جامعة طرابلس
كلية تقنية المعلومات

# البرمجة المرئية
# Visual Programming

ITSE423

إعداد: حسن علي حسن إبراهيم
المحاضر: فاطمة علي بن الأشهر

**المحاضرة السادسة**

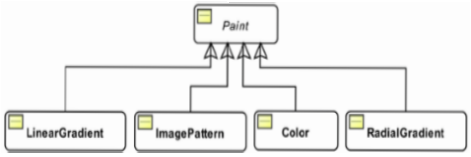
## مواضيع المحاضرة

- **اللعب بالألوان**
- **تطبيق التأثيرات**

- كيف يتم تمثيل الألوان في JavaFX
- ما هي أنماط الألوان المختلفة
- كيفية استخدام نمط الصورة
- كيفية استخدام التدرج اللوني الخطي
- كيفية استخدام التدرج اللوني الشعاعي

## Colors الألوان

- لتطبيق الألوان على أحد التطبيقات، توفر JavaFX فئات مختلفة في الحزمة javafx.scene.paint. تحتوي هذه الحزمة على فئة تسمى الرسام paint وهي الفئة الأساسية لجميع الفئات المستخدمة لتطبيق الألوان.
- باستخدام هذه الفئات ، يمكنك تطبيق الألوان في الأنماط التالية:
- - لون موحد.- نمط الصورة.- التدرج اللون.
  - Uniform color.
  - Image pattern.
  - Color gradient.






## Colors الألوان

- لون موحد Uniform: في هذا النمط ، يتم تطبيق اللون بشكل موحد في جميع أنحاء العقدة.
- نمط الصورة Image Pattern: يتيح لك هذا ملء منطقة العقدة بنمط صورة.
- التدرج اللوني Gradient: في هذا النمط ، يختلف اللون المطبق على العقدة من نقطة إلى أخرى.
- لديها نوعان من التدرجات وهما التدرج الخطي Linear Gradient والتدرج الشعاعي Radial Gradient.

## الألوان Colors

- كل فئات العقد node classes التي يمكنك تطبيق اللون عليها مثل الشكل Text والنص Shape (بما في ذلك المشهد Scene) لها طرق باسم setFill() و setStroke().
- سيساعد ذلك في ضبط قيم الألوان للعقد وحدودها على التوالي.
- تقبل هذه الطرق كائنًا من النوع Paint.

## استخدام فئة الألوان Colors

- تمثل فئة اللون لونًا solid من مساحة ألوان RGB.
- كل لون له قيمة محددة بين 0.0 إلى 1.0 أو من 0 إلى 255.
- قيمة اللون 0.0 أو 0 تعني أن اللون شفاف تمامًا، وقيمة اللون 1.0 أو 255 تشير إلى لون معتم تمامًا.
- بشكل افتراضي، يتم تعيين قيمة اللون على 1.0.
- يمكن الحصول على مثيل لفئة اللون بثلاث طرق:
- باستخدام المنشئ constructor.
- باستخدام إحدى طرق المصنع one of the factory methods.
- استخدام أحد ثوابت اللون المعلنة في فئة اللون one of the color constants declared in the Color class

## استخدام المنشئ constructor

- تحتوي فئة Color على مُنشئ واحد فقط يسمح لك بتحديد RGB والعتامة في النطاق بين 0.0 و 1.0:
- public Color(double red, double green, double blue, double opacity)
- يُنشئ الجزء البرمجي التالي لونًا أزرق معتمًا تمامًا:
- Color blue = new Color(0.0, 0.0, 1.0, 1.0);
- يمكنك استخدام الطرق الثابتة التالية في فئة Color لإنشاء كائنات ملونة.
- يجب أن تكون القيم بين 0.0 و 1.0 أو قيم صحيحة بين 0 و 255:



