

おみくじ

1. ディスプレイ・ウィンドウの表示

```
void setup() {  
  size(1024, 576);  
}  
  
void draw() {  
}
```

2. 背景画像の表示

```
PImage backgroundImage;  
  
void setup() {  
  size(1024, 576);  
  backgroundImage = loadImage("tellFortune.png");  
}  
  
void draw() {  
  image(backgroundImage, 0, 0);  
}
```

3. 背景画像を交互に切り替える

```
PImage backgroundImage;  
// 「占う」シーン  
PImage tellFortuneBackgroundImage;  
// 「結果」シーン  
PImage chukichiBackgroundImage;  
  
void setup() {  
  size(1024, 576);  
  // それぞれのシーンの画像を読み込む  
  tellFortuneBackgroundImage = loadImage("tellFortune.png");  
  chukichiBackgroundImage = loadImage("chukichi.png");  
  
  // 最初は「占う」シーン  
  backgroundImage = tellFortuneBackgroundImage;  
}  
  
void draw() {  
  image(backgroundImage, 0, 0);  
}  
  
void mousePressed() {  
  // 画像を切り替える  
  if (backgroundImage == tellFortuneBackgroundImage) {  
    backgroundImage = chukichiBackgroundImage;  
  } else {  
    backgroundImage = tellFortuneBackgroundImage;  
  }  
}
```

4. 結果をランダムに

```
PImage backgroundImage;  
PImage tellFortuneBackgroundImage;  
// 託宣の一覧  
String[] oracles = {  
    "daikichi.png",  
    "kichi.png",  
    "chukichi.png",  
    "shoukichi.png",  
    "suekichi.png",  
    "kyou.png",  
    "daikyou.png"  
};  
  
void setup() {  
    size(1024, 576);  
    tellFortuneBackgroundImage = loadImage("tellFortune.png");  
    backgroundImage = tellFortuneBackgroundImage;  
}  
  
void draw() {  
    image(backgroundImage, 0, 0);  
}  
  
void mousePressed() {  
    if (backgroundImage == tellFortuneBackgroundImage) {  
        int number = int(random(oracles.length));  
        backgroundImage = loadImage(oracles[number]);  
    } else {  
        backgroundImage = tellFortuneBackgroundImage;  
    }  
}
```

5. ステートパターン

5.1 シーンクラス テンプレ

```
class Scene {  
    Scene() {  
    }  
  
    void draw() {  
    }  
  
    void mousePressed() {  
    }  
}
```

5.2 「占う」シーン

```
class TellFortuneScene extends Scene {  
    PImage backgroundImage;  
  
    TellFortuneScene() {  
        backgroundImage = loadImage("tellFortune.png");  
    }  
  
    void draw() {  
        image(backgroundImage, 0, 0);  
    }  
  
    void mousePressed() {  
        // クリックされたら「結果」シーン  
        scene = new ResultScene();  
    }  
}
```

5.3 「結果」シーン

```
class ResultScene extends Scene {
    PImage backgroundImage;
    String[] oracles = {
        "daikichi.png",
        "kichi.png",
        "chukichi.png",
        "shoukichi.png",
        "suekichi.png",
        "kyou.png",
        "daikyou.png"
    };

    ResultScene() {
        // ランダムな結果を選ぶ
        int number = int(random(oracles.length));
        backgroundImage = loadImage(oracles[number]);
    }

    void draw() {
        image(backgroundImage, 0, 0);
    }

    void mousePressed() {
        // クリックされたら「占う」シーンへ
        scene = new TellFortuneScene();
    }
}
```

5.4 メイン・プログラム

```
Scene scene;

void setup() {
    size(1024, 576);
    // 最初のシーンは「占う」シーン
    scene = new TellFortuneScene();
}

void draw() {
    // 現在のシーンを描画
    scene.draw();
}

void mousePressed() {
    // 現在のシーンのマウスクリック処理
    scene.mousePressed();
}
```

6. ステートパターン 整理したもの

6.1 メイン・プログラム

変更なし ※以降、変更のない部分は掲載しない

6.2 シーンクラス 「占う」と「結果」シーンの共通部分を移植

```
class Scene {  
    PImage backgroundImage;  
  
    Scene() {  
    }  
  
    void draw() {  
        image(backgroundImage, 0, 0);  
    }  
  
    void mousePressed() {  
    }  
}
```


6.3 「占う」シーン 共通部分をシーンクラスへ

```
class TellFortuneScene extends Scene {  
    TellFortuneScene() {  
        backgroundImage = loadImage("tellFortune.png");  
    }  
  
    void draw() {  
        super.draw();  
    }  
  
    void mousePressed() {  
        scene = new ResultScene();  
    }  
}
```

