はどうでもいい
dr

```

**プログラムを画像で貼り付けた場合は再提出
(締め切りに間に合わなければ不可判定)**

り判定、ゲームモードを切り替える

レポート内容について

タイトルスライド（1枚目）

- タイトルスライド
- タイトル：コンピュータ基礎演習 制作課題レポート
- サブタイトル：学籍番号、氏名、学部、学科を記入
- 必要事項が入力されていれば形式は問わない

作品紹介（2枚目）

- 作品を紹介するスライド
- 自分の作った作品のタイトル
- 作品の簡単な紹介を書く
 - 紹介文は自由に書いていい
- 1枚に収まらないなら、複数枚に分けて書いてもいい
 - 1枚増えるごとに+0.5点（5枚まで）
 - 5枚異なる実行画像を貼れば最高2.5点
 - ただし、それぞれ違う画像にすること

変数・配列・メソッド説明

- 宣言した変数や配列、メソッドの説明をする
 - 変数の役割と、変数名をセットで紹介
- 全ての変数等の説明はしなくてもいい
 - 全部書くと大変なので
 - 変数、配列、メソッド、合わせて5つの説明をすること
 - 普通の変数が1つもない場合は、配列とメソッド合わせて3つの説明で良い
 - 配列、メソッドはあれば説明する、無いと駄目なわけではない
- 説明が5個以上で1つにつき+0.5点(5個まで)
 - 変数説明が10個で最高2.5点

工夫点紹介

- プログラムの工夫点を紹介していく
 - まず文章で工夫点の内容を説明
 - 工夫したところを教員にアピールするつもりで
 - つぎに、工夫点のプログラムをスライドに貼り付け
 - プログラムすべてではなく、その工夫点に関する部分だけを貼り付ける
 - 長い場合には、スライドからはみ出してもいい
- 簡単なことでも良いので、最低3つの工夫点を書くこと
 - 工夫点が3つ無いものは再提出
 - 期限内の再提出が間に合わないなら不可となる
 - 同じ工夫はカウントしない(同じような工夫も×)
 - 右に動かす工夫を5つ書いても、1つのみカウント

感想

- 制作課題に取り組んだ感想や反省点を自由に
- 授業を通しての感想を書いても良い

プログラムスライド

- プログラムの全文を貼り付ける
 - 画像で貼り付けてはいけない
- 見た目が汚くなっても気にしない
 - 気になる人は綺麗に整形しても良いが、代わりにオンラインテキストにもプログラム全文を貼り付けること
 - 必須ではないが、全員、オンラインテキストにもプログラムを貼り付けること
 - プログラムスライドを綺麗にすることによって、点数が上がることはない



レポートの評価と提出

レポートの評価

- 基準点：20点
 - 最低基準を満たしたレポートを提出
 - 必要項目が不足することにより減点
- 工夫点：0～20点
 - 工夫点に下記が含まれるとそれぞれに加点
 - 0.5点：変数、背景を描いた、色を工夫した、座標を調整
 - 1点：条件分岐、繰り返し、メソッド、配列、座標変換
 - 工夫点は重複して計算
 - 例) **メソッド**内で**繰り返し**を使って**座標変換**を利用した**条件分岐**で変化する図形を描き、そのメソッドを**繰り返し**で複数描画した場合は、メソッド + 繰り返し + 座標変換 + 条件分岐 + 繰り返しで5点の工夫
 - さらに、変数の工夫などが混ざると点数UP
 - 工夫点を上手く説明出来れば、1つの工夫点で10点程度取ることも出来る（質より量でも勿論問題ない）

その他の加点(やりたい人だけ)

- 画像の加工
 - 作品紹介スライドの画像を画像加工アプリなどで加工
 - 元の絵は未加工で貼ること
 - 1種類につき0.5点の加点 (最大3回まで)
 - 加工したアプリの紹介スライドを追加
 - アプリのどの機能を使って加工したのかを説明
 - Processingでも実現出来そうかどうかの考察をする
 - 1種類につき1点の加点 (最大3回まで)
- スライドの工夫 (これも工夫点スライドで説明すること)
 - スライドの背景にProcessingで描いた絵を使う
 - アニメーションで工夫点を説明
 - 等、内容に応じて最大3点の加点

画像加工 1 (例)



- PowerPointの機能で加工した。
- 図のスタイルを面取り楕円にして、「パッチワーク」のアート効果を付けた

画像加工 2 (例)



- iPhoneアプリの「Artomaton」という写真加工アプリで画像を加工した。
- AIが筆や鉛筆、マーカーなど様々なタッチで写真を書き直してくれる
- Processingで同じことは何とか実現出来そう (AIこの絵のようにするだけなら)
 - 丸の位置を避けてある程度の規則性を持ったランダムな線を引いていけばいい?
 - 具体的にどうすれば出来そうか、も記載すること

レポートの提出方法

- いつも通り、Moodleで提出
 - オンラインコメントにもプログラムを貼り付けておく
- 特にいつもと手順は変わらない
- ただし、共有リンクのコピーを間違えないように！！
 - PCで提出する場合は、ファイルのアップロードもしておくこと
 - 両方しておくことと安心

最低限のレポート例

他のレポートや小テストがある程度できているなら、
何とか単位を取れるかも・・・程度の内容

いい成績は取れない、場合によっては単位を落とす
(ギリギリを狙うと大抵評点が足りなくなる)

作品タイトル：〜〜〜

動く図形



動かない図形



- 〜〜の背景の上で図形をたくさん動かした

実行画像と作品の概要を簡単に説明

変数説明

- 右に動く丸の変数
 - float x1, y1;
- 左に動く四角の変数
 - float x2, y2;
- 左上に動きながら大きくなる丸の変数
 - float x3, y3, s3;
- 右端まで行ったら元の場所に戻る四角の変数
 - float x4, y4, vx4, vy4;
- 右下に動く三角形の変数
 - float x5, y5;

最低限、5個分の変数宣言部分

工夫点 1 : 右端まで行ったら元の場所に戻る四角

- 右端まで行ったら、最初の位置に戻る四角形をかいた

