بسم الله الرحمن الرحيم

|  |  |
| --- | --- |
| Smart Online Photoshop | Final  Documentation  Students : **Hawa’ Osama**  **Hiba Ghannam**  Supervisor : Haya Sama’neh  2011-2012 |

An-Najah National University

Faculty of Engineering

Department of Computer Engineering



Contents

Table of Contents

[Abstract](#_Toc307905846) 4

[Introduction 5](#_Toc307905847)

[1.What is a smart Photo editor 6](#_Toc307905848)

[1.1. Idea 6](#_Toc307905849)

[1.2. Features 6](#_Toc307905850)

[1.3. Application](#_Toc307905852) 7

[1.4. Application Snapshot and view 8](#_Toc307905853)

[2.Development tools and programming languages](#_Toc307905854) 10

[2.1.What We Have used?](#_Toc307905855) 10

[2.2.Smart client and.net framework 10](#_Toc307905856)

[2.3.Microsoft visual studio 2010 11](#_Toc307905857)

2.4 [C# Programming Language](#_Toc307905858) 12

2.5 Microsoft Silverlight  [12](#_Toc307905861)

2.6 Microsoft Expression blend  [13](#_Toc307905861)

2.7 XAML  [13](#_Toc307905861)

2.8DirectX [13](#_Toc307905861)

2.8High Level Shading Language  [14](#_Toc307905861)

3.Functionality  [16](#_Toc307905861)

3.1 Menu Bar  [16](#_Toc307905861)

3.2.Tool Box  [23](#_Toc307905861)

3.3.Advance Brushes [35](#_Toc307905861)

3.4.Scenario [37](#_Toc307905861)

4.Problems and Development40

5. Conclusion [42](#_Toc307905861)

6. References43

Appendix44

Abstract

In this project we devoted our work to develop an online web application for graphics and image editing. This “Smart Online Photoshop” is a smart client, meaning that it provides a good performance without needing to contact the server. “Smart” also means that you can continue working in the event of an interruption of the internet service, so it is an offline application. For application functionality you can choose your photo and start editing it using filters, crop, add shapes, change colors, add backgrounds and duplicate any element you want by copying and pasting that selected element. Also, you can draw any thing you need, fill colors, and add different brushes. When using the application you will see many different and interesting tools and you can save anything you want with any image extension.

Introduction

This documentation is aimed at equipping you to have the technical understanding needed for our project .This will enable you to become familiar with the idea of the project, as well as what we envision as its possible future development.

In chapter 1, we will introduce the project idea. We will explain why we choose a web- based application, why we choose our application to be smart and why it is a photo editor .Also chapter 1 will explain the features and possible users of our application. After that, we provided snapshots for the main parts of the project, which will be explained in more details in later chapters.

Chapter 2 will give you explanatory definitions about the development tools and programming languages that we use in our project. This chapter will talk about Microsoft Silverlight, Microsoft Expression Blend, C#, high level shader languages, DirectX and Shazzam – Shader Editor.

Chapter 3 will explain the functionality of the application by snapshots, for examples it will show drawing tools, brushes, selection tools and image processing functionality like filters .We will talk about algorithms we used or programmed.

In chapter 4 we will discuss the problems we faced and solutions that we reached. Also chapter 4 will explain any possible future development for our projects and finally the conclusions.

This documentation is provided with an appendix to explain some definitions that are used through this document.

Finally, the references of all researches of the project are followed to our documentation

1. What is smart Photo Editor

**1.1Idea**

These days we notice that businesses try to develop web applications to replace their old desktop applications, for many reasons, first the information in web application is accessible to a wide audience anywhere in the world, second it is available twenty four hours a day, seven days a week, and the updates can be made quickly and easily, finally no special configuration or changes are need on user's PCs. But user productivity of the web application has decreased. This is because web-based user interfaces base on server-side HTML generation are typically not as responsive, have fewer hot keys, require more use of the mouse and are unreliable when handling large files. Therefore we try to replace Photoshop Program with an online Photoshop website. However, we try to avoid the disadvantages of previous web applications and take the advantage of the web application, by using a Smart Client applications approach that bridge the gap between web applications and desktop applications. They provide the benefits of a web application (such as leveraging the internet and offering remote access to data) while still providing the snappy look and feel inherent to desktop applications.

