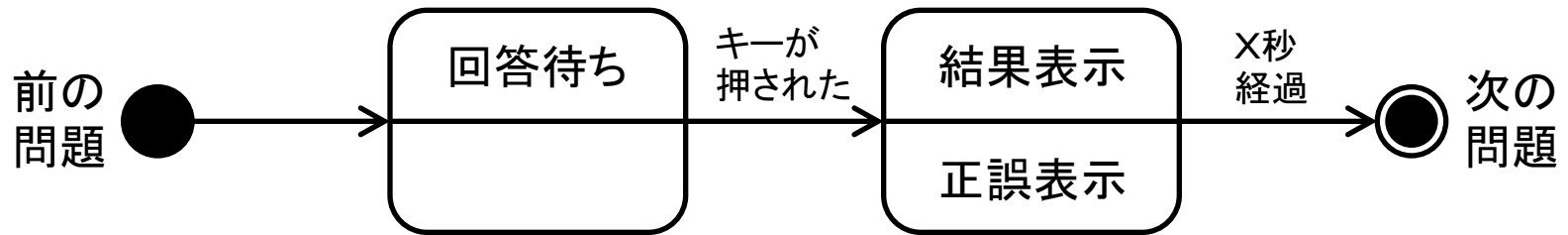


プログラミング演習I クイズをつくる：1

担当：小久保温

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]
```

クイズ2を作る

quiz-1.18.zip

- ▶ Q2タブを新しくつくり、Q1をコピーし書き換える

Q2タブ(新しく作る)

```
void Q2() {
    char answer = '2'; // 正解

    [中略]
    textAlign(CENTER, CENTER); // 文字列を中央揃え

    textFont(font18);
    text("浦島太郎がもらったのは?", 40, 40, 240, 80);

    text("1: 非時香菓", 80, 120, 160, 40);
    text("2: 玉手箱", 80, 180, 160, 40);
    text("3: 蓬萊の玉の枝", 80, 240, 160, 40);

    if (isAnswered == false && keyPressed == true) {
    [以下略]
```


クイズ2へ移動するようにする quiz-1.19.zip

playingタブ

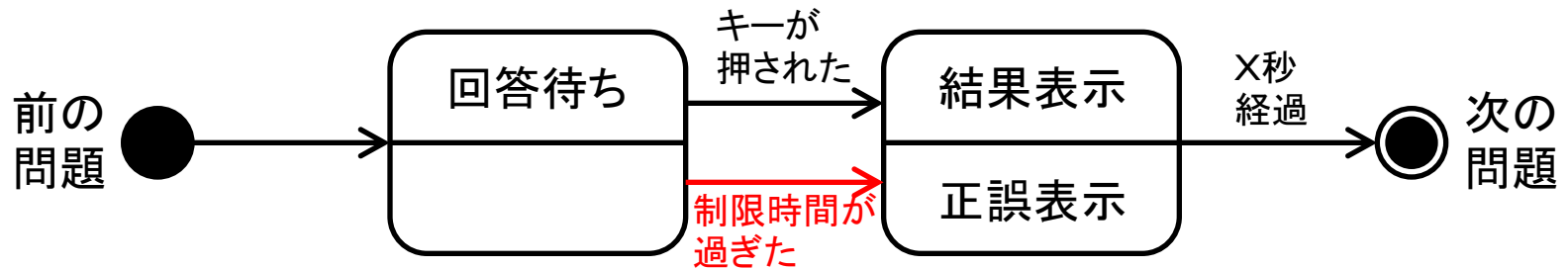
```
void playing() {  
    switch(quizNum) {  
        case 1:  
            Q1(); // クイズ1へ  
            break;  
        case 2:  
            Q2(); // クイズ2へ  
            break;  
        default:  
        default:  
            break;  
    }  
}
```

Q1タブ

```
void Q1() {  
    [中略]  
    if (millis() / 1000 - lapseAnswered > 2) {  
        // 回答して2秒経過したら  
        isAnswered = false; // 未回答に戻す  
        isCorrect = false; // 誤答に戻す  
        quizNum = 2; // クイズ2へ  
    } else {  
        [以下略]  
    }  
}
```

回答に制限時間をつけてみよう

▶ 「状態遷移図」を変更する



▶ 新たに必要な情報(変数)は？

▶ 問題を表示してからの経過時間

- ▶ 起動時から現在までの経過時間は`millis() / 1000`
- ▶ 問題を表示しはじめたときの経過時間 `lapseDisplayed` をとっておくと「現在までの経過時間 - 問題を表示しはじめたときの経過時間」で求められる

回答制限時間を表す変数の追加

quiz-1.20.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

