



جامعة طرابلس كلية تقنية المعلومات



البرمجة المرئية Visual Programming

ITSE423

أستاذ المادة - حسن علي حسن
h.ebrahim@uot.edu.ly

المحاضرة الثانية

Stage-Scene-Node





مواضيع المحاضرة

► ماهو المسرح Stage

- عرض Stage الرئيسية
- إنهاء البرنامج
- تحديد حدود Stage
- تهيئة منطقة Stage

► ماهو المشهد Scene

► ماهي العقدة Node

- ▶ يعتبر المسرح Stage في JavaFX هو الحاوية العليا وتستضيف بداخلها المشهد Scene الذي يتكون من عناصر مرئية.
- ▶ تمثل فئة Class Stage الموجودة في الحزمة `javafx.stage` المسرح في تطبيق JavaFX.
- ▶ يتم إنشاء `primary stage` بواسطة النظام الأساسي `platform` ويتم تمريرها إلى `start(Stage s)`.
- ▶ ملاحظة: Stage في تطبيق JavaFX هي الحاوية العليا. هذا لا يعني أنه يتم عرضه دائمًا كنافذة منفصلة. على سبيل المثال ، في بيئة الويب ، يتم تضمين المرحلة الأساسية لتطبيق JavaFX داخل نافذة المتصفح.

- ▶ كائن Stage object يجب أن يكون :
 - تم إنشاؤه وتعديله في `JavaFX Application Thread`.
 - يتم استدعاء `start()` في `JavaFX Application Thread`.
- ▶ يتم إنشاء `primary Stage` وتمريرها إلى `start()`.
- ▶ لاحظ أن `primary Stage` التي يتم تمريرها إلى `start()` لا تظهر، وبالتالي تحتاج إلى استدعاء `show()` لإظهارها.
- ▶ تلميح Tip: يجب أن ترث كل `application class` من `Application` class لكي يتم تنفيذ `start(Stage stage)`.



عرض Stage الرئيسية

```
package runningapp;

import javafx.application.Application;
import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

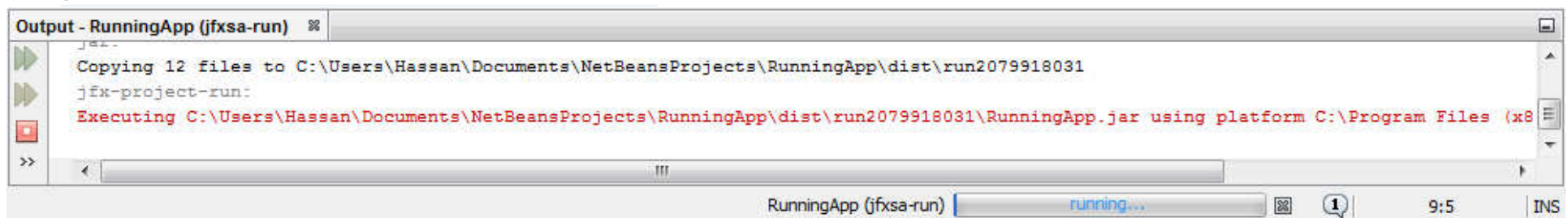
    @Override
    public void start(Stage primaryStage) {
        //Do not write any code
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        launch(args);
    }
}
```

▶ في المثال، لا تحتوي start() على اسطر برمجية.

▶ عند تنفيذ التطبيق، لا تظهر أي نافذة ، ولا تظهر أي مخرجات على console.

▶ ويبقى البرنامج يشتغل دون توقف .ever-running program





إنهاء البرنامج

البرنامج السابق يبقى يشتغل ولكي يتم
إيقافه يتم استخدام
Platform.exit()
داخل **start()** تعالج المشكلة
ever-running program

```
1  package runningapp;
2
3  import javafx.application.Application;
4  import javafx.application.Platform;
5  import javafx.stage.Stage;
6
7  /**
8   *
9   * @author Hassan
10  */
11  public class RunningApp extends Application {
12
13      @Override
14      public void start(Stage primaryStage) {
15          Platform.exit();
16      }
17
18      /**
19       * @param args the command line arguments
20       */
21      public static void main(String[] args) {
22          launch(args);
23      }
24  }
```



```
10 public class RunningApp extends Application {
11
12     @Override
13     public void start(Stage primaryStage) {
14         //Platform.exit();
15         primaryStage.show();
16         primaryStage.close();
17     }
18
19     /**
20      * @param args the command line arguments
21      */
22     public static void main(String[] args) {
23         launch(args);
24     }
25 }
```

► يمكن ايضا انهاء البرنامج السابق باستخدام `close()` مع stage ولكن يجب أن يتم عرض stage أولاً.

