

## Exercise 1: Creating the Monster Factory

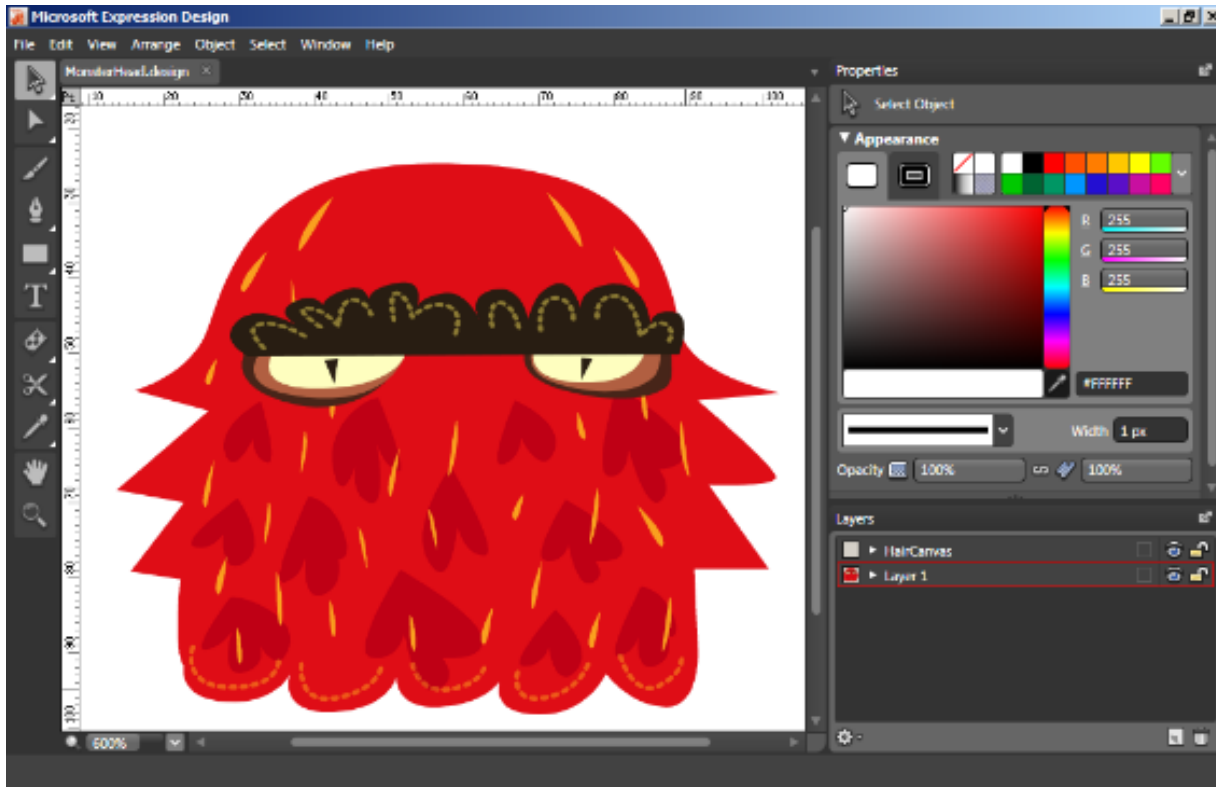
Language Filter: All

In this exercise you will use the **Expression Design** to build a **XAML** monster picture. Then you will consume the **XAML** monster picture using **JavaScript** and **Silverlight**.

### Task 1 – Editing a XAML monster picture

In this task you create the mouth of the monster using **Expression Design**.

1. Open **Expression Design** from **Start | All Programs | Microsoft Expression | Microsoft Expression Design**.
2. Open the **MonsterHead.design** file located in the **SilverlightMonsterFactory\Ex01-CreatingTheMonsterFactory\begin\xaml** folder. To do so, from the **Expression Design** menu, select **File | Open** and then browse to the **SilverlightMonsterFactory\Ex01-CreatingTheMonsterFactory\begin\xaml\MonsterHead.design** file. You should see the following image.

