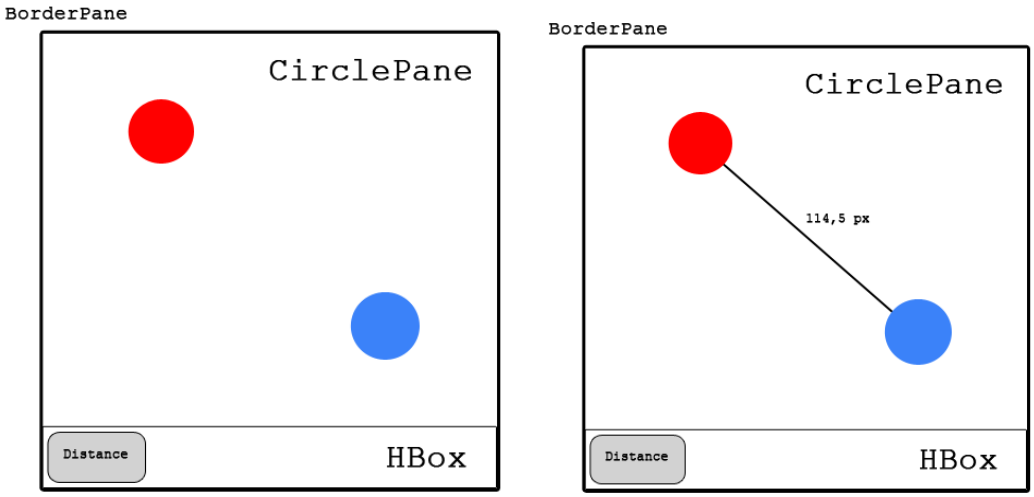
Assignment 9
Hand-in date: Monday 09-11-2015 15:00h

We would like to create the following application that displays two random circles and shows the distance between the two center points when clicking on a button:



Notice how the GUI consists of a **BorderPane** that contains an custom made **CirclePane** (with circles) and an **HBox** (with a button).

10.1 To get started the first thing you should do is to create the JavaFX Main Class and implement the **start** method.

10.2 The **CirclePane** is a custom class because we want to perform custom actions on the two circles. So first of all we need to write the **CirclePane** class. This is done by creating the inner class **CirclePane** and let it extend the **Pane** class. Inside the class we need two fields: One for each **Circle**. (Note that you must use the **javafx.scene.shape.Circle** class)

In the constructor of the class we set the properties of the circles. The first circle has the following properties:

- The center point is random (random double for x and y coordinates).
- The radius is 15px.
- The color is RED.

The second circle has the exact same properties; only the color is BLUE Instead of RED. In the constructor we also need to add the two circles to the pane. This is done by calling **getChildren().add(circle)** for each circle. Lastly you must add a method called **showDistance()** to the **CirclePane** class. For now, we will leave the method empty. In the JavaFX main class you must create a **CirclePane** field in order to reference it later on.

Assignment 9
Hand-in date: Monday 09-11-2015 15:00h

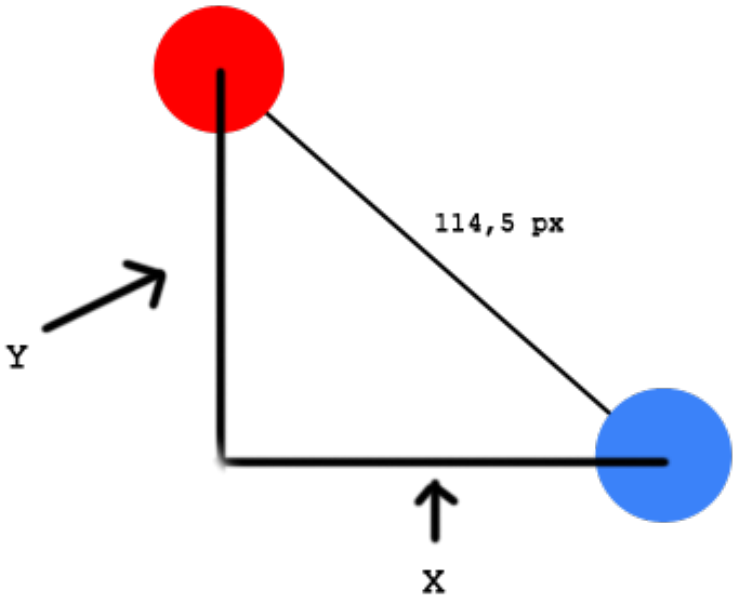
10.3 Create the **HBox** containing the ”Distance” button. To do so you must create a new **HBox** in the **start()** method of the JavaFX main class. Then create a button with the desired text and add it to the **HBox**.

10.4 Now that we have created what we want to show, we can create the **BorderPane** that contains the **CirclePane** and **HBox**. Do so, by creating a new **BorderPane** and assigning the CirclePane to the center of the BorderPane and the **HBox** to the bottom of the **BorderPane**. (Hint: use the **setCenter()** and **setBottom()** methods) Add the **BorderPane** to the scene and show it. When running the program you should see a GUI corresponding to the leftmost image at the top of this assignment.

10.5 At this point the button does not do anything at all. We need to register an event handler to it. Create an event handler called **DistanceHandler** that calls the **showDistance** method of the **CirclePane** class. Register the handler to the button.

10.6 Now we need to write the code for the **showDistance** method in the **CirclePane** class. Remember that this method needs to calculate the distance between the center points of the two circles and display it with a **Line** and the text showing the distance. To do so you need to display a **Line** going from the center of the first circle to the center of the second circle and then calculate the length of this Line. The distance can be calculated as can be seen on the illustration below. You should either show the distance text next to the Line or next to the “Distance” Button.

Assignment 9
Hand-in date: Monday 09-11-2015 15:00h




```
Y = Math.abs(y1-y2)

X = Math.abs(x1-x2)

Distance = Math.sqrt(Math.pow(Y,2)+ Math.pow(X,2))
```

10.7 (OPTIONAL) Make it possible to drag one of the circles around using the mouse.
(Hint: set the center x and y points of the circle to the x and y coordinates of the mouse when it is being dragged).

10.8 (OPTIONAL) If the Distance Line is shown while the user drags around a Circle, make the Distance Line and the Distance text change accordingly.
(Hint: Use Property Binding)