6.4 「結果」シーン 共通部分をシーンクラスへ

```
class ResultScene extends Scene {
    String[] oracles = {
        "daikichi.png",
        "kichi.png",
        "chukichi.png",
        "shoukichi.png",
        "suekichi.png",
        "kyou.png",
        "daikyou.png"
    };

    ResultScene() {
        int number = int(random(oracles.length));
        backgroundImage = loadImage(oracles[number]);
    }

    void draw() {
        super.draw();
    }

    void mousePressed() {
        scene = new TellFortuneScene();
    }
}
```

7. ボタン画像を独立に

7.1 シーンクラス

```
class Scene {  
    PImage backgroundImage;  
  
    Scene() {  
    }  
  
    void draw() {  
        imageMode(CENTER);  
        image(backgroundImage, width / 2.0, height / 2.0);  
    }  
    void mousePressed() {  
    }  
}
```

7.2 「占う」シーン

```
class TellFortuneScene extends Scene {
    PImage tellFortuneButtonBackgroundImage;

    TellFortuneScene() {
        backgroundImage = loadImage("tellFortune.png");
        tellFortuneButtonBackgroundImage = loadImage("button.png");
    }

    void draw() {
        super.draw();
        image(tellFortuneButtonBackgroundImage, 658, 417);
    }

    void mousePressed() {
        scene = new ResultScene();
    }
}
```

7.3 「結果」シーン

```
class ResultScene extends Scene {
    String[] oracles = {
        "daikichi.png",
        "kichi.png",
        "chukichi.png",
        "shoukichi.png",
        "suekichi.png",
        "kyou.png",
        "daikyou.png"
    };
    PImage rewindButtonBackgroundImage;

    ResultScene() {
        int number = int(random(oracles.length));
        backgroundImage = loadImage(oracles[number]);
        rewindButtonBackgroundImage = loadImage("button.png");
    }

    void draw() {
        super.draw();
        image(rewindButtonBackgroundImage, 658, 417);
    }

    void mousePressed() {
        scene = new TellFortuneScene();
    }
}
```

8. ボタンに文字列を表示

8.1 シーンクラス

```
class Scene {  
  PImage backgroundImage;  
  
  Scene() {  
  }  
  
  void draw() {  
    imageMode(CENTER);  
    textAlign(CENTER, CENTER);  
    image(backgroundImage, width / 2.0, height / 2.0);  
  }  
  void mousePressed() {  
  }  
}
```

8.3 「占う」シーン

```
class TellFortuneScene extends Scene {
    PImage tellFortuneButtonBackgroundImage;
    String tellFortuneButtonText;
    PFont tellFortuneButtonFont;

    TellFortuneScene() {
        backgroundImage = loadImage("tellFortune.png");
        tellFortuneButtonBackgroundImage = loadImage("button.png");
        tellFortuneButtonText = "占う!";
        tellFortuneButtonFont = createFont("MS Gothic", 48);
    }

    void draw() {
        super.draw();
        image(tellFortuneButtonBackgroundImage, 658, 417);
        textFont(tellFortuneButtonFont);
        fill(255);
        text(tellFortuneButtonText, 658, 417);
    }

    void mousePressed() {
        scene = new ResultScene();
    }
}
```

8.4 「結果」シーン

```
class ResultScene extends Scene {
    String[] oracles = {
        "daikichi.png",
        "kichi.png",
        "chukichi.png",
        "shoukichi.png",
        "suekichi.png",
        "kyou.png",
        "daikyou.png"
    };
    PImage rewindButtonBackgroundImage;
    String rewindButtonText;
    PFont rewindButtonFont;

    ResultScene() {
        int number = int(random(oracles.length));
        backgroundImage = loadImage(oracles[number]);
        rewindButtonBackgroundImage = loadImage("button.png");
        rewindButtonText = "◀戻る";
        rewindButtonFont = createFont("MS Gothic", 48);
    }

    void draw() {
        super.draw();
        image(rewindButtonBackgroundImage, 658, 417);
        textFont(rewindButtonFont);
        fill(255);
        text(rewindButtonText, 658, 417);
    }

    void mousePressed() {
        scene = new TellFortuneScene();
    }
}
```



```
}  
}
```