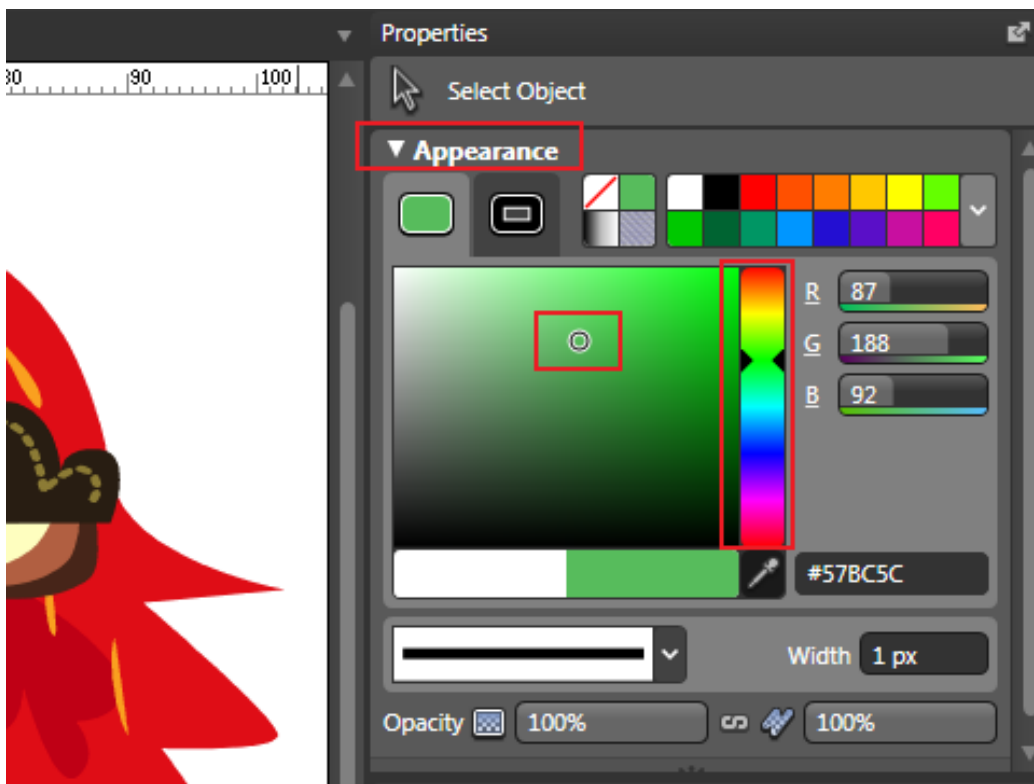


**Figure 1**  
Opening the *MonsterHead.design* in the Expression Design Editor

#### Note:

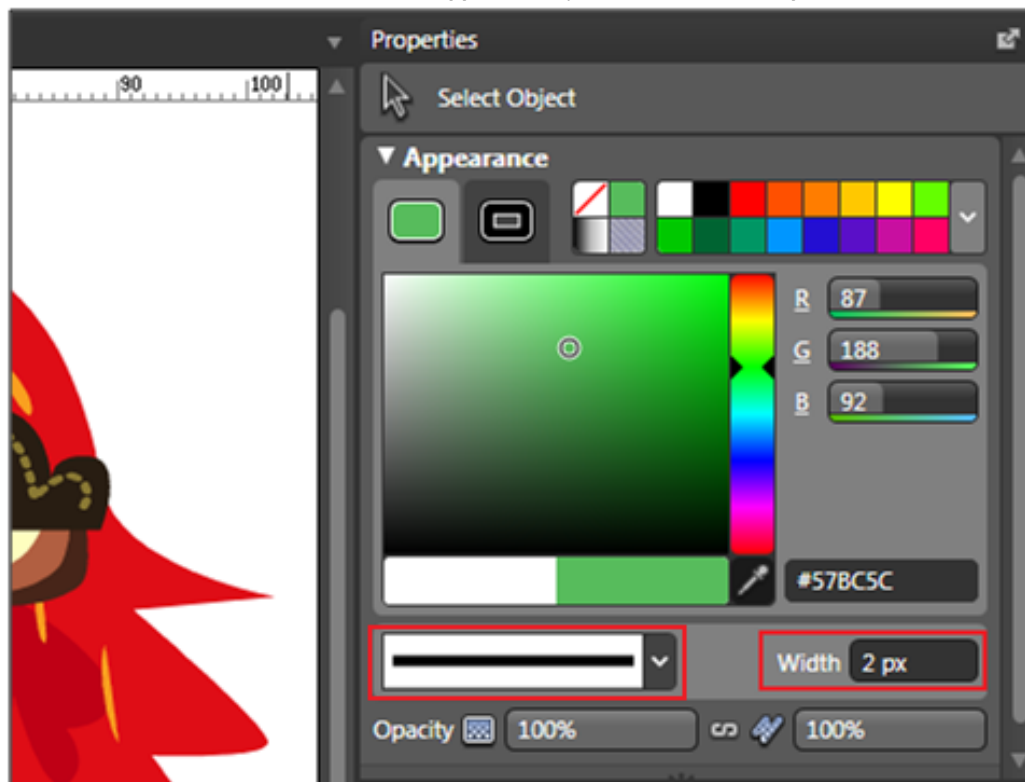
In the **Layers** panel (bottom-left) there are two layers. *Layer 1* contains the picture and *HairCanvas* layer will be use later in the exercise to place hair into the monster.

