IST400/600 Scripting for Games, Spring 2010
Lab 3
Instructor: Keisuke Inoue

Description:

In this lab, you will create the "Messy Room" game that was discussed in the previous class. The following instructions will help you to create the game, but will not tell you all the small steps. The lab is designed to encourage your learning and enhance your understanding. **To maximize your learning experience, you are encouraged to work on it independently. You may ask questions to instructors and/or friends (but keep it quiet) and look up previous lecture materials, textbook, and other resources. Do not copy-and-paste any resources other than you created.**

Peer Survey:

In this lab, we will be doing a peer survey instead of the peer evaluation we have been doing previously. You will be in one of the two groups: Group 1 for undergraduate students and Group 2 for the graduate or non-matriculated students. After you finish your lab, review other people's games, and select one person (other than yourself), who you think made the best game. Your lab's evaluation will be done by the instructor (10 points) but you will receive extra 2 points, if your game was chosen as the best game. All the students must vote for someone! (Your lab will not be graded, until you finish your lab and survey.)

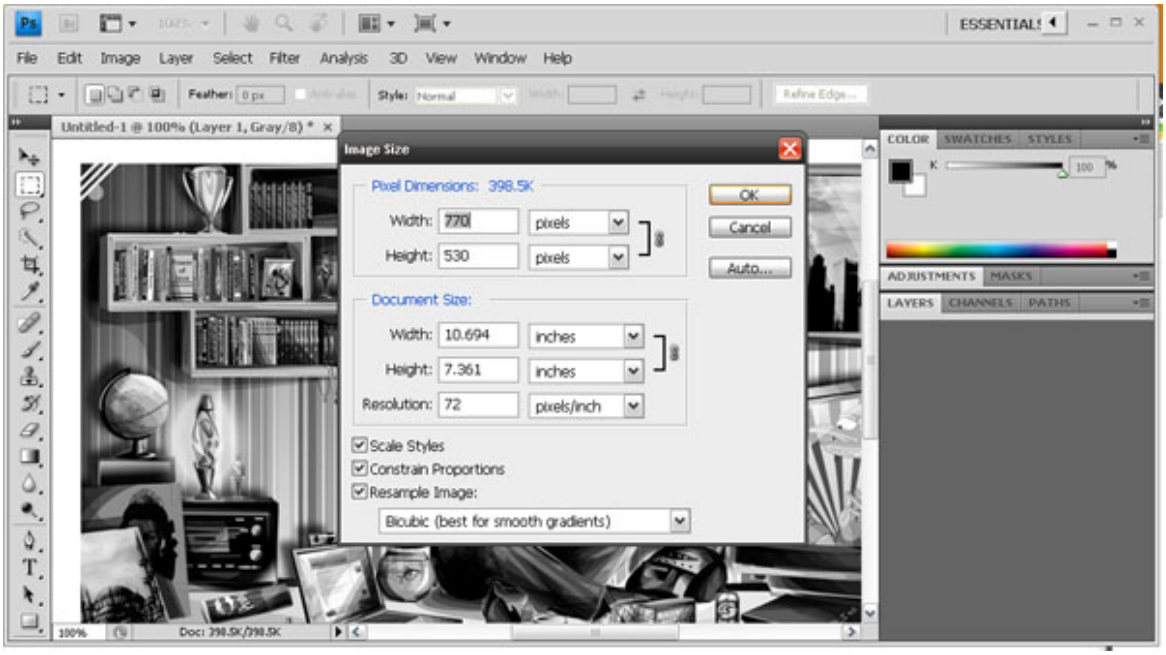
Game Description:

The player will be given a list of items and picture. In order to win this game, the player has to find and click all the items on the picture within a given time.

Lab Instructions:

1. **Preparing an Image**

   In this lab, you need an image with an approximate size 800 (width) x 600 (height). Your image does not have to be exactly in this size, but it has to fit within the display. If you need to resize your image, you can use Adobe Photoshop, which is installed in the lab PC. To resize the image, open the file with Photoshop, and select "Image" → "Image Size…" If you have troubles, let the instructor know.



2. Displaying the game

2.1) Question

Create a new file on your text editor, and save it with a name “Lab3.html” (or whatever you want). As always, insert <script> and <body> elements.

Now, insert a description of the game in the beginning of the <body> element. Say something like “Find the following items.” You can use simple texts.