9. ボタンをクリックできるように

9.1 「占う」シーン

```
class TellFortuneScene extends Scene {
    PImage tellFortuneButtonBackgroundImage;
    String tellFortuneButtonText;
    PFont tellFortuneButtonFont;

    TellFortuneScene() {
        backgroundImage = loadImage("tellFortune.png");
        tellFortuneButtonBackgroundImage = loadImage("button.png");
        tellFortuneButtonText = "占う!";
        tellFortuneButtonFont = createFont("MS Gothic", 48);
    }

    void draw() {
        super.draw();
        image(tellFortuneButtonBackgroundImage, 658, 417);
        textFont(tellFortuneButtonFont);
        fill(255);
        text(tellFortuneButtonText, 658, 417);
    }

    void mousePressed() {
        float buttonX = 457;
        float buttonY = 376;
        float buttonWidth = 501;
        float buttonHeight = 82;

        if (buttonX < mouseX && mouseX < buttonX + buttonWidth
            && buttonY < mouseY && mouseY < buttonY + buttonHeight) {
            scene = new ResultScene();
        }
    }
}
```

}
}

9,2 「結果」シーン

```
class ResultScene extends Scene {
    String[] oracles = {
        "daikichi.png",
        "kichi.png",
        "chukichi.png",
        "shoukichi.png",
        "suekichi.png",
        "kyou.png",
        "daikyou.png"
    };
    PImage rewindButtonBackgroundImage;
    String rewindButtonText;
    PFont rewindButtonFont;

    ResultScene() {
        int number = int(random(oracles.length));
        backgroundImage = loadImage(oracles[number]);
        rewindButtonBackgroundImage = loadImage("button.png");
        rewindButtonText = "◀戻る";
        rewindButtonFont = createFont("MS Gothic", 48);
    }

    void draw() {
        super.draw();
        image(rewindButtonBackgroundImage, 658, 417);
        textFont(rewindButtonFont);
        fill(255);
        text(rewindButtonText, 658, 417);
    }

    void mousePressed() {
        float buttonX = 457;
```

```
float buttonY = 376;
float buttonWidth = 501;
float buttonHeight = 82;

if (buttonX < mouseX && mouseX < buttonX + buttonWidth
    && buttonY < mouseY && mouseY < buttonY + buttonHeight) {
    scene = new TellFortuneScene();
}
}
}
```

10. ボタンクラスの導入

10.1 ボタンクラス

```
class Button {
    PImage backgroundImage;
    String text;
    PFont font;
    color textColor;
    float x;
    float y;

    Button() {
    }

    void draw() {
        image(backgroundImage, x, y);
        textFont(font);
        fill(textColor);
        text(text, x, y);
    }

    void mousePressed() {
    }

    boolean isTouched() {
        boolean result = false;
        if (x - backgroundImage.width / 2.0 < mouseX
            && mouseX < x + backgroundImage.width / 2.0
            && y - backgroundImage.height / 2.0 < mouseY
            && mouseY < y + backgroundImage.height / 2.0) {
            result = true;
        }
        return result;
    }
}
```

```
}  
}
```

10.2 「占う！」ボタン

```
class TellFortuneButton extends Button {  
    TellFortuneButton() {  
        backgroundImage = loadImage("button.png");  
        text = "占う！";  
        font = createFont("MS Gothic", 48);  
        textColor = color(255);  
        x = 658;  
        y = 417;  
    }  
  
    void mousePressed() {  
        if (isTouched() == true) {  
            scene = new ResultScene();  
        }  
    }  
}
```

10.3 「戻る」ボタン

```
class RewindButton extends Button {
  RewindButton() {
    backgroundImage = loadImage("button.png");
    text = "◀戻る";
    font = createFont("MS Gothic", 48);
    textColor = color(255);
    x = 658;
    y = 417;
  }

  void mousePressed() {
    if (isTouched() == true) {
      scene = new TellFortuneScene();
    }
  }
}
```


10.4 「占う」シーン

```
class TellFortuneScene extends Scene {
    TellFortuneButton tellFortuneButton;

    TellFortuneScene() {
        backgroundImage = loadImage("tellFortune.png");
        tellFortuneButton = new TellFortuneButton();
    }

    void draw() {
        super.draw();
        tellFortuneButton.draw();
    }

    void mousePressed() {
        tellFortuneButton.mousePressed();
    }
}
```

10.5 「結果」 シーン

```
class ResultScene extends Scene {
    String[] oracles = {
        "daikichi.png",
        "kichi.png",
        "chukichi.png",
        "shoukichi.png",
        "suekichi.png",
        "kyou.png",
        "daikyou.png"
    };
    RewindButton rewindButton;

    ResultScene() {
        int number = int(random(oracles.length));
        backgroundImage = loadImage(oracles[number]);
        rewindButton = new RewindButton();
    }

    void draw() {
        super.draw();
        rewindButton.draw();
    }

    void mousePressed() {
        rewindButton.mousePressed();
    }
}
```

11. 占断クラスの導入

11.1 占断クラス

```
class Oracle {  
    String name;  
    String description;  
    PImage image;  
  
    Oracle() {  
    }  
  
    Oracle(String name, String description, PImage image) {  
        this.name = name;  
        this.description = description;  
        this.image = image;  
    }  
}
```