```
fill(255, 255, 0);  
rect(x5, y5, 10, 10);  
x5 += vx5;  
if (x5>360) {  
  x5 = 0;  
}
```

変数、条件分岐 : +1.5

工夫点2：左上に動きながら大きくなる丸

- 移動するだけでなく、大きさも変わる図形を作った
- 丸の幅と高さを両方s4にして、正円が大きくなるようにした
- `fill(x4, y4, s4, 400-y4);` で、動きに合わせて色が変わるようにした

```
fill(x4, y4, s4, 400-y4);  
ellipse(x4, y4, s4, s4);  
x4+=vx4;  
y4+=vy4;  
s4+=vs4;
```

変数(座標)、変数(サイズ)：1点

工夫点3：背景を描いた

- 動く図形を描く前に、背景になる絵を描いた
- 先に背景を描くことで、背景の上で図形を動かした

```
rect(100,100,60,200);  
fill(100,58,240);  
rect(200,200,60,50);
```

背景：0.5点

これはさすがに短すぎるが、
考え作られた絵であれば+1点くらいは付けられる
(色と座標の工夫があれば0.5点ずつ)

工夫点4：右端まで行ったら元の場所に戻る丸⁴⁶

- 右端まで行ったら、最初の位置に戻る丸をかいた

```
fill(255, 255, 0);  
ellipse(x6, y6, 10, 10);  
x6 += vx6;  
if (x6>360) {  
  x6 = 0;  
}
```

同じ工夫3つ目はカウントしない

感想

プログラム

```
fill(0, 255, 255);
rect(x2, y2, 10, 10);
y2 += 1;
```

```
fill(255, 0, 255);
triangle(x3, y3, x3+10, y3+10, x3-40, x3+50);
x3++;
y3++;
```

5 ～ 6 個の図形を動かすだけでも、プログラムははみ出すはず

```
x4+=vx4;
y4+=vy4;
s4+=vs4;
```

内容にもよるが、条件分岐・繰り返し・メソッド・配列などが無いプログラムの場合、100行程度無いと手抜きと見なす可能性が高い
(区切りの改行は含めてもいい(改行除いて80行程度?))

```
if (x5>360) {
  x5 = 0;
}
```

```
fill(255, 0, 255, 50);
```


最低限のレポートの場合の点数

- 基準点：20点
 - ここでミスがあれば単位を落とす
- 工夫点：3.5点
- 合計23.5点
 - 背景を頑張って書けば+1～2点くらいは取れる
 - そうすると24点（6割）取れることになる
 - 小テスト、他のレポート6割でギリギリ可となるライン
 - この辺りもさぼっていたなら単位を落とします
- **ただし、最低限の場合、出し直しを指示される可能性が高いものと考えておくこと**

レポートチェックリスト（第13回）

- ミニテストを受験した(レポートの前と後どちらでも)
- PowerPointでレポートを作成した
 - タイトルスライドにタイトル、学籍番号、名前、学部学科
 - 作品紹介スライドに画像を貼って説明を書いた
 - 変数紹介スライド(または配列)を書いた
 - 工夫点スライドが3セット以上ある
 - プログラムを画像で貼り付けていない
 - 感想スライドに、感想を書いた(何もないなら無しで良い)
 - プログラムスライドに**プログラムを文字で**貼り付けた
 - 画像で貼り付けていない
- レポートを提出した
 - Moodleに提出した

可となる最低ラインについて

- 小テスト6点以上（10点取りましょう）
 - 何回か解けば必ず満点に出来る、満点を取れないのはただサボっているだけ
- 1～12回のレポート6割以上
 - 1回～3回で11点：出していれば11点取れている
 - 11回、12回で4点：出せば4点取れる
 - ここまでで21点
 - 残り7回で15点取れていればギリギリ単位習得ライン
 - 3点×5回で単位が取れる
- 第13回：24点(6割)
 - ギリギリ手抜きと判定されない程度のプログラムとレポート
- **ただし、ギリギリを狙うと大抵落ちる**
 - **少なくとも5点程度は余裕を持つておくこと**

単位を落とす例

- 不正（まず間違いなくバレます）
 - インターネット上のプログラムや、Processingの参考書のプログラムをコピー
 - ただし、事前に相談があればアレンジを認めることもある
 - 友人と(殆ど)同じプログラムを提出
 - 全員0点にします
- 手抜き
 - 雛型やサンプルと同じプログラムを提出
 - 当然、0点です
 - レポートと同じプログラムを提出
 - その回のレポートに多くの時間をかけたのであれば、それなりの点数は取れるが、元のレポートが手抜きなら単位を落とす
 - そもそも、プログラムが短い
 - 最低でも200分程度の努力が見えない場合は0点
 - 短いプログラムでも、すごい工夫をしてきた場合は除く
 - まず無いケース。もしあれば追加で説明を求めます
- 不正が疑われるプログラムを提出してきた場合は、電話で試問を行います(自分のプログラムを説明できない場合は不可とする)
 - 意図して不正を行わない限り、こんなことにはなりません