3. You will set up the monster mouth. To do so, from the **Appearance** palette, select the mouth color.



**Figure 2**  
*Selecting the fill color*

4. Select the stroke thickness. To do so, in the **Appearance** palette, set **Width** to **2px** and the line stroke from the combo box.



**Figure 3**  
*Setting the stroke thickness*

5. Set the stroke color. To do so, in the **Appearance** palette click on the stroke area and select a color.

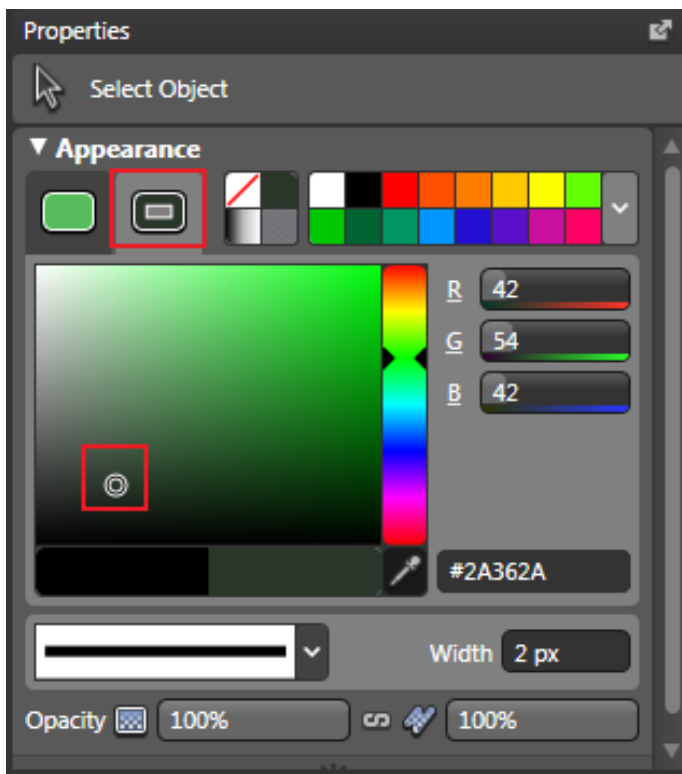


Figure 4

Selecting the stroke color

6. Select the **B-Spline**. To do so, on the left panel, right click on the **Pen** icon and select **B-Spline**.

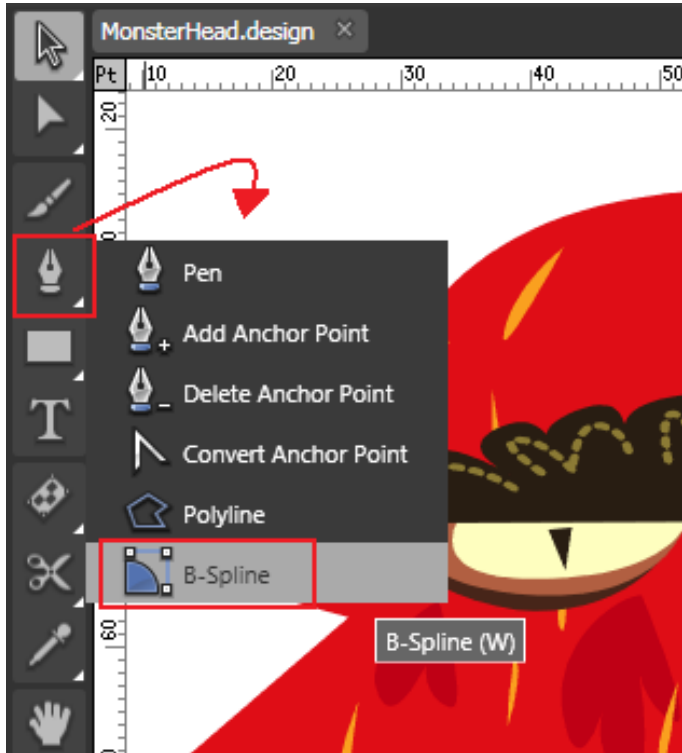
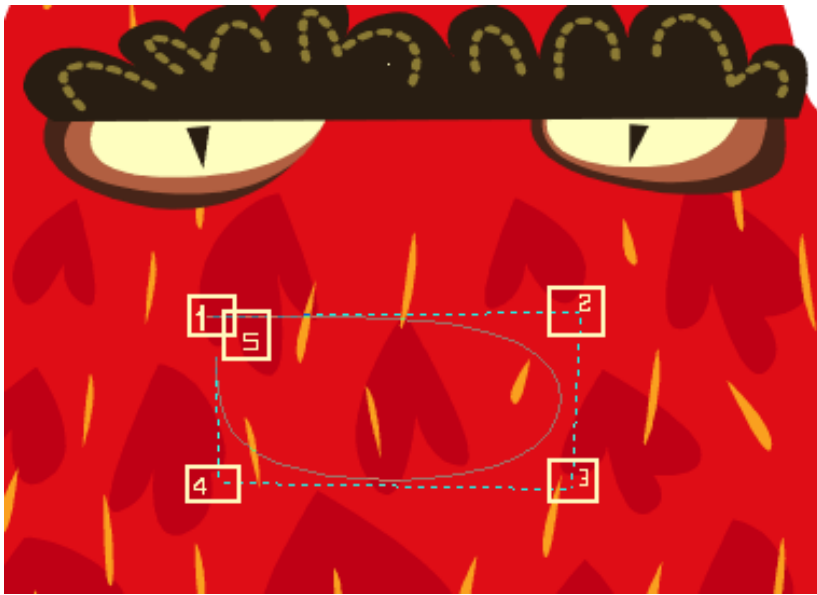


