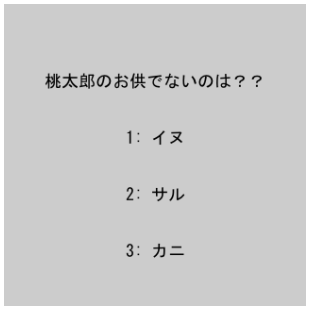
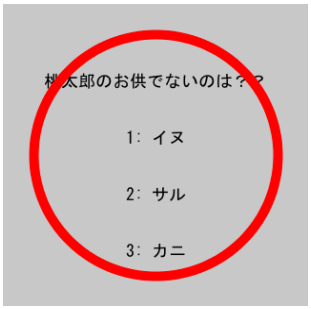
プログラミング演習I
クイズをつくる：1

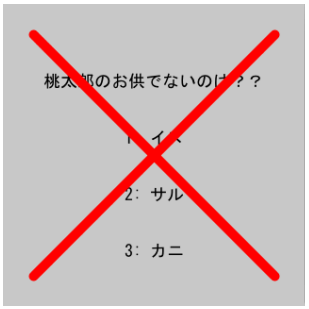
担当：小久保温

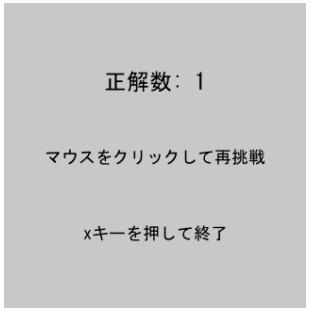
仕様（ルール）と
外部設計（見た目）

- 開始画面
  - タイトル
  - スタート方法
- プレイ画面
  - 問題文と選択肢3つ表示
  - 1、2、3のキーで回答
  - 正解で◯、間違いで×を表示
- 結果画面
  - 正解数を表示
  - リトライ、終了を選択









ゲームの枠になるプログラムを作成
quiz-1.0.zip

メインのタブ

```
void setup() {
}

void draw() {
}
```

openingタブ

```
void opening() {
}
```

playingタブ

```
void playing() {
}
```

resultタブ

```
void result() {
}
```

※この段階で実行してエラーがないことを確認する
エラーがなければ保存しておこう

メインのタブに全体の構造を作る

quiz-1.1.zip

メインのタブ

```
int scene; // シーン番号

void setup() {
  size(320, 320);
  scene = 0; // シーン番号を0に
}

void draw() {
  switch(scene) {
  case 0:
    opening(); // 開始画面
    break;
  case 1:
    playing(); // プレイ画面
    break;
  case 2:
    result(); // 結果画面
    break;
  default:
    break;
  }
}
```

開始画面を作る
quiz-1.2.zip

openingタブ

```
void opening() {
  background(204); // 背景色
  fill(0); // 塗り(文字)の色
  textAlign(CENTER, CENTER); // 文字列を上下左右中央揃え

  text("3択クイズ", 40, 40, 240, 80);
  text("クリックして開始", 40, 240, 240, 40);

  if (mousePressed == true) {
    // マウスをクリックしたら
    scene = 1; // プレイ画面へ移動
  }
}
```

フォントの生成
quiz-1.3.zip

メインのタブ

```
int scene; // シーン番号

PFont font18; // 18ピクセルのフォント
PFont font36; // 36ピクセルのフォント

void setup() {
  size(320, 320);
  scene = 0; // シーン番号を0に

  // フォントの生成
  font18 = createFont("MS Gothic", 18);
  font36 = createFont("MS Gothic", 36);
}

[以下略]
```

フォントを変える
quiz-1.4.zip

openingタブ

```
void opening() {
  background(204); // 背景色
  fill(0); // 塗り(文字)の色
  textAlign(CENTER, CENTER); // 文字列を上下左右中央揃え

  textFont(font36); // 32ピクセルのフォントに
  text("3択クイズ", 40, 40, 240, 80);

  textFont(font18); // 18ピクセルのフォントに
  text("クリックして開始", 40, 240, 240, 40);

  if (mousePressed == true) {
    // マウスをクリックしたら
    scene = 1; // プレイ画面へ移動
  }
}
```