11.2 メイン・プログラム

```
Scene scene;
ArrayList<Oracle> oracles;

void setup() {
    size(1024, 576);
    scene = new TellFortuneScene();

    oracles = new ArrayList<Oracle>();
    oracles.add(new Oracle("大吉", "絶好調です!",
        loadImage("daikichi.png")));
    oracles.add(new Oracle("吉", "いいことがあるみたいです!",
        loadImage("kichi.png")));
    oracles.add(new Oracle("中吉", "なかなかいい感じです!",
        loadImage("chukichi.png")));
    oracles.add(new Oracle("小吉", "ちょっといいことあるかもね!",
        loadImage("shoukichi.png")));
    oracles.add(new Oracle("末吉", "後でいいことがあるみたいです!",
        loadImage("suekichi.png")));
    oracles.add(new Oracle("凶", "そういうこともありますよね!",
        loadImage("kyou.png")));
    oracles.add(new Oracle("大凶", "慎重に行動しましょう!",
        loadImage("daikyou.png")));
}

void draw() {
    scene.draw();
}

void mousePressed() {
    scene.mousePressed();
}
```

11.3 「結果」シーン

```
class ResultScene extends Scene {
    RewindButton rewindButton;
    Oracle oracle;
    PFont nameFont;
    PFont descriptionFont;

    ResultScene() {
        backgroundImage = loadImage("result.png");
        int number = int(random(oracles.size()));
        oracle = oracles.get(number);
        rewindButton = new RewindButton();
        nameFont = createFont("MS Gothic", 60);
        descriptionFont = createFont("MS Gothic", 28);
    }

    void draw() {
        super.draw();

        textAlign(LEFT, TOP);
        fill(0);
        textFont(nameFont);
        text("あなたの運勢は" + oracle.name + "です!", 457, 106, 440, 153);

        textFont(descriptionFont);
        text(oracle.description, 446, 276);

        image(oracle.image, 213, 289);

        rewindButton.draw();
    }

    void mousePressed() {
```

```
    rewindButton.mousePressed();  
}  
}
```

11.4 ボタンクラス

```
class Button {
    PImage backgroundImage;
    String text;
    PFont font;
    color textColor;
    float x;
    float y;

    Button() {
    }

    void draw() {
        image(backgroundImage, x, y);
        textAlign(CENTER, CENTER);
        textFont(font);
        fill(textColor);
        text(text, x, y);
    }

    void mousePressed() {
    }

    boolean isTouched() {
        boolean result = false;
        if (x - backgroundImage.width / 2.0 < mouseX
            && mouseX < x + backgroundImage.width / 2.0
            && y - backgroundImage.height / 2.0 < mouseY
            && mouseY < y + backgroundImage.height / 2.0) {
            result = true;
        }
        return result;
    }
}
```

}

12. 占断データを CSV ファイルに

12.1 メイン・クラス

```
Scene scene;
ArrayList<Oracle> oracles;

void setup() {
    size(1024, 576);
    scene = new TellFortuneScene();

    loadOracles();
}

void draw() {
    scene.draw();
}

void mousePressed() {
    scene.mousePressed();
}

void loadOracles() {
    oracles = new ArrayList<Oracle>();

    Table table;
    table = loadTable("oracles.csv", "header");
    for (TableRow tableRow : table.rows()) {
        String name = tableRow.getString("名称");
        String description = tableRow.getString("説明");
        String imageFile = tableRow.getString("画像ファイル");
        PImage image = loadImage(imageFile);
        Oracle oracle = new Oracle(name, description, image);
        oracles.add(oracle);
    }
}
```

```
}  
}
```

13. コンストラクタを改良

13.1 占断クラス

```
class Oracle {
    String name;
    String description;
    PImage image;

    Oracle() {
    }

    Oracle(String name, String description, PImage image) {
        this.name = name;
        this.description = description;
        this.image = image;
    }

    Oracle(TableRow tableRow) {
        this.name = tableRow.getString("名称");
        this.description = tableRow.getString("説明");
        String imageFile = tableRow.getString("画像ファイル");
        this.image = loadImage(imageFile);
    }
}
```

13.2 メイン・クラス

```
Scene scene;
ArrayList<Oracle> oracles;

void setup() {
    size(1024, 576);
    scene = new TellFortuneScene();

    loadOracles();
}

void draw() {
    scene.draw();
}

void mousePressed() {
    scene.mousePressed();
}

void loadOracles() {
    oracles = new ArrayList<Oracle>();

    Table table;
    table = loadTable("oracles.csv", "header");
    for (TableRow tableRow : table.rows()) {
        Oracle oracle = new Oracle(tableRow);
        oracles.add(oracle);
    }
}
```