Figure 5

Selecting B-Spline

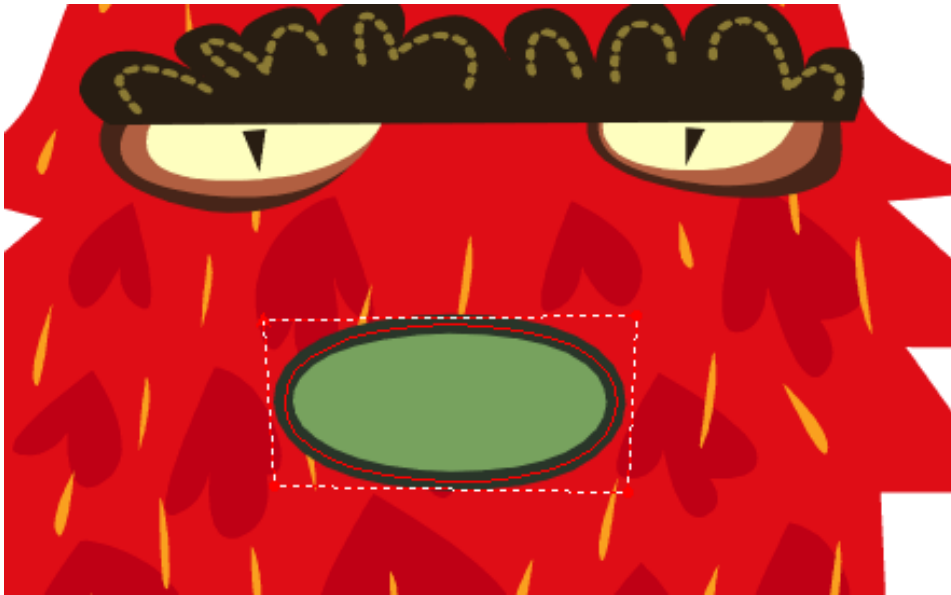
7. Try to create a rectangle by clicking on its vertices. Those points are called "anchor points".



**Figure 6**

*Setting the anchor points*

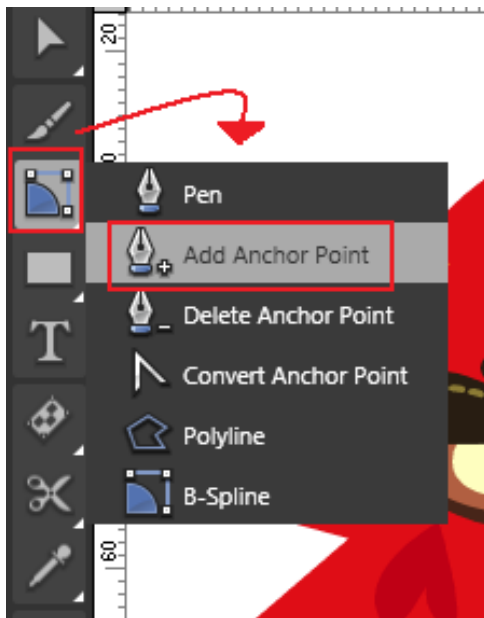
8. You should see an output similar to the following.