```
[中略]
float lapseAnswered; // 回答時経過時間
float lapseDisplayed; // 問題を表示しはじめたときの経過時間
[以下略]
```

openingタブ

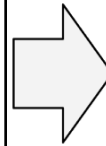
```
void opening() {
[中略]
    isCorrect = false; // 誤答に
    lapseDisplayed = millis() / 1000; // 問題表示開始時間を記録
}
}
```

Q1の整理

quiz-1.21.zip

```
void Q1() {
  [中略: 1]

  if (isAnswered == false && keyPressed == true) {
    // 未回答時にキーを押したら
    [中略: 2]
    if (key == answer) {
      // 正解だったら
      [中略: 3]
    } else {
      isCorrect = false; // 間違えた
    }
  } 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
      [中略: 4]
    } else {
      // 正誤を表示
      [中略: 5]
    }
  }
}
```



```
void Q1() {
  [中略: 1]

  if (isAnswered == false) {
    // 回答待ち
    if (keyPressed == true) {
      // 回答したら
      [中略: 2]
      if (key == answer) {
        // 正解だったら
        [中略: 3]
      } else {
        isCorrect = false; // 間違えた
      }
    }
  } else {
    // 結果表示
    if (millis() / 1000 - lapseAnswered > 2) {
      // 回答して2秒経過したら
      [中略: 4]
    } else {
      // 正誤を表示
      [中略: 5]
    }
  }
}
```

2スペース分
右にずらした

制限時間を入れる

quiz-1.22.zip

Q1タブ

```
void Q1() {
    char answer = '3'; // 正解
    [中略]
        isCorrect = false; // 間違えた
    }
}

// 制限時間
float timeLimit = 5 - (millis() / 1000 - lapseDisplayed);
if (timeLimit < 0) {
    // 回答制限時間を超えたら
    isAnswered = true; // 回答したことにする
    lapseAnswered = millis() / 1000; // 回答時経過時間を記録
    isCorrect = false; // 間違えたことにする
}
} else {
    // 結果表示
    if (millis() / 1000 - lapseAnswered > 2) {
    [以下略]
```

制限時間を表示

quiz-1.23.zip

Q1タブ

```
void Q1() {  
    char answer = '3'; // 正解  
    [中略]  
  
    // 制限時間  
    float timeLimit = 5 - (millis() / 1000 - lapseDisplayed);  
    textAlign(RIGHT, CENTER); // 文字列を右揃え  
    text("残り時間: " + timeLimit, 310, 15);  
  
    if (timeLimit < 0) {  
        // 回答制限時間を超えたら  
        isAnswered = true; // 回答したことにする  
    }  
    [以下略]
```

クイズ2へ移動する前に問題表示開始時間を記録

quiz-1.24.zip

Q1タブ

[中略]

```
// 結果表示
```

```
if (millis() / 1000 - lapseAnswered > 2) {
```

```
    // 回答して2秒経過したら
```

```
    isAnswered = false; // 未回答に戻す
```

```
    isCorrect = false; // 誤答に戻す
```

```
    lapseDisplayed = millis() / 1000; // 問題表示開始時間を記録
```

```
    quizNum = 2; // クイズ2へ
```

```
} else {
```

[以下略]

Q2も変更

quiz-1.25.zip

Q2タブ

```
void Q2() {  
[中略]
```

```
    if (isAnswered == false && keyPressed == true) {  
        // 未回答時にキーを押したら  
        [中略]  
    }
```



```
void Q2() {  
[中略]
```

```
    if (isAnswered == false) {  
        // 回答待ち  
        [中略]  
    }
```

Q1タブからコピーする

結果画面に移動するように変更

quiz-1.26.zip

Q2タブ

```
void Q2() {  
  [中略]  
  } else {  
    // 結果表示  
    if (millis() / 1000 - lapseAnswered > 2) {  
      // 回答して2秒経過したら  
      isAnswered = false; // 未回答に戻す  
      isCorrect = false; // 誤答に戻す  
      lapseDisplayed = millis() / 1000; // 問題表示開始時間を記録  
      scene = 2; // 結果画面へ  
    } else {  
      [以下略]
```

課題

- ▶ クイズ3を追加してみよう