► لأن الدالة `close()` لا تقوم بإنهاء stage طالما stage غير نشطة.

► تلميح Tip: سيتم اثناء البرنامج مباشرة بعد تنفيذه.



البدء في تشغيل التطبيق

▶ أنت الآن جاهز لتشغيل تطبيق JavaFX الأول الخاص بك. يمكنك

استخدام أحد الخيارين التاليين لتشغيله:

- ليس من الضروري أن يكون لديك `main()` لبدء تشغيل التطبيق. حيث يتم تنفيذ

الدالة `.init()`.

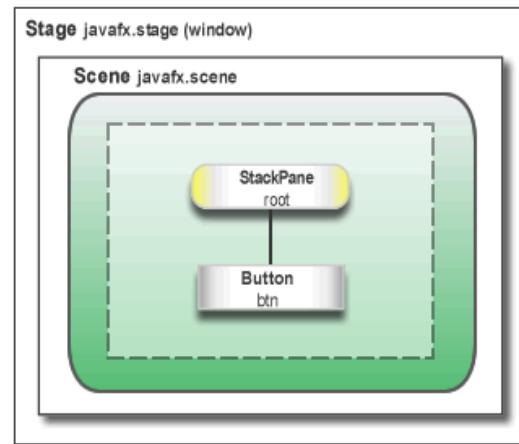
- إذا تم تضمين `main()` في تطبيق JavaFX فيتم استدعاء `launch()` التابعة

للفئة `Application class` لتشغيل التطبيق.

تحديد حدود المسرح Stage

- ▶ تتكون حدود Stage من أربع خصائص: x و y و $width$ و $height$. تحدد الخاصيتين (x, y) موقع stage في الزاوية العلوية اليسرى.
- ▶ تحدد الخاصيتين العرض $width$ والارتفاع $height$ الحجم.

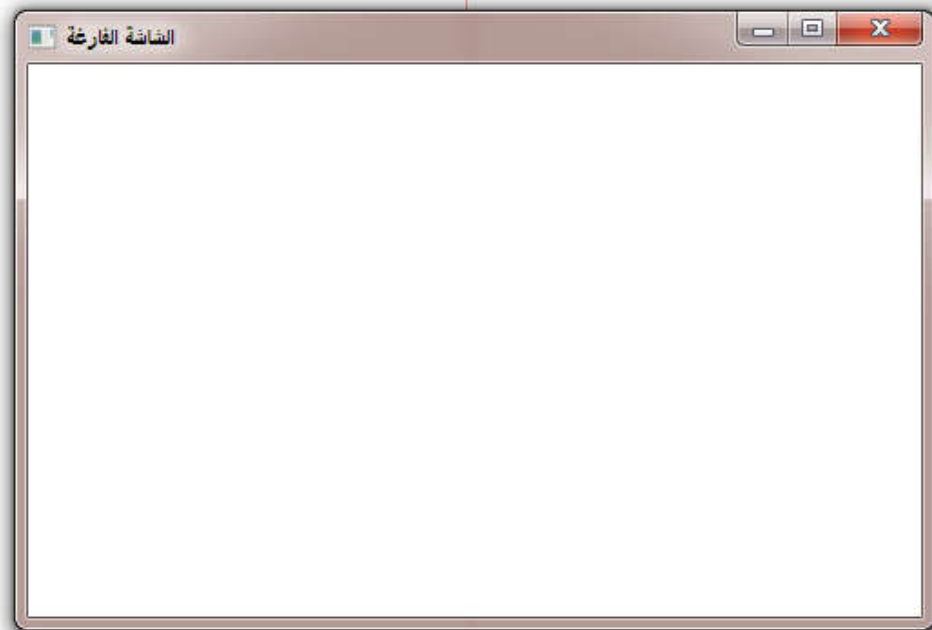
◦ تلميح Tip: عندما لا تحتوي stage على scene ولم يتم تعيين موضعها وحجمها بشكل صريح، بالتالي يتم تحديد موضعها وحجمها بواسطة المنصة `.platform`





مثال: تحديد حدود المسرح Stage

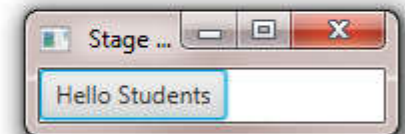
```
3 import javafx.application.Application;
4 import javafx.stage.Stage;
5
6 /**
7  *
8  * @author Hassan
9  */
10 public class RunningApp extends Application {
11
12     @Override
13     public void start(Stage primaryStage) {
14         //Platform.exit();
15         primaryStage.setTitle("الشاشة الفارغة");
16         primaryStage.show();
17         //primaryStage.close();
18     }
19
20     /**
21     * @param args the command line arguments
22     */
23     public static void main(String[] args) {
24         launch(args);
25     }
26 }
```



نلاحظ أن الخلفية لونها أبيض بشكل افتراضي.