プレイ画面(仮)を作る
quiz-1.5.zip

playingタブ　※openingをコピーして書き換えると楽かも！

```
void playing() {
  background(204); // 背景色
  fill(0); // 塗り(文字)の色
  textAlign(CENTER, CENTER); // 文字列を上下左右中央揃え

  textFont(font36);
  text("プレイ画面", 40, 40, 240, 80);

  textFont(font18);
  text("キーを押して結果画面へ", 40, 240, 240, 40);

  if (keyPressed == true) {
    // マウスをクリックしたら
    scene = 2; // 結果画面へ移動
  }
}
```

結果画面を作る
quiz-1.6.zip

resultタブ　※openingをコピーして書き換えると楽かも！

```
void result() {
  background(204); // 背景色
  fill(0); // 塗り(文字)の色
  textAlign(CENTER, CENTER); // 文字列を上下左右中央揃え

  textFont(font36);
  text("正解数:", 40, 40, 240, 80);

  textFont(font18);
  text("マウスをクリックして再挑戦", 40, 140, 240, 40);

  if (mousePressed == true) {
    // マウスをクリックしたら
    scene = 0; // 開始画面へ移動
  }
}
```

結果画面でxキーで終了
quiz-1.7.zip

resultタブ

```
void result() {
  background(204); // 背景色
  fill(0); // 塗り(文字)の色
  textAlign(CENTER, CENTER); // 文字列を上下左右中央揃え

  textFont(font36);
  text("正解数:", 40, 40, 240, 80);

  textFont(font18);
  text("マウスをクリックして再挑戦", 40, 140, 240, 40);
  text("xキーを押して終了", 40, 220, 240, 40);

  if (mousePressed == true) {
    // マウスをクリックしたら
    scene = 0; // 開始画面へ移動
  } else if (keyPressed == true && key == 'x') {
    exit();
  }
}
```

正解数を表示
quiz-1.8.zip

メインのタブ

```
int scene; // シーン番号
int score; // 正解数

[以下略]
```

openingタブ

```
void opening() {
[中略]
    scene = 1; // プレイ画面へ移動
    score = 0; // 正解数を0に
  }
}
```

resultタブ

```
void result() {
[中略]

  textFont(font36);
  text("正解数:" + score, 40, 40, 240, 80);

[以下略]
```

クイズ番号の追加
quiz-1.9.zip

- 複数のクイズが出せるように「クイズ番号」という変数を用意し、プレイ画面に行く直前に値を設定する

メインのタブ

```
int scene; // シーン番号
int score; // 正解数
int quizNum; // クイズ番号

[以下略]
```

openingタブ

```
void opening() {
[中略]
    scene = 1; // プレイ画面へ移動
    score = 0; // 正解数を0に
    quizNum = 1; // クイズ番号を1に
  }
}
```

クイズ1(仮)を作る
quiz-1.10.zip

Q1タブ
※新規に作る。playingをコピーして書き換えると楽！

```
void Q1() {
  background(204); // 背景色
  fill(0); // 塗り(文字)の色
  textAlign(CENTER, CENTER); // 文字列を中央揃え

  textFont(font18);
  text("クイズ1", 40, 40, 240, 80);
  text("キーを押して結果画面へ", 40, 240, 240, 40);

  if (keyPressed == true) {
    // キーを押したら
    scene = 2; // 結果画面へ移動
  }
}
```

playingタブ
※中身を消して書き換える

```
void playing() {
  switch(quizNum) {
  case 1:
    Q1(); // クイズ1へ
    break;
  default:
    break;
  }
}
```

クイズ1の表示を作る
quiz-1.11.zip

Q1タブ

```
void Q1() {
  background(204); // 背景色
  fill(0); // 塗り(文字)の色
  textAlign(CENTER, CENTER); // 文字列を中央揃え

  textFont(font18);
  text("じゃんけんでグーに勝つのは？", 40, 40, 240, 80);

  text("1: グー", 80, 120, 160, 40);
  text("2: チョキ", 80, 180, 160, 40);
  text("3: パー", 80, 240, 160, 40);

  if (keyPressed == true) {
    // キーを押したら
    scene = 2; // 結果画面へ移動
  }
}
```