**Figure 7**

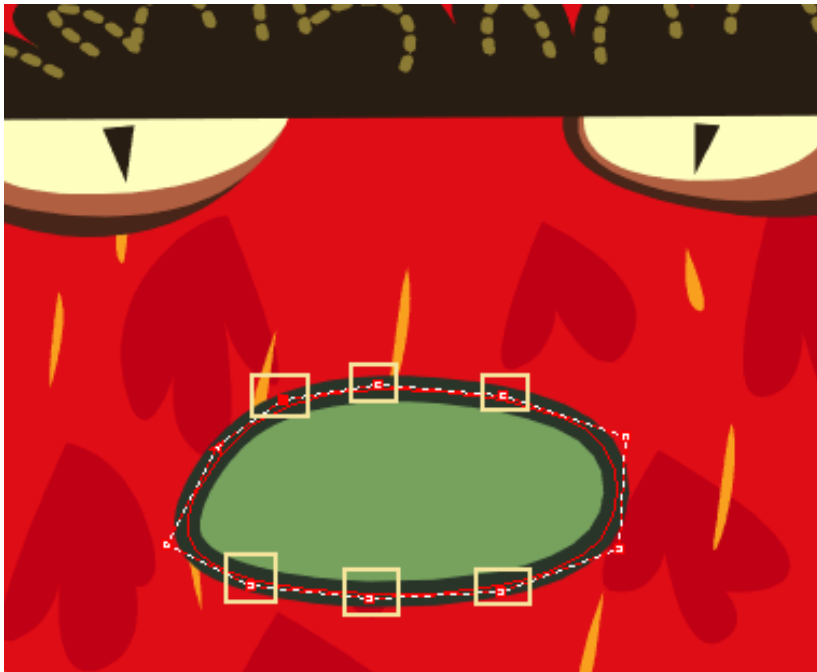
*Monster Mouth*

9. Now, you will modify the monster mouth. To do so, in the right panel, right click the **B-Spline** and select **Add Anchor Point**.



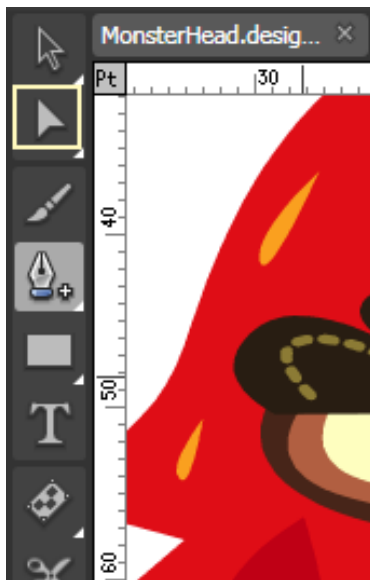
**Figure 8**  
*Using the Add Anchor Point utility*

10. Click in some point around the mouth path to, then, change the circle shape.



**Figure 9**  
*Adding Anchor Points*

11. In the left panel, click on the **Direct Selection** tool.



**Figure 10**

*Selecting the Direct Selection tool*

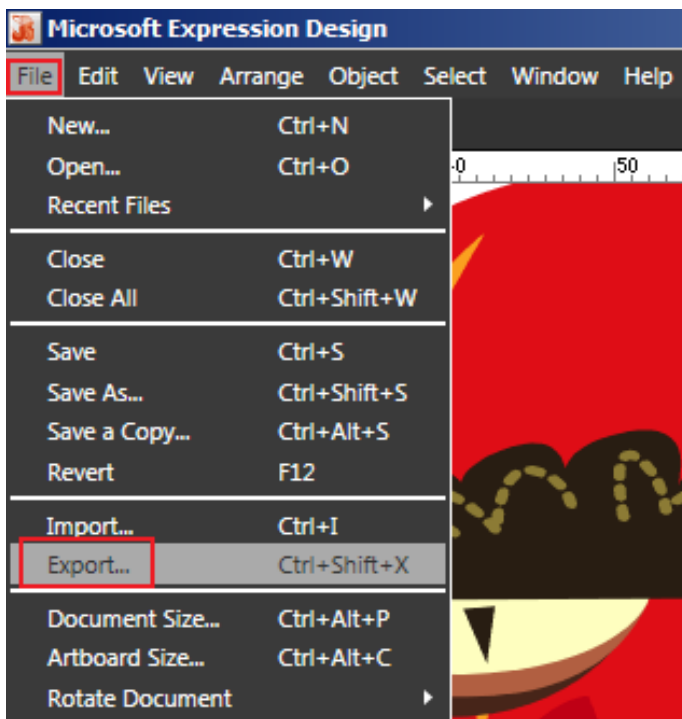
12. Using the **Direct Selection** tool, select an **Anchor Point** and move it. Move more points if you want to, and turn the monster mouth into one you like.