## التعامل مع الألوان

- Color color(double red, double green, double blue, double opacity)
- Color hsb(double hue, double saturation, double brightness)
- Color hsb(double hue, double saturation, double brightness, double opacity) //التدرج التشبع السطوع
- Color rgb(int red, int green, int blue)
- Color rgb(int red, int green, int blue, double opacity)

- https://htmlcolorcodes.com

## استخدام بعض الدوال

- * باستخدام إحدى طرق المصنع
- تتيح لك أساليب المصنع valueOf () وweb () إنشاء كائنات ملونة من سلاسل بتنسيقات قيم ألوان الويب.يُنشئ المقتطف التالي من التعليمات البرمجية كائنات زرقاء اللون باستخدام تنسيقات سلسلة مختلفة:

```
Color blue = Color.valueOf("blue");
Color blue = Color.web("blue");
Color blue = Color.web("#0000FF");
Color blue = Color.web("0X0000FF");
Color blue = Color.web("rgb(0, 0, 255)");
Color blue = Color.web("rgba(0, 0, 255, 0.5)"); // 50% transparent blue
```



## استخدام أحد ثوابت اللون المعلنة في فئة اللون

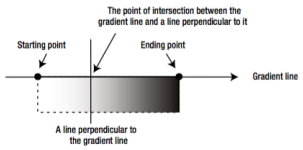
- **تحدد فئة اللون حوالي 140 من ثوابت اللون ، على سبيل المثال ، RED، BLUE، من بين أمور أخرى. الألوان المحددة بواسطة هذه الثوابت غير شفافة تمامًا.لون أزرق**

- Color.BLUE

## استخدام فئة LinearGradient

- في JavaFX، يمثل مثيل فئة LinearGradient تدرج لوني خطي. يحتوي الفصل على المنشئين التاليين. تختلف أنواع حججهم الأخيرة:

- LinearGradient(double startX, double startY, double endX, double endY, boolean proportional, CycleMethod cycleMethod, List<Stop> stops)
- LinearGradient(double startX, double startY, double endX, double endY, boolean proportional, CycleMethod cycleMethod, Stop... stops)




Hassan Ebrahem

11

## خصائص فئة LinearGradient

- خصائص التدرج الخطي ملاحظة: يجب أن تكون إحداثيات xو y لنقطتي البداية والنهاية بين 0.0 و 1.0 (مزدوجة).

| Property | Data Type / Description |
|---|---|
| startX | `Double` Set X coordinate of the gradient axis start point. |
| startY | `Double` Set Y coordinate of the gradient axis start point. |
| endX | `Double` Set X coordinate of the gradient axis end point. |
| endY | `Double` Set Y coordinate of the gradient axis end point. |
| proportional | `Boolean` Set whether the coordinates are proportional to the shape. true will use unit square coordinates, otherwise use screen coordinate system. |
| cycleMethod | `CycleMethod` Set cycle method applied to the gradient. |
| stops | `List<Stop>` Set stop list for gradient's color specification. |

Hassan Ebrahem

12

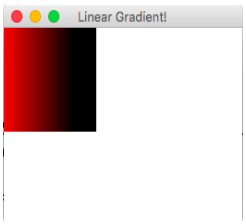
# خصائص فئة LinearGradient

- قيمته هي أحد ثوابت التعداد المحددة في CycleMethod enum:1.CycleMethod.NO_CYCLE.2.CycleMethod.REFLECT.3.CycleMethod.REPEAT. تحدد وسيطة التوقف نقاط توقف اللون على طول خط التدرج اللوني. يتم تمثيل نقطة توقف اللون بمثيل من فئة Stop، والذي يحتوي على مُنشئ واحد فقط: توقف (إزاحة مزدوجة ، لون اللون)

```
Stop[] stops = new Stop[]{new Stop(0, Color.RED),
                          new Stop(0.25, Color.GREEN),
                          new Stop(0.50, Color.BLUE),
                          new Stop(0.75, Color.ORANGE),
                          new Stop(1, Color.YELLOW)};
LinearGradient lg = new LinearGradient(0, 0, 1, 0, true, NO_CYCLE, stops);
Rectangle r = new Rectangle(200, 100);
r.setFill(lg);
```