- ▶ سنقوم بالتعديل على البرنامج السابق.
- ▶ سيتم اضافة زر Button داخل Group ثم يتم اضافتها إلى Scene. تكون start() المعدلة كما يلي:

```
3 import javafx.application.Application;
4 import javafx.scene.Group;
5 import javafx.scene.Scene;
6 import javafx.scene.control.Button;
7 import javafx.stage.Stage;
8
9 /**
10  *
11  * @author Hassan
12  */
13 public class RunningApp extends Application {
14
15     @Override
16     public void start(Stage primaryStage) {
17         primaryStage.setTitle(" Stage with Button");
18         Group root = new Group(new Button("Hello Students"));
19         Scene scene = new Scene(root);
20         primaryStage.setScene(scene);
21         primaryStage.show();
22         //primaryStage.close();
23     }
```

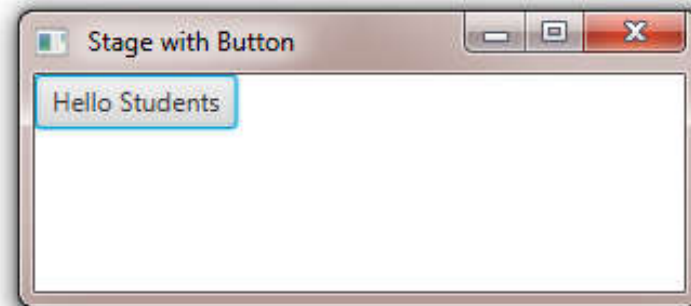


► لضبط عرض وارتفاع Scene إلى 300 و 100، يكون على النحو التالي:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle(" Stage with Button");
        Group root = new Group(new Button("Hello Students"));
        Scene scene = new Scene(root, 300, 100);
        primaryStage.setScene(scene);
        primaryStage.show();
        //primaryStage.close();
    }
}
```



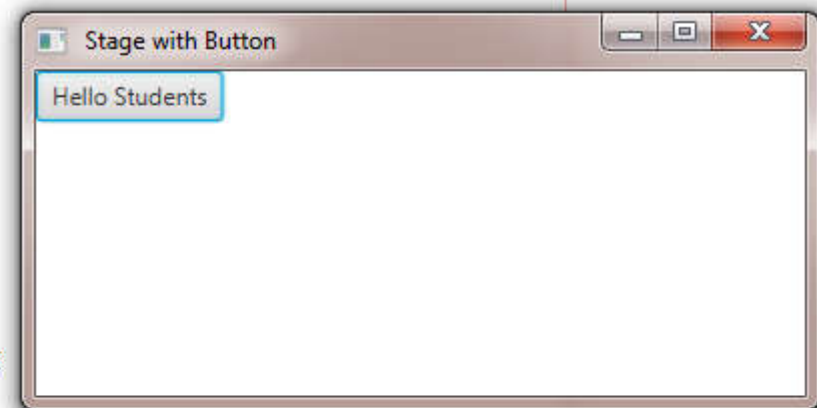
► يمكن التعديل في حجم stage و Scene باستخدام الكود التالي:

```
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.control.Button;

import javafx.stage.Stage;