**Figure 11**

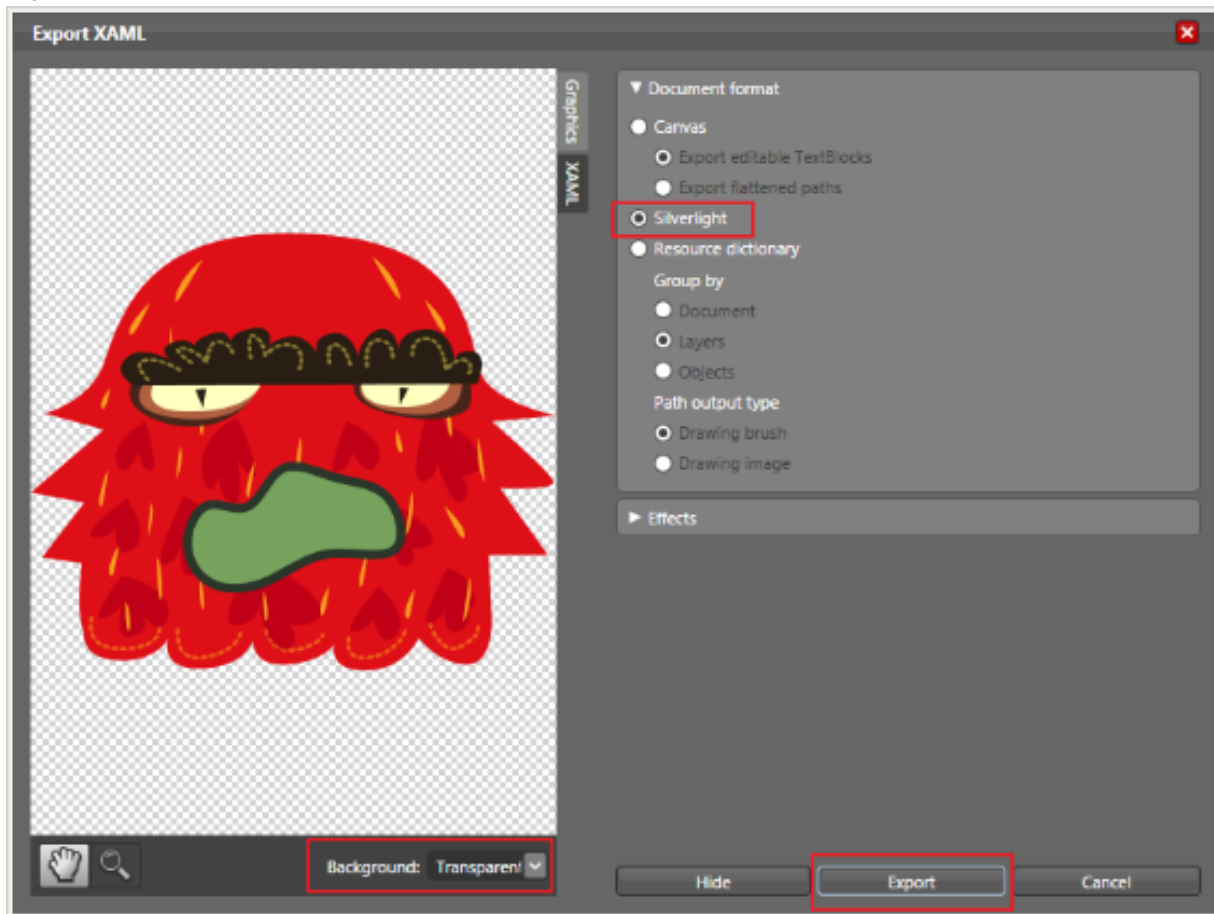
*Moving the Anchor Points*

13. Export the new item template to a **XAML** file. To do so, from the **File** menu, select **Export**. Save it in the **SilverlightMonsterFactory\Ex01-CreatingTheMonsterFactory\begin\xaml** folder and name it **MonsterHead**. Ensure that the file type is **XAML**.



**Figure 12**  
Exporting the monster item template

14. In the **Export XAML** dialog window, ensure that **Background** is set to **Transparent** and the **Document Format** is set to **Silverlight**. Then click the **Export** button.



**Figure 13**  
Exporting the monster picture

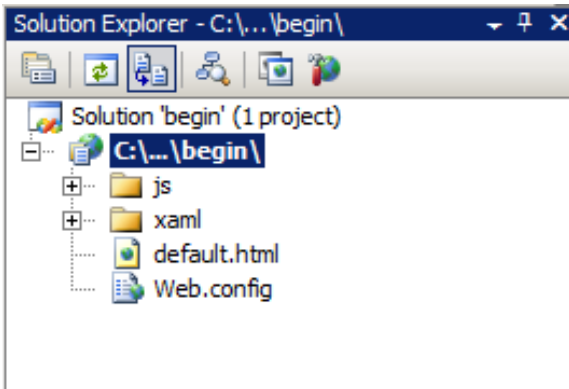
15. Now, close the **Expression Design**. It is not necessary to save the design.

## Task 2 – Create a monster clicking on the browser window.

In this task you create a **JavaScript** handler to create monster in every mouse click.