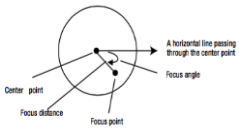
# مثال

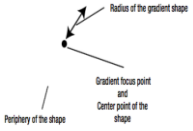
```
public void start(Stage primaryStage) {

    VBox box = new VBox();

    Stop[] stops = new Stop[] { new Stop(0, Color.RED),
        new Stop(.75, Color.BLACK)};

    LinearGradient lg1 = new LinearGradient(
            0,
            0,
            1,
            0,
            true,
            CycleMethod.NO_CYCLE,
            stops);

    Rectangle r1 = new Rectangle(0, 0, 100, 100);
    r1.setFill(lg1);

    box.getChildren().add(r1);
```

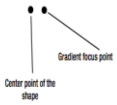


# Radial Gradient تدرج الألوان

- فهم تدرج اللون الشعاعي في التدرج اللوني الشعاعي ، تبدأ الألوان من نقطة واحدة ، وتنتقل بسلاسة إلى الخارج في شكل دائري أو بيضاوي. يتم تعريف التدرج اللوني الشعاعي باستخدام ثلاثة مكونات: شكل متدرج (مركز ونصف قطر دائرة التدرج)نقطة تركيز لها أول لون من التدرج اللوني يتوقف اللون




Hassan Ebrahem



# Radial Gradient تدرج الألوان

| Property | Data Type/Description |
|---|---|
| focusAngle | `Double` Set the angle in degrees from the center of the gradient to the focus point where the first color is mapped. |
| focusDistance | `Double` Set the distance from the center of the gradient to the focus point where the first color is mapped. |
| centerX | `Double` Set X coordinate of the center point of the gradient's circle. |
| centerY | `Double` Set Y coordinate of the center point of the gradient's circle. |
| radius | `Double` Set the radius of the circle of the color gradient. |
| proportional | `boolean` Set coordinates and sizes are proportional to the shape. |
| cycleMethod | `CycleMethod` Set the Cycle method applied to the gradient. |
| Stops | `List<Stop>` Set stop list for gradient's color |

Hassan Ebrahem

## Radial Gradient تدرج الألوان

- يمثل مثيل فئة RadialGradientتدرج لوني شعاعي. يحتوي الفصل على المُنشئين التاليين اللذين يختلفان في أنواع وسيطتهما الأخيرة: انحدار شعاعي (زاوية تركيز مزدوج ، تركيز مزدوج المسافة ، مركز مزدوج X، مركز مزدوج Y، نصف قطر مزدوج ، نسبي منطقي ، طريقة دورة طريقة الدورة ، قائمة <إيقاف> توقف)



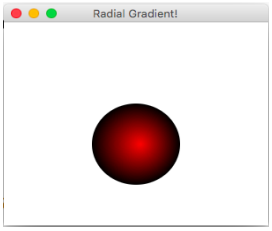
## Radial Gradient تدرج الألوان

- ) RadialGradientزاوية تركيز مزدوجة ، تركيز مزدوج المسافة ، مركز مزدوج X، مركز مزدوج Y، نصف قطر مزدوج ، منطقي متناسب ، دورة طريقة دورة ، توقف ... توقف)

```
Stop[] stops = new Stop[]{new Stop(0, Color.WHITE), new Stop(1, Color.BLACK)};
RadialGradient rg = new RadialGradient(0, 0, 0.5, 0.5, 0.5, true, NO_CYCLE, stops);
Circle c = new Circle(50, 50, 50);
c.setFill(rg);
```