2.2) Item list

Below the question text, insert a <form> element and name it “itemList”. In the form element, insert at least three check boxes for the items that you want the player to find on the picture. Name each checkbox differently, and set the “disabled” attribute. The code should look like the following:

```
<form name=”itemList”>
  <input type=”checkbox” disabled=”true” name=”laptop”>Laptop</input> <br />
  …
</form>
```

2.3) Image

Below the question, insert an image that you brought. In this lab, save your image file in the same folder as the HTML file. Use a relative path to specify the image, for example:

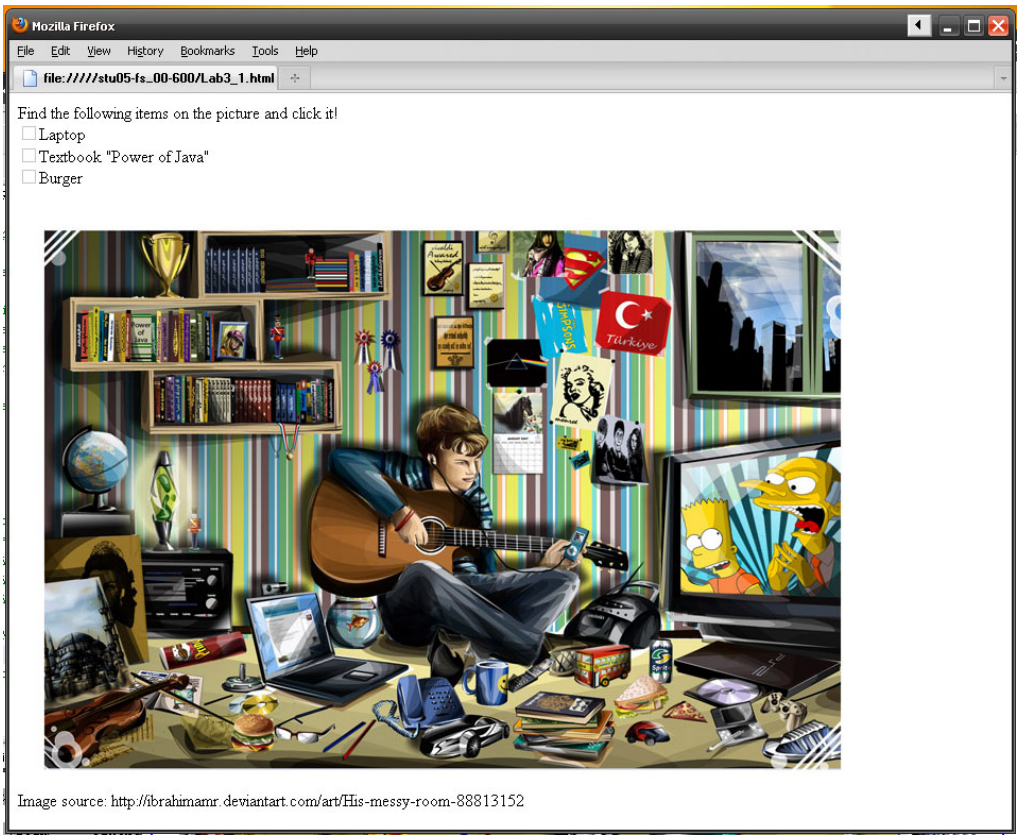
```
<img src=”myImage.jpg” > <!-- myImage.jpg is in the same folder. -->
```

Do not adjust the size of the image in the <img> tag (e.g. Do not use width or height attributes.).

2.4) Image source (optional)

If you copied an image from some public domain, insert a text to site the image source.

At this point, save your file and open the file on the browser. It should show everything you need for the game, except the timer.



3. Implementing the logic of the game

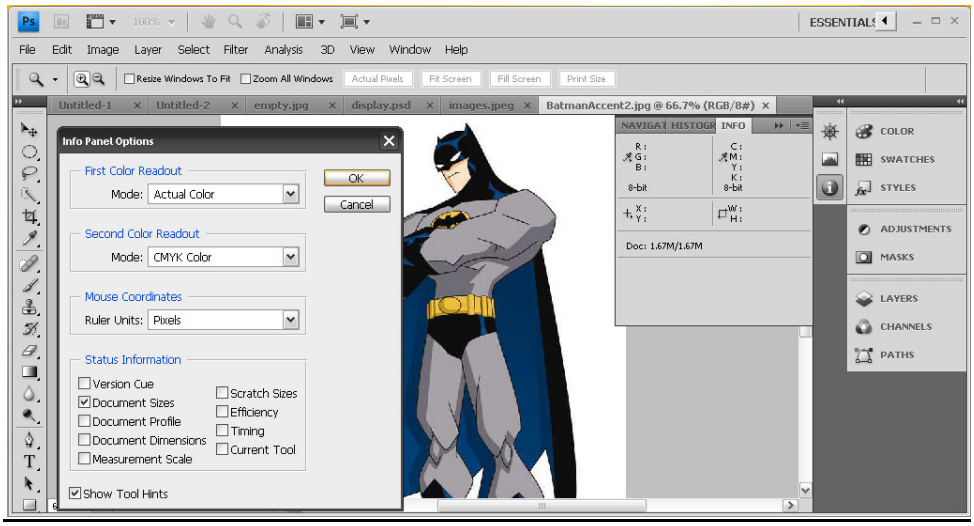
3.1) Defining Map

In the body element, insert a <map> element, with a name “map1” (or whatever you want). In the map, define the shapes that corresponds to the items that you want the player to find. There are three possible shapes: “circle”, “rect”, and “polygon” but you do not need to use all of the shapes. Make sure to name the shapes differently.