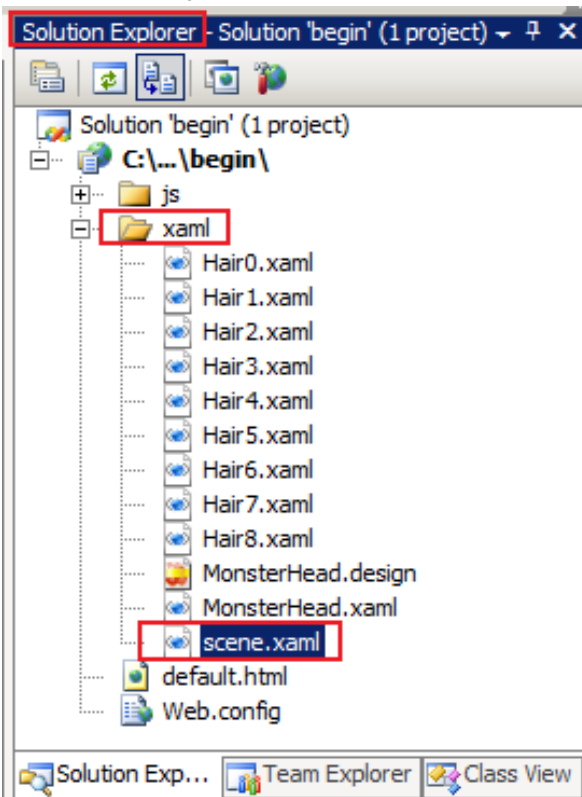
1. Open **Microsoft Visual Studio 2008** from **Start | All Programs | Microsoft Visual Studio 2008 | Microsoft Visual Studio 2008**.

- In **Visual Studio**, open the provided web site. To do so, press **Alt + Shift + O**. Browse to **SilverlightMonsterFactory\Ex01-CreatingTheMonsterFactory\begin** folder and click the **Open** button. You should see the following in the **Solution Explorer** window.



**Figure 14**  
*Solution Tree*

- In the **Solution Explorer**, double click on the **scene.xaml** file inside the **xaml** folder.



**Figure 15**  
*Opening the scene.xaml file*

- Add the following line of code in the **scene.xaml** file. This adds an event handler to the *left button click* event on the canvas.

#### XAML



```
<Canvas
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Width="1500" Height="1000" Loaded="onLoaded" x:Name="MonsterLand" Background="#235153" MouseLeftButtonDown="onMouseClicked">
```

- Press **Ctrl + S** to save the file.
- In the **Solution Explorer**, double click on the **MonsterLand.js** file inside the **js** folder.



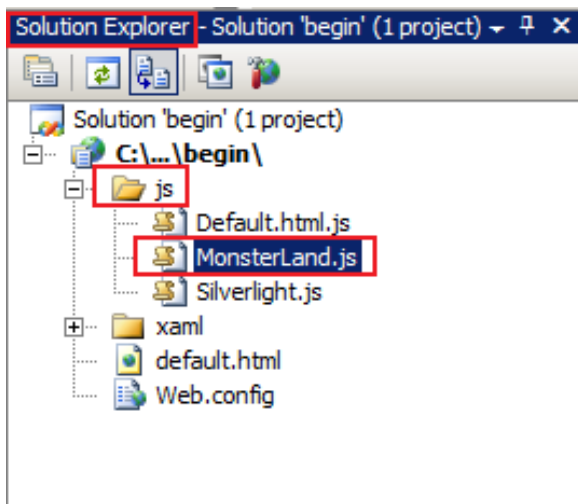


Figure 16

Opening *MonsterLand.js* file**Note:**

The **MonsterLand.js** file contains some **JavaScript** functions that load asynchronously the **MonsterHead.xaml** and the **Hair\*.xaml** files into the browser when the **default.html** page is loaded in the browser using some **JavaScriptobjects** from **Silverlightlibrary**. It also has some code to handle **MouseLeftButtonDown** event that you will complete in the following steps.

7. Add the following code inside the **addMonster** function to create an object from the **xaml** file.

**JAVASCRIPT**

```
function addMonster(x, y)
{
    var monster = slPlugin.content.createFromXaml(itemXaml, true);
}
```

**Note:**

The **createFromXaml** function creates a **JavaScript** object from the **xaml** text that the **itemXaml** variable contains. In this way you may modify the **xaml** with an object notation as you will notice in the following steps.