正解すると正解数が増えるようにする
quiz-1.12.zip

Q1タブ

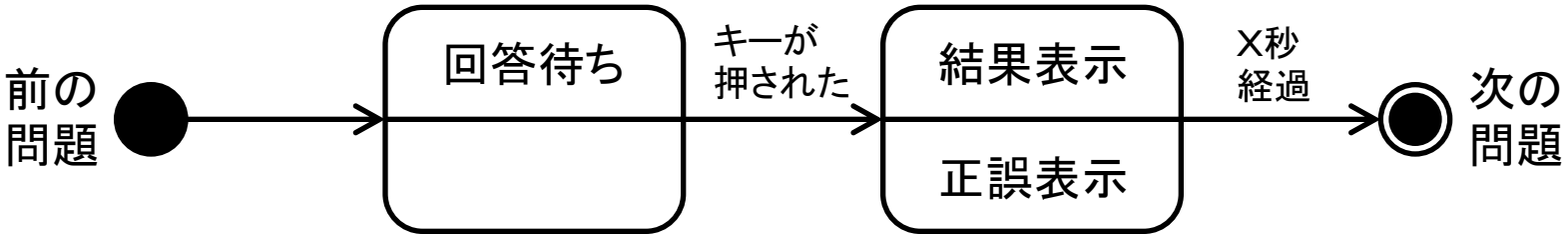
```
void Q1() {
  char answer = '3'; // 正解

  background(204); // 背景色
[中略]

  if (keyPressed == true) {
    // キーを押したら
    if (key == answer) {
      // 正解だったら
      score++; // 正解数を増やす
    }
    scene = 2; // 結果画面へ移動
  }
}
```

複雑な変化は「状態遷移図」で考える

- 「状態遷移図」とはゲームの状態の変化の図



- 必要な情報(変数)は？
  - 回答したか、まだか: 変数 isAnswered: true/false
  - 正解か、誤りか: 変数 isCorrect: true/false
  - 結果表示の経過時間
    - 起動時から現在までの経過時間はmillis() / 1000
    - 回答したときの経過時間 lapseAnswered をとっておけば、「現在までの経過時間 – 回答したときの経過時間」で求められる

状態を表す変数の追加
quiz-1.13.zip

- 変数を用意し、プレイ画面に行く直前に値を設定する

メインのタブ

```
int scene; // シーン番号
int score; // スコア
int quizNum; // クイズ番号
boolean isAnswered; // 回答したか
boolean isCorrect; // 正解か
float lapseAnswered; // 回答時経過時間

[以下略]
```

openingタブ

```
void opening() {

[中略]
    scene = 1; // プレイ画面へ移動
    score = 0; // 正解数を0に
    quizNum = 1; // クイズ番号を1に
    isAnswered = false; // 未回答に
    isCorrect = false; // 誤答に
  }
}
```

キーを押したときの動作を書く
quiz-1.14.zip

Q1タブ

```
void Q1() {
  char answer = '3'; // 正解

  background(204); // 背景色
[中略]

  if (isAnswered == false && keyPressed == true) {
    // 未回答時にキーを押したら
    isAnswered = true; // 回答した
    lapseAnswered = millis() / 1000; // 回答時経過時間を記録
    if (key == answer) {
      // 正解だったら
      score++; // 正解数を増やす
      isCorrect = true; // 正解した
    } else {
      isCorrect = false; // 間違えた
    }
  } else if (isAnswered == true) {
    // 回答したら
    scene = 2; // 結果画面へ移動
  }
}
```

回答して一定時間経過してから結果画面へ
quiz-1.15.zip

Q1タブ

```
void Q1() {
[中略]
  } else if (isAnswered == true) {
    // 回答したら
    if (millis() / 1000 - lapseAnswered > 2) {
      // 回答して2秒経過したら
      scene = 2; // 結果画面へ移動
    }
  }
}
```

正誤を表示
quiz-1.16.zip

Q1タブ

```
void Q1() {
[中略]
        scene = 2; // 結果画面へ移動
      } else {
        // 正誤を表示
        if (isCorrect == true) {
          // マルの表示
          ellipse(width / 2, height / 2, 0.8 * width, 0.8 * height);
        } else {
          // バツの表示
          line(0.1 * width, 0.1 * height, 0.9 * width, 0.9 * height);
          line(0.1 * width, 0.9 * height, 0.9 * width, 0.1 * height);
        }
      }
    }
}
```

正誤の表示を改善
quiz-1.17.zip

Q1タブ

```
void Q1() {
[中略]
      scene = 2; // 結果画面へ移動
    } else {
      // 正誤を表示
      noFill(); // 塗りつぶさない
      stroke(255, 0, 0); // 線を赤で
      strokeWeight(10); // 線の太さを10に

      if (isCorrect == true) {
[以下略]
```