## مثال

```
// Radial gradient color
RadialGradient gradient1 = new RadialGradient(0,
    .1,
    .5,
    .5,
    .5,
    true,
    CycleMethod.NO_CYCLE,
    new Stop(0, Color.RED),
    new Stop(1, Color.BLACK));

Circle c1 = new Circle(150, 150, 50);
c1.setFill(gradient1);


G.getChildren().add(c1);
```

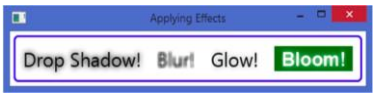



Hassan Ebrahem



## تطبيق التأثير Effects

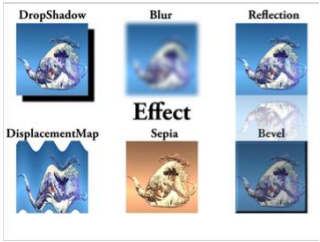
- ما هو التأثير؟ التأثير هو مرشح يقبل واحدًا أو أكثر من المدخلات الرسومية ، ويطبق خوارزمية على المدخلات ، وينتج مخرجات. عادةً ما يتم تطبيق التأثيرات على العقد لإنشاء واجهات مستخدم جذابة بصريًا. أمثلة على التأثيرات هي الظل ، الضبابية ، الالتفاف ، التوهج، الانعكاس ، المزج ، أنواع مختلفة من الإضاءة ، من بين أمور أخرى.
- ملاحظة: يتم تطبيق التأثير المطبق على مجموعة على كافة العناصر التابعة لها. من الممكن أيضًا ربط تأثيرات متعددة




Hassan Ebrahem

## تطبيق التأثير Effects على العقدة Node

- يمكنك تطبيق تأثير على عقدة باستخدام طريقة setEffect (). لهذه الطريقة ، تحتاج إلى تمرير كائن التأثير. لتطبيق تأثير على عقدة ، تحتاج إلى إنشاء العقدة. قم بتجسيد الفئة المعنية للتأثير المطلوب لتطبيقه. اضبط خصائص التأثير. قم بتطبيق التأثير على العقدة باستخدام طريقة setEffect ().






## تأثير التظليل

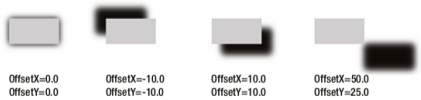
- تأثيرات التظليل يرسم تأثير التظليل ظلًا ويطبقه على أحد المدخلات.
  يدعم JavaFX ثلاثة أنواع من تأثيرات:
  - التظليل
  - شبح الهبوط
  - ظل

1. DropShadow
2. InnerShadow
3. Shadow

## تأثير التظليل DropShadow

- عند تطبيق هذا التأثير على العقدة ، سيتم إنشاء ظل خلف العقدة المحددة.
- الفئة المسماة DropShadow للحزمة javafx.scene.effect تمثل تأثير الظل المسقط.
- تحتوي فئة DropShadow على العديد من المنشئات التي تتيح لك تحديد القيم الأولية للخصائص:

- OffsetX , offsetY ,
- color, blurType,
- radius, spread,
- width, height,
- input.





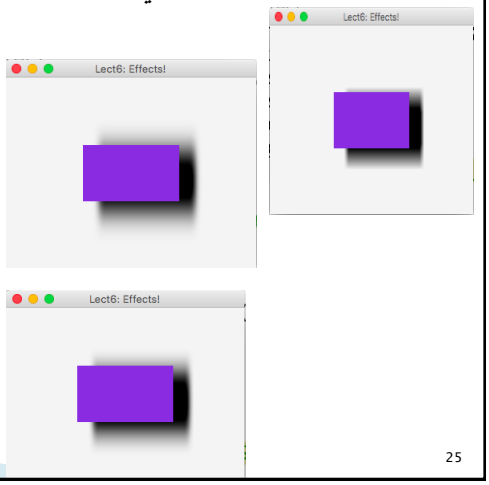

## تأثير التظليل DropShadow

- `DropShadow()`
- `DropShadow(BlurType blurType, Color color, double radius, double spread, double offsetX, double offsetY)`
- `DropShadow(double radius, Color color)`
- `DropShadow(double radius, double offsetX, double offsetY, Color color)`