**1.2Features**

1. has the look and feel of Photoshop Program.

2. Does not require installation

3. Utilizes Local Resources

Our application takes the advantage of local user CPU, GPU, local memory and disk. This is because it runs on the client side, so when the user applies any filter to the image, the filter code runs on the local user GPU, not on server GPU. The user only connects to the server to fetch the page, and all functionality is done using the local user recourses.

4.Offline Capability

Because our Application runs on the local machine, one of the key benefits that smart our application offers is that it can function even when the user is not connected. For applications running in occasional or intermittent connectivity situations, such as those used by traveling workers or even those running on laptops, tablets, PDA's, and so on, where connectivity cannot be guaranteed at all times, being able to work while disconnected is essential.

5. Our smart Online Photoshop software makes it easier than ever to edit your photographs wherever you are.

6. Because our project runs inside a web browser, it is not dependent on the operating system your computer uses. Our project can run on Mac OS, Linux, Windows, and any other operating system you care to throw at it. This also means that the only thing you need to use our project is any Internet connected computer.

7. Simplicity

The user interface of our project is actually very simple and extremely easy to understand. It also enables you to quickly and easily adjust the colors; Images can also be flipped or rotated.

**1.3 Application**

1. Create professional, beautiful graphic works

2. Edit image and apply multiple filter to it

3. Easy to touch photos up and make them look much better. This is made possible without making the photographs look unnatural. Red eyes, can be removed with a few clicks of the mouse