/**
 *
 * @author Hassan
 */
public class RunningApp extends Application {

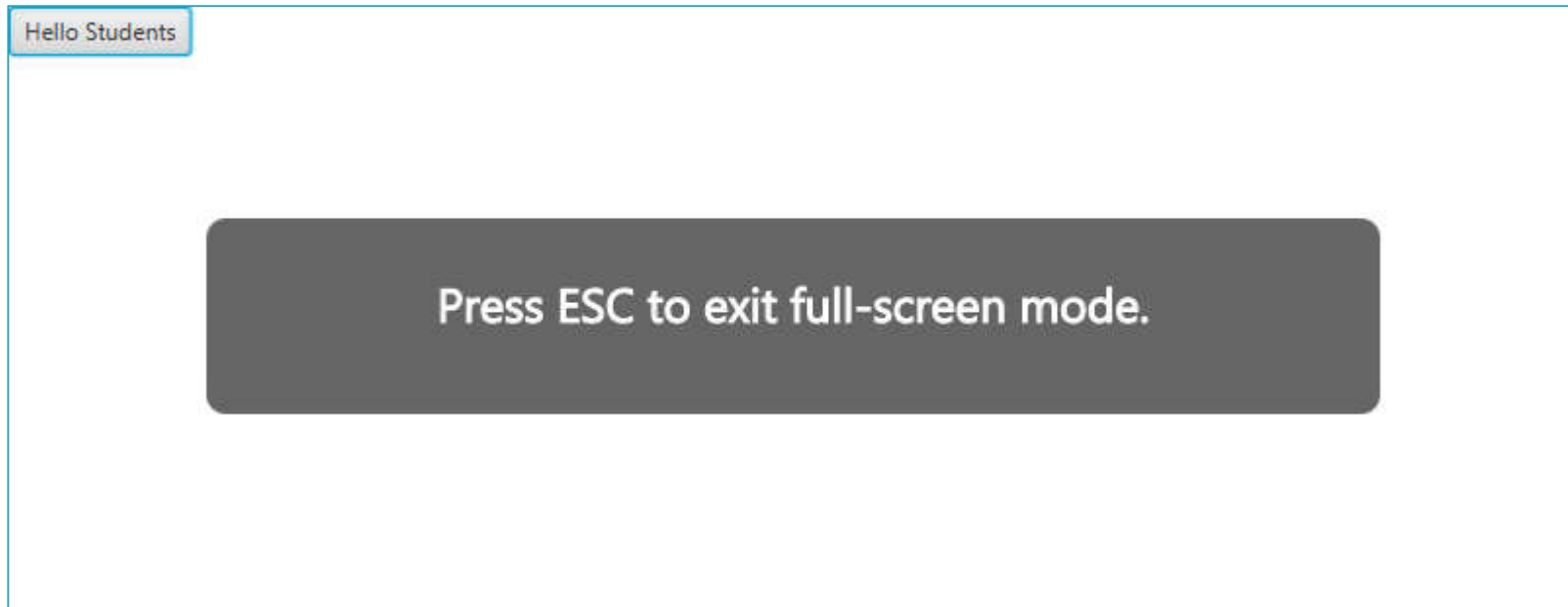
    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle(" Stage with Button");
        Group root = new Group(new Button("Hello Students"));
        Scene scene = new Scene(root, 300, 100);
        primaryStage.setScene(scene);
        primaryStage.setWidth(400);
        primaryStage.setHeight(200);
        primaryStage.show();
        //primaryStage.close();
    }
}
```



- ▶ يمكن وضع Stage في وضع ملء الشاشة full-screen عن طريق استدعاء `.setFullScreen(true)`.
- ▶ عندما يتم عرض Stage في وضع ملء الشاشة، يتم عرض رسالة مختصرة حول كيفية الخروج من وضع ملء الشاشة، يتم الضغط على مفتاح ESC للخروج من وضع ملء الشاشة.
- ▶ يمكنك الخروج من وضع ملء الشاشة برمجياً عن طريق استدعاء `.setFullScreen(false)`.
- ▶ كما يمكن استخدام `isFullScreen()` للتحقق مما إذا كانت stage في وضع ملء الشاشة أو لا.



عرض Stage في وضع ملء الشاشة