In the <img> tag, add the “usemap” attribute to specify the <map>. Remember to put the hash sign (“#”) in front of the name of the <map> element.

```
<map name=”map1”>
  <!-- You define areas here -->
</map>
…
<img usemap=”#map1” src=”myImage.jpg”>
```

To figure out the coordinate of the object, open the image file with Photoshop, and use the “Info” window (which pops up if you click “Window” → “Info”.). Open the “Panel Options…” on the upper right corner of the Info window, and make sure that “Ruler Unit” of “Mouse Coordinates” is “Pixels”. The X: and Y: on the Info window shows the coordinates of the mouse.



<u>If you have problems in finding the coordinates of the items using Photoshop, ask the instructor.</u>

3.2) Defining a function to update the item list

In the <script> element, define a function called “foundItem()” with one parameter “item”. The function needs to do the following:

1. Check if the parameter “item” matches to one of the items you specified in the “itemList” check boxes, and if so, update the value of the check box. (You need to use an if-statement for each item.) For example, if the item is “laptop”, the if-statement for the item should look like:

   ```
   if (item == ‘laptop’) { document.itemList.laptop.checked = true; }
   ```

2. Check if all the items have been found. If so, Display a message saying the player won the game.

3.4) Resetting the item list

In order to reset the item list, define a function named “resetAll()” that assigns “false” to all the check box values. Call the function in the “onLoad” event in the <body> tag.

```
<script>
  …
  Function resetAll()
```

```
  {
    // Rest the check box variables here.
  }
</script>
<body onLoad="resetAll();">
…
```

3.3) Defining events

Now, go back to the <map> element. For each <area> element, set the "onClick" attribute to call the "foundItem()" function you just defined. Make sure to set the appropriate parameter.

At this point, save the file and reload the file on the browser. Your game should work fine, except the timer. Play around a bit to make sure everything is working. If you have troubles making it work, ask for help.

4. Implementing a timer

4.1) Displaying a remained time

In the itemList <form> element, insert a line to display a remained time. You need to mix simple texts and text field. Remember to use <input> element with type="text". Name it "timeLeft".

4.2) Declaring a gTimeLeft global variable

In the beginning of the <script> element, declare a global variable, "gTimeLeft", and initialize it with 15 (or whatever the time you want in seconds). This variable will be used to maintain the remained time.

4.3) Starting a timer

In the next line (still in the <script> element), define a variable called "timer", and initialize it by calling a function "setInterval()". This line should look like the following:

```
var timer = setInterval(updateTimer, 1000);
```

This line specifies the function "updateTimer()" should be called every 1000 milliseconds (= 1 second).

4.4) Defining a timer function

Now, let's define the "updateTimer()" function that you specified just now. The function needs to do the following:

1. Update the value of the "timeLeft" text field with the global variable "gTimeLeft".

2. Check if the "gTimeLeft" is less than or equal to 0, and if so, display a message that the player lost. Also, stop the timer, by calling "clearInterval(timer);".

3. Otherwise, subtract 1 from gTimeLeft.

4.5) Adjust the foundItem function

In the definition of the "foundItem()" function, add a line to stop the timer, when the player has won.

5. Extra Task for Graduate Students

In the current implementation, the game starts as soon as the page is loaded to the Web browser. This is not an ideal, because the player does not have a choice not to play the game, and the clock starts automatically, making the player a looser by default, if she/he chose not to play.

For an extra task, implement a "start" button, that start the timer and display the image. This changes the flow of the game as follows:

The browser loads the page.
→ Display the question, the list of items, the time remained, and the start button.
→ The player clicks the start button.
→ The timer starts and the picture is displayed
The rest of the game remains the same.

In order to change the image from JavaScript. Name the <img> element (say, "myImage") and initially specify an empty image file. (You can create one with Photoshop, or download http://kinoue.mysite.syr.edu/IST400-600/empty.jpg.)

Then you can change the image by assigning a value, like the following:

```
var image = document.getElementsByName("myImage");
image[0].src = "MessyRoom.jpg";
```

This is a technique using the DOM (Document Object Model), which we will learn next week. ☺

Submitting to Discussion Board

In order to handle multiple files (including images), from this lab, we will use the Web hosting service that SU provides, called MySite. The Instruction of MySite is found at http://its.syr.edu/mysite/.

Once you setup your MySite account, you will be able to copy your lab assignment to the MySite admin folder, and it will be accessible from the Web.

You will only need to submit the URL to page you created to the discussion board on the ILMS. For example, I would submit:

http://kinoue.mysite.syr.edu/IST400-600/MessyRoom.html

Grading Criteria

The lab assignment must be submitted by Tuesday Feb. 18rd midnight. The grading will be based on the following criteria:

- Game presentation: 2 points
- Logic: 2 points
- Map Syntax: 2 points
- JavaScript Syntax: 2 points
- HTML syntax: 2 points

Late submission will be accepted within one week from the original due date, with 2 points deduction. 2 points will be deducted for improper submissions (e.g. not using MySite) or for not submitting the peer survey as well.