## تأثير التظليل DropShadow

- تحدد الخاصية blurTypeنوع خوارزمية التمويه للظل. قيمته هي أحد الثوابت التالية في تعداد

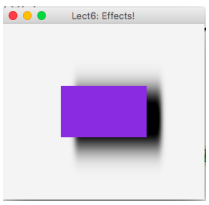
1. ONE_PASS_BOX
2. TWO_PASS_BOX
3. THREE_PASS_BOX
4. GAUSSIAN






## مثال

```
//DropShadow Effect
DropShadow dseffect = new DropShadow();
dseffect.setRadius(10);
dseffect.setOffsetX(20);
dseffect.setOffsetY(10);
dseffect.setSpread(0.2);
dseffect.setHeight(100);
dseffect.setBlurType(BlurType.TWO_PASS_BOX);

/////
Rectangle rect = new Rectangle(120, 70,Color.BLUEVIOLET);
rect.setEffect(dseffect);
////
root.getChildren().add(rect);
```



## تأثير التظليل InnerShadow

- InnerShadowيعمل تأثير InnerShadowبشكل مشابه جدًا لتأثير DropShadow. عند تطبيق هذا التأثير على العقدة ، سيتم إنشاء ظل داخل حواف العقدة.الفئة المسماة InnerShadow للحزمة javafx.scene.effectتمثل تأثير الظل الداخلي.
- Properties: OffsetX , offsetY , color, blurType, radius, choke, width, height, input

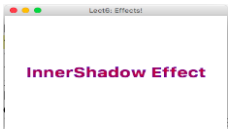
## تأثير التظليل InnerShadow

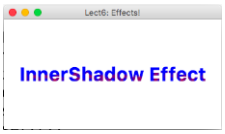
- InnerShadow()
- InnerShadow(BlurType blurType, Color color, double radius, double choke, double offsetX, double offsetY)
- InnerShadow(double radius, Color color)
- InnerShadow(double radius, double offsetX, double offsetY, Color color)

JavaFx

## مثال

```
Text t1 = new Text();
t1.setFont(Font.font("Airal", FontWeight.BOLD, 30));
t1.setText("InnerShadow Effect");
t1.setFill(Color.RED);

// InnerShadow Effect
InnerShadow iseffect = new InnerShadow();
iseffect.setRadius(5);
iseffect.setColor(Color.BLUE);
iseffect.setOffsetX(5);
iseffect.setOffsetY(7);
iseffect.setChoke(0.2);
iseffect.setBlurType(BlurType.THREE_PASS_BOX);

t1.setEffect(iseffect);
root.getChildren().add(t1);
```



Hassan Ebrahem



JavaFx

## تأثير الظل Shadow

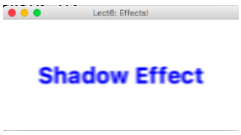
- تأثير الظل ينشئ تأثير الظل ظلًا بحواف ضبابية لمدخلاته. على عكس DropShadowو InnerShadow، فإنه يعدل المدخلات الأصلية نفسها لتحويلها إلى ظل.تحتوي فئة Shadowعلى العديد من المنشئات التي تتيح لك تحديد القيم الأولية للخصائص:الخصائص: اللون ، نوع التمويه ، نصف القطر ، العرض ، الارتفاع ، الإدخال
- Properties: Color, blurType, radius, width, height, input

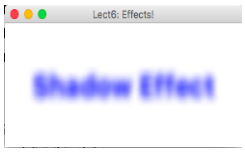
- Shadow()
- Shadow(BlurType blurType, Color color, double radius)
- Shadow(double radius, Color color)