```
public class RunningApp extends Application {  
  
    @Override  
    public void start(Stage primaryStage) {  
        primaryStage.setTitle(" Stage with Button");  
        Group root = new Group(new Button("Hello Students"));  
        Scene scene = new Scene(root, 300, 100);  
        primaryStage.setScene(scene);  
        // primaryStage.setWidth(400);  
        // primaryStage.setHeight(200);  
        primaryStage.setFullScreen(true);  
        primaryStage.show();  
        //primaryStage.close();  
    }  
}
```



تهيئة منطقة المسرح Stage

► يمكن تقسيم منطقة stage إلى قسمين: منطقة المحتوى content

area والزخارف decorations.

► تعرض منطقة المحتوى content المحتويات المرئية الخاص بي scene.

► تتكون الزخارف decorations عادةً من شريط عنوان title bar والحدود borders.

► في JavaFX، خصائص Stage تحدد لون الخلفية والزخارف، يمكنك الحصول على الأنواع الخمسة التالية من Stages :

- 1) Decorated 2) Undecorated 3) Transparent
- 4) Unified 5) Utility



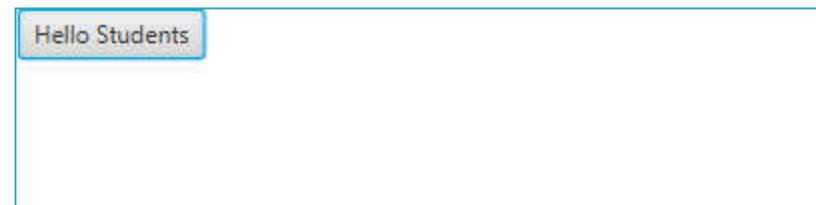
تهيئة منطقة المسرح Stage

▶ الأنواع الخمسة التالية من Stages :

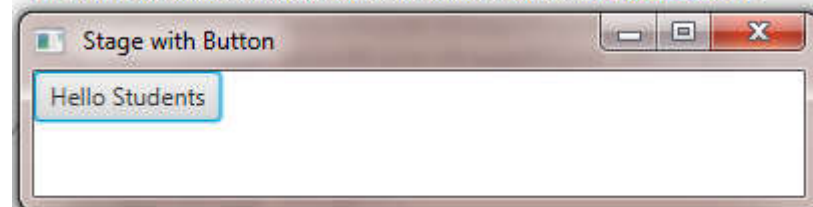
- ▶ 1) Decorated 2) Undecorated 3) Transparent
- 4) Unified 5) `StageStyle.UTILITY`

▶ ملاحظة: إن الأنواع الخمسة تحدد الزخارف فقط. يتم التحكم في لون الخلفية من خلال خلفية scene الخاصة به ، والتي تكون بيضاء بشكل افتراضي.

```
public void start(Stage primaryStage) {  
    primaryStage.setTitle(" Stage with Button");  
    Group root = new Group(new Button("Hello Students"));  
    Scene scene = new Scene(root, 300, 100);  
    primaryStage.setScene(scene);  
    primaryStage.setWidth(400);  
    primaryStage.setHeight(100);  
    primaryStage.initStyle(StageStyle.UTILITY);  
}
```