8. Add the following code to set the location of the monster and its parent control.

**JAVASCRIPT**

```
function addMonster(x, y)
{
    var monster = slPlugin.content.createFromXaml(itemXaml, true);

    //set the location of the monster
    monster["Canvas.Left"] = x;
    monster["Canvas.Top"] = y;

    //add the monster to the MonsterCanvas
    monsterCanvas.Children.add(monster);
}
```

9. Edit the **OnMouseClick** function with the following code to locate the monster in the mouse position.

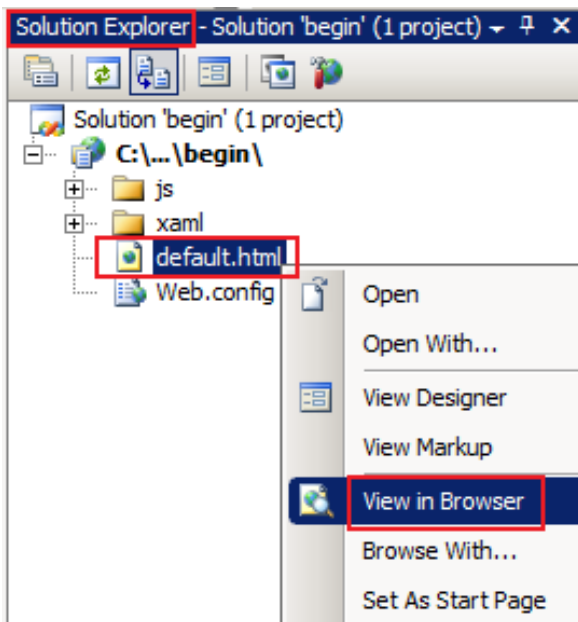
**JAVASCRIPT**

```
function onMouseClick(sender, mouseArgs)
{
    //calculate the monster position to be centered around the mouse
    var x = mouseArgs.getPosition(monsterCanvas).x - 75;
    var y = mouseArgs.getPosition(monsterCanvas).y - 75;

    addMonster(x, y);
}
```

10. Press **Ctrl + S** to save the file.

11. In **Solution Explorer**, right click **default.aspx** file and select **View in Browser**.



**Figure 17**

*View default.aspx page in the browser*

- Click on the browser window in random places and you should see an output similar to the following.



**Figure 18**

*Sample output.*

- Close the browser.

### Task 3 – Customize each monster

In this task you add a different hair to each monster.

- In the **Solution Explorer**, double click on the **MonsterLand.js** file inside the **js** folder.

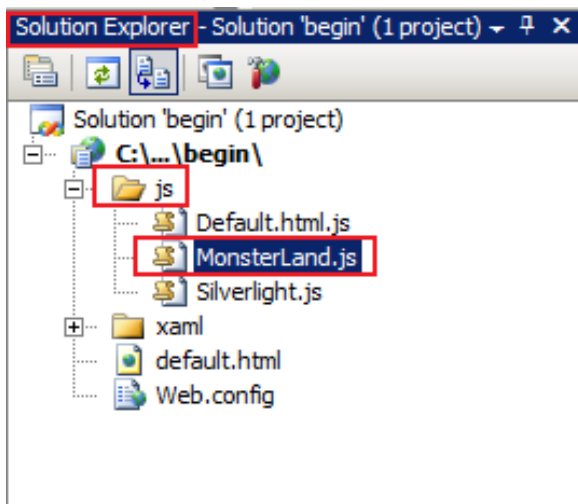


Figure 19

Opening *MonsterLand.js* file

2. Add the following code to the **AddMonster** function to select a random hair.

## JAVASCRIPT



```
function addMonster(x, y)
{
    var monster = slPlugin.content.createFromXaml(itemXaml, true);

    //pick a random hair, and create its visuals
    var rnd = Math.floor(Math.random() * numHairTypes);
    var hairXaml = hairXamlArray["xaml/Hair" + rnd + ".xaml"];

    //set the location of the monster
    monster["Canvas.Left"] = x;
    monster["Canvas.Top"] = y;

    //add the monster to the MonsterCanvas
    monsterCanvas.Children.add(monster);
}
```

3. Add the following code to add the hair **xaml** into the monster **xaml**.

## JAVASCRIPT



```
function addMonster(x, y)
{
    var monster = slPlugin.content.createFromXaml(itemXaml, true);

    //pick a random hair, and create its visuals
    var rnd = Math.floor(Math.random() * numHairTypes);
    var hairXaml = hairXamlArray["xaml/Hair" + rnd + ".xaml"];
    var newHair = slPlugin.content.createFromXaml(hairXaml, true)

    //add the hair to the new monster
    monster.findName("HairCanvas").Children.add(newHair);

    //set the location of the monster
    monster["Canvas.Left"] = x;
    monster["Canvas.Top"] = y;

    //add the monster to the MonsterCanvas
    monsterCanvas.Children.add(monster);
}
```

**Note:**

In these lines of code you are adding the hair **xaml** into the *HairCanvas* that was noticed in the *Task1*. This was an empty layer from the **MonsterHead.xaml**.

4. Press **Ctrl + S** to save the file.