**1.4 Application snapshot and view**

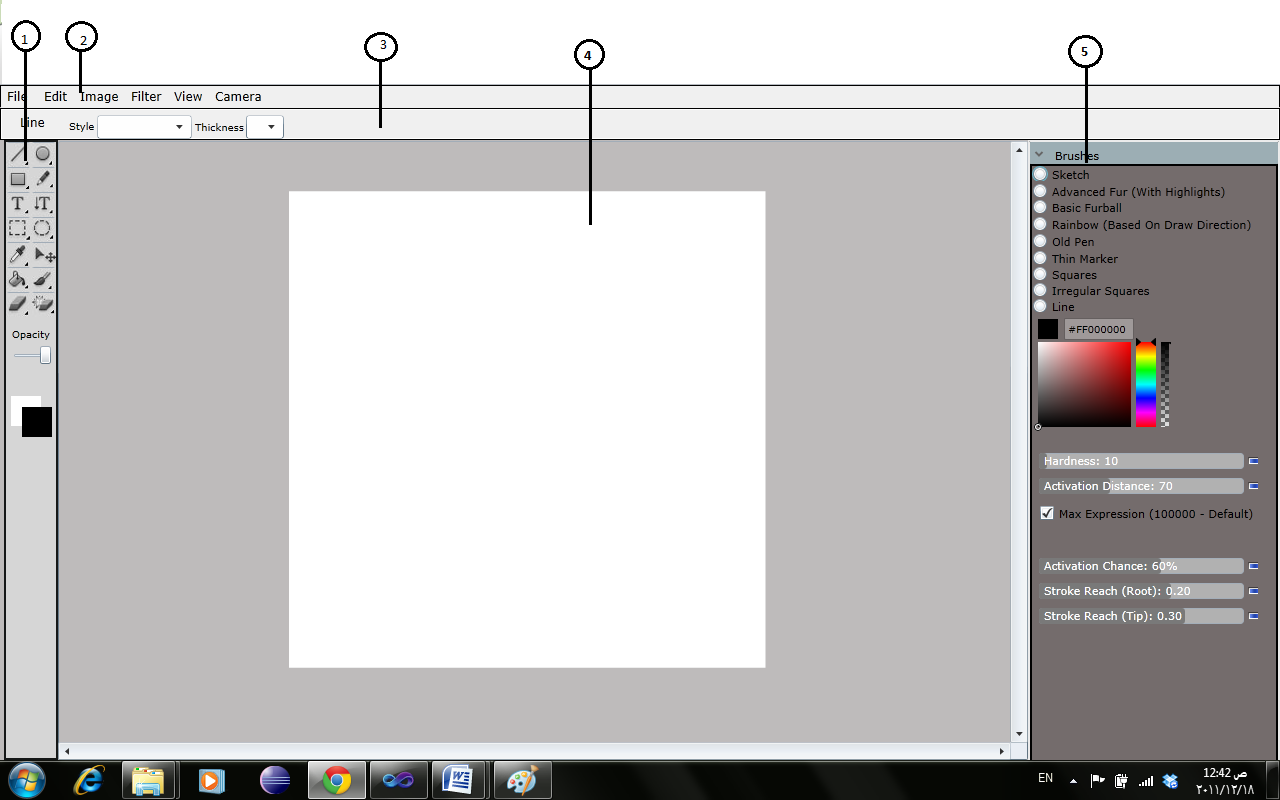


Figure 1.4.1:Smart Online Photoshop

As you see above our project consist of the many Region

* Region 1: it is the tool box which used to draw different shape like rectangle, line, and it also contains brushes and color picker.
* Region 2:it is the menu bar than that contain any menus like File,Edit,Filter,…etc.
* Region 3:this region is change depending on the selected item in the toolbox, for example if the user press the line the line control displayed in this region which allow the user to control the property of line like thickness and Style in figure 1 the user select the line.
* Region 4: is the Cnavas which is the area where the user open image draw shapes ….Etc.
* Region 5:is the Advance Brush which allow user to draw different nice brush

Chapter 2: Development Tools and

Programming Languages

**2.1: What we have used?**

Before you can start looking at the future, it is good to understand the current suite of web development and deployment technologies that are available to you. For us , when we choose the idea we decided to develop our application as a smart client , so we choose .NET frame work and the following programming languages and development tools that used in web development for .NET frame work.

* Microsoft Visual Studio 2010
* C#
* Microsoft Silverlight
* Microsoft Expression Blend
* XAML
* High Level shading Language
* DirectX
* Shazzam - Shader Editor

**Later there is an explanation of why we choose each one.**

**2.2: Smart Clients and the .NET Framework**

The .NET Framework has very effectively solved the problem of version conflicts between assemblies shared by more than one application. A .NET application or assembly has a strong coupling to the assemblies and components they depend on. Assemblies are annotated with meta-data, which specifies both their exact version and the versions of all dependent assemblies. Multiple versions of the same assembly can be installed side-by-side so that all applications can be run with the exact same version of the assembly that they were built and tested against.

This assembly binding mechanism also provides important securiurity safeguards. Assemblies can be cryptographically signed to prevent malicious code being run inadvertently, either through tampering or through luring attacks. In addition, .NET Code Access urity (CAS) allows assemblies to be granted specific permissions so that they can be used in semi-trusted scenarios.

These features effectively address application stability and predictability, but what about deployment? Assemblies do not need to be registered on the local machine so they can simply be copied to the machine before they are run. This greatly simplifies the deployment of the application. Alternatively, assemblies can be downloaded from a central server, either at install or at run time.

.NET also provides a deployment feature called No-Touch Deployment, which allows applications to be run on a URL. When the user clicks on the URL, the runtime automatically downloads the application, and all related assemblies, to a special download cache. This mechanism also checks for version updates so that the user never has to worry about whether they are running the latest version of the application or not.