```
primaryStage.initStyle(StageStyle.DECORATED);
```



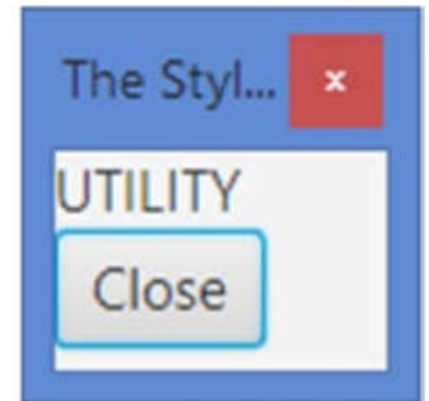
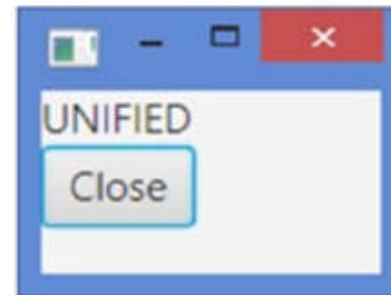
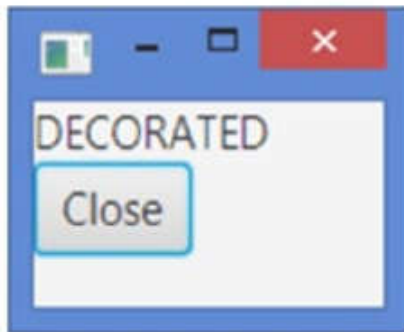
- ▶ يمكن التحكم في درجة شفافية Stage باستخدام التالي:
 - ▶ `primaryStage.setOpacity(0.5);`
- ▶ يمكن التحكم في عدم السماح بتكبير أو تصغير Stage
 - ▶ `primaryStage.setResizable(false);`
- ▶ كما يمكن تحديد أعلى قيمة وأقل قيمة للعرض أو الارتفاع
 - ▶ `setMinWidth()`, `setMinHeight()`, `setMaxWidth()`, and `setMaxHeight()`
 - ▶ `primaryStage.setMinWidth(200);`



تهيئة منطقة المسرح Stage

▶ الأنواع الخمسة التالية من Stages :

- ▶ 1) Decorated
- 2) Undecorated
- 3) Transparent
- 4) Unified
- 5) Utility



- ▶ يمثل Scene المحتويات المرئية الخاصة بي Stage.
- ▶ تمثل الفئة Scene class في الحزمة javafx.scene المشهد في البرنامج.
- ▶ يتم إرفاق Scene object واحد الى stage في كل مرة.
- ▶ إذا تم إرفاق scene متصل stage أخرى، فيجب أن يتم فصله أولاً عن stage السابقة.
- ▶ المسرح Stage يحتوي على Scene واحد في أي وقت.

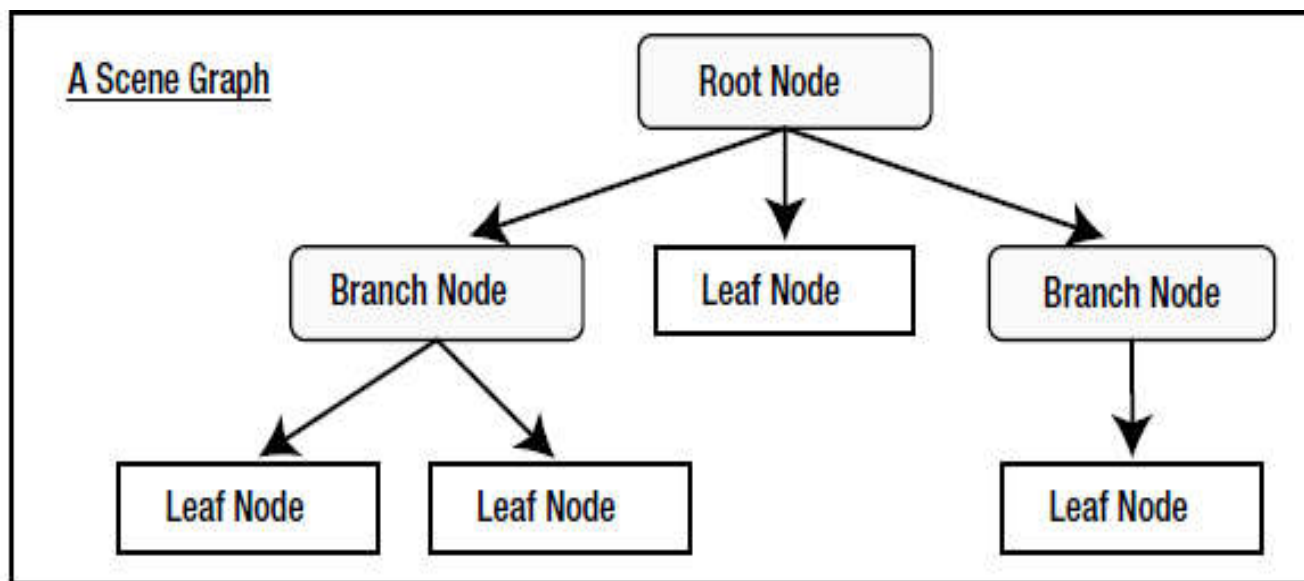


مثال: المشهد Scene

```
public void start(Stage primaryStage) {  
    Button btn = new Button();  
    btn.setText("Say 'First Scene'");  
  
    Group root = new Group();  
    root.getChildren().add(btn);  
  
    Group root1 = new Group();  
  
