

■課題 1 クリスマスカードを作成しましょう。実行例に示すようにキャンバスの中央にメッセージ「Merry Xmas」の文字列を描画します。さらに、キャンバス上をマウスでクリックすると、クリック座標を中心にして雪の結晶画像が表示されます。

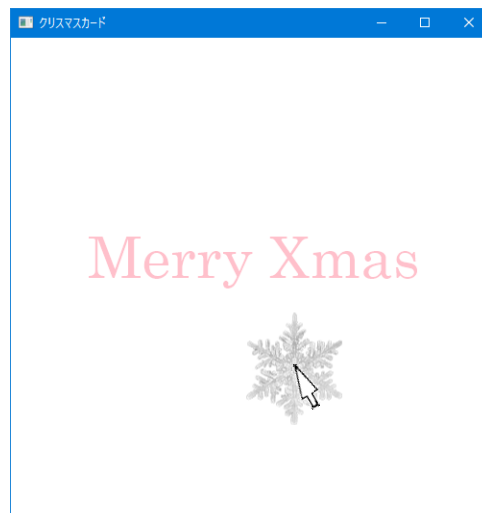
その他の設定は次のとおりです。参考にしましょう。

キャンバスサイズ	→ 横 500 ピクセル 縦 500 ピクセル
メッセージのフォント	→ "Century" の 60 ポイント [new Font("Century", 60);]
メッセージの座標	→ (80, 250)
メッセージの色	→ ピンク [Color.PINK]
ウィンドウのタイトル	→ クリスマスカード

ヒント：Image クラスのオブジェクト（画像）のサイズは、getWidth()と getHeight()で取得できます。これらを用いてクリック位置に画像の中心が描画されるようにしましょう。

※おおよそ実行例のような画面になれば OK です

〔実行例〕



■課題 2 グリーティングカードアプリを作成しましょう。実行例のように GUI 部品を配置します。ラジオボタンで描画色を選択し、キャンバス上でマウスをドラッグすると指定した色で描画されます。下にあるクリアボタンを押すとキャンバスがクリアされます。また保存ボタンを押すと現在のキャンバス画像が画像ファイルに保存されます。

その他の設定は次のとおりです。参考にしましょう。

HBox レイアウト周りの空白エリア → 10 ピクセル [setPadding(new Insets(10));]
HBox-GUI 部品間の空白エリア → 10 ピクセル [setSpacing(10);]
シーンの色 → オレンジ [Color.ORANGE]
※レイアウト VBox の背景色を透明にします
レイアウト VBox の setBackground(null); を実行
ウィンドウのタイトル → グリーティングカード

ヒント 1: マウスのドラッグイベントが発生したらそのマウス位置に直径 10 ピクセルの円を描画します。毎回描画するとき、キャンバスをクリアしなければ前回描画した円は消えずに残ります。ドラッグイベントが発生する度に円を描画して軌跡を残していきます。

ヒント 2: クリアボタンが押されたときのみ、キャンバスをクリアします。

ヒント 3: 保存ボタンが押されたら、次のコードを実行してファイルに画像を保存します。

〔ファイル CanvasImage.png に画像を保存するコード〕

```
WritableImage bi = new WritableImage((int)cv.getWidth(), (int)cv.getHeight());
cv.snapshot(null, bi);
File file = new File("CanvasImage.png");
try{
    ImageIO.write(SwingFXUtils.fromFXImage(bi, null), "png", file);
}catch(Exception ee){}
```

※変数 `cv` はクラス `Canvas` 型の変数です。このキャンバスが画像ファイルに保存されます。

※画像ファイルへの保存を行う各クラスを利用するには次の `import` 文を加えましょう。

```
import javafx.embed.swing.*;
import javax.imageio.*;
import java.io.*;
```

※画像は HP よりダウンロードしてソースファイルと同じフォルダに保存しておきましょう

※おおよそ実行例のような画面になれば OK です

〔実行例〕