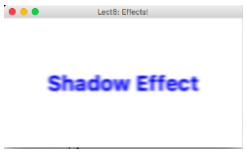
Hassan Ebrahem

## تأثير الظل Shadow

```
// Create a Shadow Effect
Shadow sh = new Shadow();
sh.setRadius(5);
sh.setColor(Color.BLUE);
sh.setBlurType(BlurType.ONE_PASS_BOX);
sh.setHeight(10);
sh.setWidth(20);
t.setEffect(sh);

root.getChildren().add(t);
```



## تأثير Bloom

- تأثير بلوم يضيف تأثير Bloomتوهجًا إلى وحدات البكسل الخاصة بإدخالها والتي لها لمعان أكبر من أو يساوي حدًا معينًا. يمثل مثيل فئة Bloomتأثير .Bloomيحتوي على خاصيتين: 1. عتبة 2. الإدخال الخاصية Thresholdهي رقم بين 0.0 و 1.0. قيمته الافتراضية 0.30 ؛ هذا يمثل قيمة عتبة لمعان وحدات البكسل للعقدة. تحتوي فئة Bloomعلى مُنشئين:

  - `Bloom()`
  - `Bloom(double threshold)`

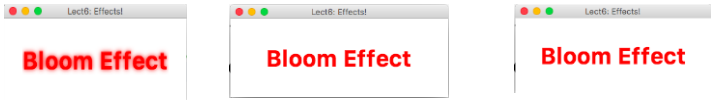
## مثال

```
Text t1 = new Text("Bloom");
t1.setFill(Color.YELLOW);
t1.setFont(Font.font(null, FontWeight.BOLD, 24));
t1.setEffect(new Bloom(0.10));
Rectangle r1 = new Rectangle(100, 50, Color.GREEN);
StackPane sp1 = new StackPane(r1, t1);
```

```
// Create a Bloom Effect
Bloom BE = new Bloom();
BE.setThreshold(0.0);
t.setEffect(BE);
```

```
// Create a Bloom Effect
Bloom BE = new Bloom();
BE.setThreshold(0.3);
t.setEffect(BE);
```

```
// Create a Bloom Effect
Bloom BE = new Bloom();
BE.setThreshold(0.8);
t.setEffect(BE);
```




Hassan Ebrahem

33

## تأثير الانعكاس Reflection

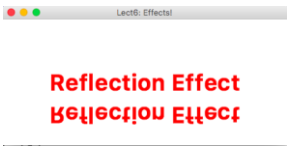
- تأثير الانعكاس يضيف تأثير الانعكاس انعكاسًا للمدخلات أسفل المدخلات. يمثل مثيل فئة الانعكاس تأثير انعكاس. يتم التحكم في موضع الانعكاس وحجمه وشفافيته بخصائص مختلفة
- 1. TopOffset 2.fraction 3.topOpacity 4.bottomOpacity 5.input
- 1: الإزاحة العلوية 2. الكسر 3. السعة العلوية 4. السفلية السفلية 5. الإدخال تحتوي فئة الانعكاس على صانعين:
  - Reflection()
  - Reflection(double topOffset, double fraction, double topOpacity, double bottomOpacity)

Hassan Ebrahem

34

## مثال

```
// Create Reflection Effect
Reflection RS = new Reflection();
RS.setTopOffset(10);
RS.setFraction(1);
RS.setTopOpacity(1);
RS.setBottomOpacity(1);

t.setEffect(RS);
```




```
// Create Reflection Effect
Reflection RS = new Reflection();
RS.setTopOffset(10);
RS.setFraction(0.8);
RS.setTopOpacity(0.5);
RS.setBottomOpacity(0.8);

t.setEffect(RS);
```