    Scene scene = new Scene(root, 300, 250);  
    Scene scene1 = new Scene(root1, 400, 200);  
    primaryStage.setTitle(" First Scene ! ");  
    primaryStage.setScene(scene);  
    primaryStage.show();  
}
```

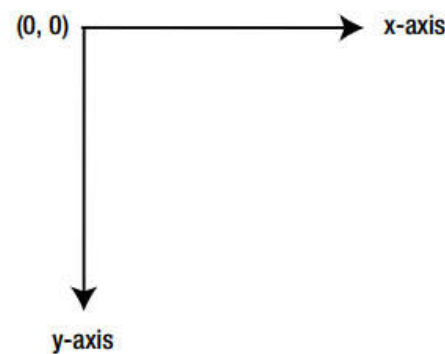
- ▶ يتكون Scene من scene graph والتي تحتوي على عقد مرئية .nodes
- ▶ يعمل Scene كحاوية لكل العقد scene graph.
- ▶ يكون scene graph كشجرة تحتوي على عقد، يمكن أن تكون العقدة إما عقدة فرعية branch node أو عقدة طرفية leaf node.
- ▶ يمكن أن تحتوي العقدة الفرعية على عقد أبناء children nodes، بينما لا يمكن للعقدة الطرفية.
- ▶ تسمى العقدة الأولى في scene بعقدة الجذر root node.
- ▶ العقدة node في scene graph عبارة عن كائن من `.javafx.scene`

► يمكن أن تحتوي root node على عقد أبناء children nodes؛ ومع ذلك ، فإنها لا تحتوي على عقدة أب parent node أعلى منها. يوضح الشكل التالي ترتيب العقد Nodes في Scene Graph.



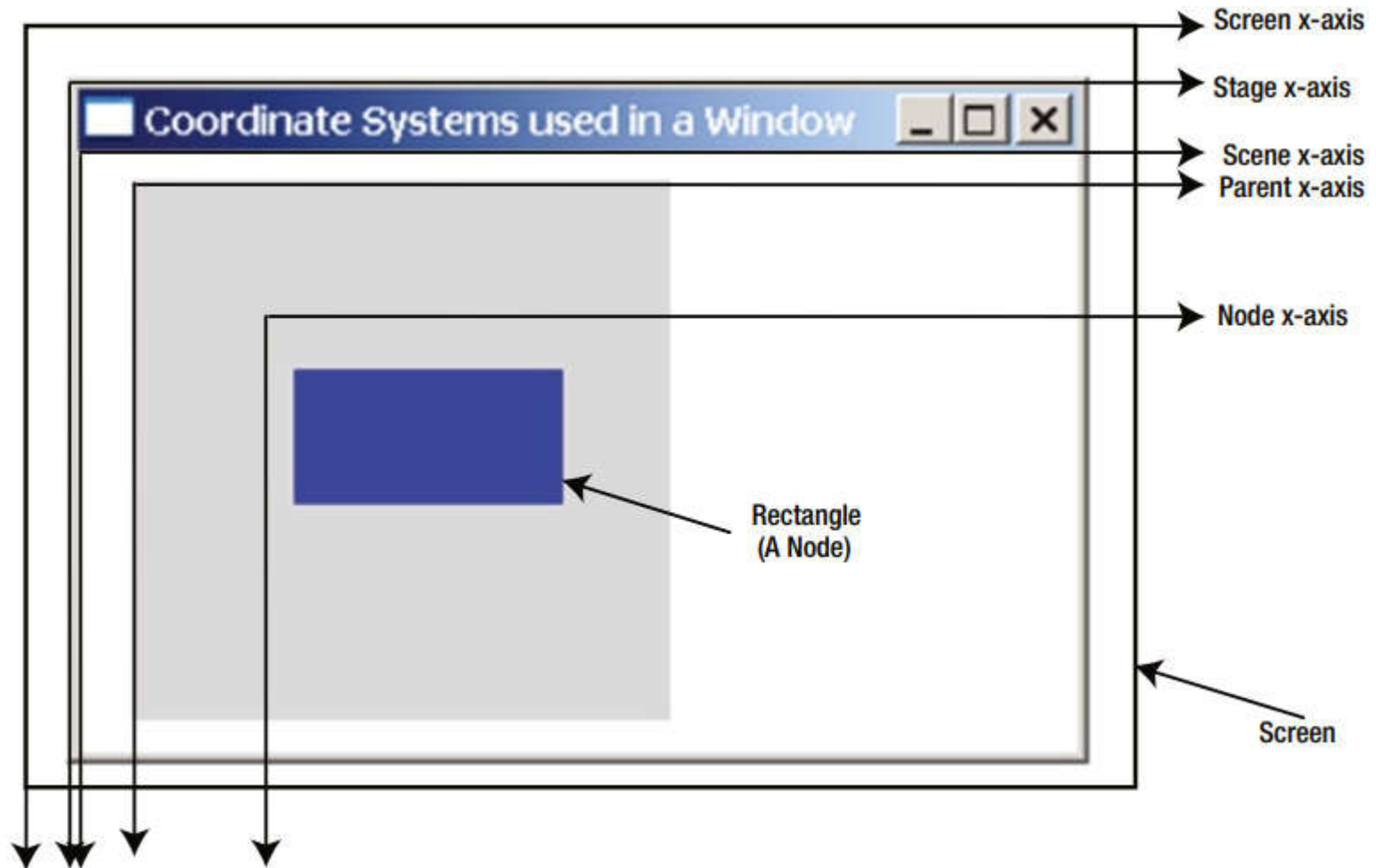
- ▶ كل عنصر في scene graph يسمى عقدة Node.
- ▶ ويوجد من ضمن المكتبة javafx.scene .
- ▶ يمكن أن تحتوي العقدة على عناصر فرعية (تسمى أيضًا children) ،
وتسمى هذه العقدة العقد الفرعية branch أو Parent.
- ▶ العقدة الفرعية Parent تكون مثل Group و WebView، بينما العقدة الطرفية مثل Rectangle و Text و ImageView .
- ▶ فقط عقدة واحدة داخل كل شجرة لن يكون لها أب parent، وهو ما يشار إليها بالعقدة الجذرية root node.

