

Apuntes T5

1. Formas. Clases y métodos específicos

Circle:

```
Circle circle1 = new Circle(/* centroX, centroY, radio */);

circle1.setCenterX(100);           // Establecer el centro en eje x
circle1.setCenterY(100);           // Establecer el centro en eje y
circle1.setRadius(50);            // Establecer radio
circle1.setStroke(Color.CYAN);     // Establecer color del borde
circle1.setFill(Color.LIGHTGRAY); // Establecer relleno del círculo

circle2.centerXProperty().bind(pane.widthProperty().divide(2));    // |
circle2.centerYProperty().bind(pane.heightProperty().divide(2)); // |__> Modificar el centro del círculo de manera dinámica en base al layout
```

Rectangle:

```
Rectangle rectangle1 = new Rectangle(/* centroX, centroY, anchura, altura */);
rectangle1.setX(50.0);           // Establecer el centro en eje x
rectangle1.setY(20.0);           // Establecer el centro en eje y
rectangle1.setWidth(120.0);      // Establece la anchura del rectángulo
rectangle1.setHeight(60.0);       // Establece la altura del rectángulo
rectangle1.setStyle("-fx-color: red");

rectangle3.setArcHeight(30); // |
rectangle3.setArcWidth(30); // |__> Métodos para redondear las esquinas

rectangle.setRotate(i * 180 / N); // Rota x grados con respecto a su centro
```

Line:

```
Line lineaRoja = new Line( x: 10, y1: 10, y2: 500, x2: 500); // inicioX, finalX, inicioY, finalY
lineaRoja.endXProperty().bind(pane.widthProperty().subtract(10)); // |
lineaRoja.endYProperty().bind(pane.heightProperty().subtract(10)); // |__> Métodos para establecer el final de la línea de forma dinámica
// Método para hacer que las líneas sean discontinuas
lineaRoja.getStrokeDashArray().addAll( ...es: 50d, lineaRoja.getStrokeWidth() + 20d); // trazo 1, hueco 1
```

2. Transformaciones

```
// Se crea un rotate con las indicaciones de cuanto y en que punto rotar
Rotate rotate = new Rotate(); // angulo, pivotX, pivotY
rotate.setAngle(valor - posAnterior);
rotate.setPivotX(circle1.getCenterX());
rotate.setPivotY(circle1.getCenterY());

// Se establece a la figura a la que queremos rotar y se guarda la posicion para que siga por donde esta
circle2.getTransforms().add(rotate);
posAnterior = valor;

// Se establecen los valores finales de la animacion
KeyValue kvRotate = new KeyValue(rotate.angleProperty(), 360);

KeyValue kvSlider = new KeyValue(slider.valueProperty(), 360);

// Se define la animacion con su respectiva duracion y valores finales
KeyFrame kf = new KeyFrame(Duration.seconds(3), kvRotate, kvSlider);
Timeline tl = new Timeline(kf);
tl.play(); // Se reproduce
tl.setCycleCount(10); // Se dice el numero de ciclos que se quieren hacer

// Se establecen unos valores para que se establezcan hasta que acabe la animacion
tl.setOnFinished( ActionEvent e1 -> {
    slider.setDisable(false);
    slider.setValue(360);
});
```

3. Métodos para eventos

SetOnMousePressed():

Se ejecuta cuando se presiona un botón del ratón sobre un control sin importar si luego se suelta o se arrastra.

```
// Se guarda donde se hace el click para guardar el pivate de rotacion
pane.setOnMousePressed( MouseEvent e -> {
    for (Rectangle r : rectangulos) {
        double[] datos = (double[]) r.getUserData();
        if (datos[3] == 1) {
            datos[0] = e.getSceneX();
            datos[1] = e.getSceneY();
        }
    }
});
```

SetOnMouseClicked():

Detecta cuando se presiona y suelta el botón del ratón sobre el mismo control, equivalente a un “clic”.

```
// Se deseleccionan los rectangulos si se pulsa en el pane
pane.setOnMouseClicked(
    MouseEvent e -> {
        for (Rectangle r : rectangulos) {
            r.setStroke(null);
            ((double[]) r.getUserData())[3] = 0;
        }
    }
);
```

SetOnMouseDragged():

Se ejecuta cuando el ratón se mueve mientras el botón está presionado sobre un control, se suele usar para rotar objetos o mover elementos, por ejemplo.

```
// Rotacion de los rectangulos en base al pivote de los rectangulos seleccionados
pane.setOnMouseDragged( MouseEvent e2 -> {
    for (Rectangle r : rectangulos) {
        double[] datos = (double[]) r.getUserData();
        if (datos[3] == 0)
            continue;

        double pivotX = r.getLayoutX() + r.getX() + r.getWidth() / 2.0;
        double pivotY = r.getLayoutY() + r.getY() + r.getHeight() / 2.0;

        double anguloPressed = Math.toDegrees(
            Math.atan2(datos[1] - pivotY, datos[0] - pivotX)
        );
        double anguloDragged = Math.toDegrees(
            Math.atan2(e2.getSceneY() - pivotY, e2.getSceneX() - pivotX)
        );

        double a = anguloDragged - anguloPressed + datos[2];
        r.setRotate(a);
    }
});
```

SetOnMouseReleased():

Detecta cuando el usuario suelta el botón del ratón, útil para terminar de desplazar un objeto y dejarlo fijo.

```
// Se guarda la rotacion hecha
pane.setOnMouseReleased( MouseEvent e3 -> {
    for (Rectangle r : rectangulos) {
        double[] datos = (double[]) r.getUserData();
        if (datos[3] == 0)
            continue;

        datos[2] = r.getRotate();
    }
});
```

GetSceneX():

Devuelve la coordenada X del ratón con respecto a la escena.

```
// Rotacion de los rectangulos en base al pivote de los rectangulos seleccionados
pane.setOnMouseDragged( MouseEvent e2 -> {
    for (Rectangle r : rectangulos) {
        double[] datos = (double[]) r.getUserData();
        if (datos[3] == 0)
            continue;

        double pivotX = r.getLayoutX() + r.getX() + r.getWidth() / 2.0;
        double pivotY = r.getLayoutY() + r.getY() + r.getHeight() / 2.0;

        double anguloPressed = Math.toDegrees(
            Math.atan2(datos[1] - pivotY, datos[0] - pivotX)
        );
        double anguloDragged = Math.toDegrees(
            Math.atan2(e2.getSceneY() - pivotY, e2.getSceneX() - pivotX)
        );

        double a = anguloDragged - anguloPressed + datos[2];
        r.setRotate(a);
    }
});
```

GetCode():

Devuelve la tecla del teclado pulsada

```
scene.setOnKeyPressed(e -> {
    if (e.getCode() == KeyCode.SPACE) {
        System.out.println("Tecla ESPACIO pulsada");
    }
});
```

Ejemplo de ChatGPT ya que no he encontrado ninguna practica con este evento