Hassan Ebrahem

35

## منظور تأثير التحويل

- منظور تأثير التحويل يعطي تأثير PerspectiveTransform للعقدة ثنائية الأبعاد مظهرًا ثلاثي الأبعاد عن طريق تعيين الزوايا إلى مواقع مختلفة. يمثل مثيل فئة PerspectiveTransform تأثيرPerspectiveTransform. يحتوي الفصل على ثماني خصائص لتحديد إحداثيات x و y للزوايا الأربع:
- ulx uly urx ury lrx lry llx lly
- ulx uly urx ury lrx lry llx lly
- يشير الحرف الأول في أسماء الخصائص ( u أو l) إلى العلوي والسفلي. يشير الحرف الثاني في أسماء الخصائص ( l أو r) إلى اليسار واليمين. يشير الحرف الأخير في أسماء الخصائص (س أو ص) إلى إحداثي س أو ص لزاوية.

Hassan Ebrahem

36

## منظور تأثير التحويل

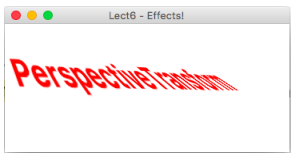
- تحتوي فئة PerspectiveTransform على مُنشئين:

  - PerspectiveTransform()
  - PerspectiveTransform(double ulx, double uly, double urx, double ury, double lrx, double lry, double llx, double lly)

```
// PerspectiveTransform Effect 1
PerspectiveTransform pt = new PerspectiveTransform();
pt.setUlx(10);
pt.setUly(10);
pt.setLlx(10);
pt.setLly(90);
pt.setUrx(300);
pt.setUry(40);
pt.setLrx(300);
pt.setLry(70);
t.setEffect(pt);
```

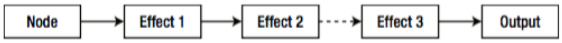
## مثال

```
// PerspectiveTransform Effect 2
PerspectiveTransform effect = new PerspectiveTransform();
effect.setUlx(0);
effect.setUly(0.0);
effect.setUrx(250.0);
effect.setUry(20.0);
effect.setLrx(310.0);
effect.setLry(60.0);
effect.setLlx(20.0);
effect.setLly(60.0);
t.setEffect(effect);
```



## Chaining Effects آثار التسلسل

- آثار التسلسل يمكن ربط بعض التأثيرات بتأثيرات أخرى عند تطبيقها بالتسلسل. يصبح ناتج التأثير الأول مدخلاً للتأثير الثاني وما إلى ذلك ، كما هو موضح في الشكل التالي:




- سلسلة من التأثيرات المطبقة على العقدة

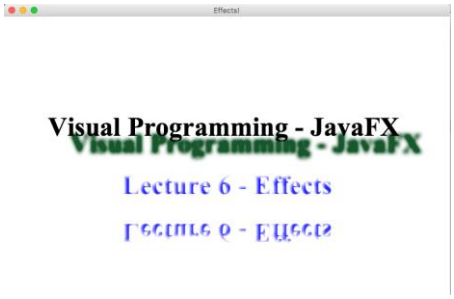


## مثال

```
// PerspectiveTransform Effect 1
PerspectiveTransform pt = new PerspectiveTransform();
pt.setUlx(10);
pt.setUly(10);
pt.setLlx(10);
pt.setLly(90);
pt.setUrx(300);
pt.setUry(40);
pt.setLrx(300);
pt.setLry(70);
t.setEffect(pt);
 //PerspectiveTransform Effect >> DropShadow Effect 2
DropShadow dseffect = new DropShadow();
dseffect.setRadius(10);
dseffect.setOffsetX(20);
dseffect.setOffsetY(10);
dseffect.setSpread(0.2);
dseffect.setBlurType(BlurType.GAUSSIAN);
pt.setInput(dseffect);

//DropShadow Effect  >> Bloom effect 3
Bloom BE = new Bloom();
BE.setThreshold(0);
dseffect.setInput(BE);

root.getChildren().add(t);
```

## واجب

- اكتب التعليمات البرمجية المصدر لتنفيذ واجهة المستخدم التالية باستخدام التخطيط المناسب.






## نهاية المحاضرة