- ▶ كل عقدة Node في scene graph لها إحداثيات خاص بها. تستخدم العقدة نظام إحداثيات يتكون من محور X ومحور Y . في أنظمة الكمبيوتر ، تزيد القيم الموجودة على المحور X جهة اليمين وتزداد القيم على المحور Y إلى أسفل .

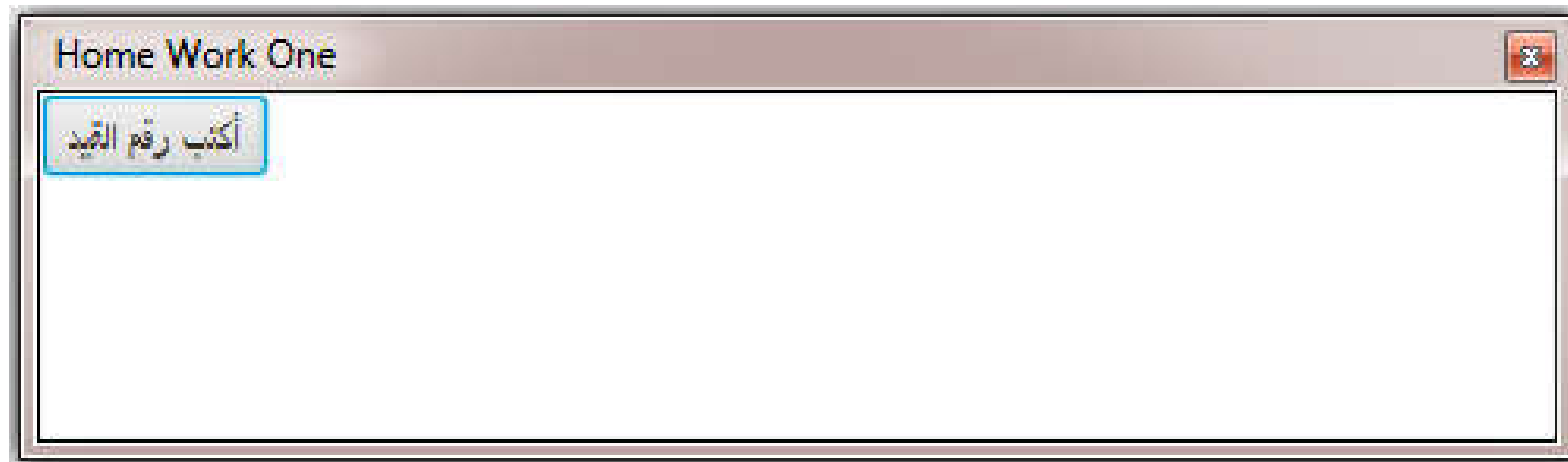


Simplified Node's Coordinate System

► في تطبيق واجهة المستخدم الرسومية GUI، يتم وضع العقد داخل أبنائهم. العقدة الجذرية root node هي الأصل لجميع العقد ويتم وضعها داخل المشهد Scene. يتم وضع Scene داخل Stage ويتم وضع Stage داخل الشاشة Screen. كل عنصر يتكون من نافذة ، له نظام إحداثيات خاص به، كما هو موضح في الشكل التالي:



▶ أكتب برنامج لعرض واجهة المستخدم التالية:



- ملاحظة: 1- اسم البرنامج يكون باسم Homework ملحق برقم قيد الطالب.
- 2- تستبدل كلمة «أكتب رقم القيد» برقم قيد الطالب داخل البرنامج.

- ▶ يعتبر المسرح Stage في JavaFX هو الحاوية العليا، وتستضيف بداخلها العناصر المرئية. حيث تم توضيح كيف يتم تعريفه وكيف يتم تحديد موقعه على الشاشة وتحديد الحدود الخاص به، كما تم توضيح المشهد Scene حيث يمثل المحتويات المرئية الخاصة بي Stage. كل عنصر في scene graph يسمى عقدة Node. يمكن أن تحتوي العقدة على عناصر فرعية (تسمى أيضًا children)، وتسمى هذه العقدة العقد الفرعية branch أو Parent.
- ▶ فقط عقدة واحدة داخل كل شجرة لن يكون لها أب parent، وهو ما يشار إليها بالعقدة الجذرية root node.

نهاية المحاضرة