Of course, the .NET Framework provides quite a few other features that affect how smart client applications can be designed and built. Of particular important are the Microsoft® .NET Windows Forms classes. These classes provide very rich user interface controls and give developers in all languages a common UI framework to work against, making development and testing much easier. The .NET Framework also provides many facilities for connecting to network services, such as SOAP-based Web services, and provides comprehensive XML support.

**2.3: Microsoft visual studio 2010**

Since we choose to work with .NET frame work, then we should work with the visual studio and the last version is 2010.

**2.4 : C# Programming Language**

We choose the c# as the main background programming language, because we have a good knowledge in it and it is available for us.

**2.5: Microsoft Silverlight**

Since our project is a smart client we find that Silverlight is a powerful development platform for creating and interactive user experiences for Web. Silverlight has a lot of features that required in our project :

* smart client features are provided by Silverlight like Automatic/intelligent installation and update broad client platform/device reach and use of local resources
* Text/Image Improvements
* Pixel Effects
* Bitmap API
* GPU Acceleration
* Create Brushes

**2.6: Microsoft Expression blend**

Expression Blend, Visual Studio, Silverlight and .NET provide the most compelling and seamless design and development workflow on the market today. Rapidly iterate on both the user experience and core architecture, evolving your ideas quickly from initial prototype through to completed project.

**Utilizing industry proven technologies Silverlight, WPF, XAML, C# and VB,** Expression Blend enables you to deliver applications that are stable, scalable, accessible and reliable, while maintaining optimum performance. The strengths of great user experience and robust core architecture are brought together with the combined power of Expression Blend and Visual Studio, enabling the accelerated delivery of applications through XAML and the Microsoft platform .**So we use expression blend behind Silverlight and visual studio for our design**.

**2.6: XAML**

Extensible Application Markup Language, or XAML (pronounced "zammel"), is an XML-based markup language developed by Microsoft. XAML is the language behind the visual presentation of an application that you develop in Microsoft Expression Blend, just as HTML is the language behind the visual presentation of a Web page. Creating an application in Expression Blend means writing XAML code, either by hand or visually by working in the Design view of Expression Blend.

The XAML for any given document in Expression Blend is stored in a .xaml file. If there is underlying code for your XAML document, that code is stored in a file of the same name, with the additional extension of .cs or .vb. For example, if your XAML document is named Window1.xaml, the code-behind file will be called Window1.xaml.cs if the programming language of the code is C# and so for our project the file is .xaml.cs.

**2.7: DirectX**

Windows supports DirectX 8.0, which enhances the multimedia capabilities of your computer. DirectX provides access to the capabilities of your display and audio cards, which enables programs to provide realistic three-dimensional (3-D) graphics and immersive music and audio effects. DirectX is a set of low-level Application Programming Interfaces (APIs) that provides Windows programs with high-performance hardware-accelerated multimedia support.

DirectX enables the program to easily determine the hardware capabilities of your computer, and then sets the program parameters to match. This allows multimedia software programs to run on any Windows-based computer with DirectX compatible hardware and drivers and ensures that the multimedia programs take full advantage of high-performance hardware.

DirectX contains a set of APIs that provide access to the advanced features of high-performance hardware, such as 3-D graphics acceleration chips and sound cards. These APIs control low-level functions, including two-dimensional (2-D) graphics acceleration; support for input devices such as joysticks, keyboards, and mice; and control of sound mixing and sound output.

In our project we use Directx with High Level Shader Language to build image processing functions like filters which we will explain in the next chapter.

**2.8: High Level Shading Language**

One of the most empowering new components of DirectX 9 is the High Level Shading Language (HLSL). Using this standard high level language, shader writers are able to think at the algorithm level while implementing shaders, rather than worry about meddlesome hardware. The HLSL also has all of the usual advantages of a high level language such as easy code reuse, improved readability and optimizations provided by the compiler. HLSL was defined by Microsoft and introduced with DirectX 9 in 2002. In terms of its syntax and functionality, HLSL is much closer to the OpenGL Shading Language than either RenderMan or ISL. HLSL supports the paradigm of programmability at the vertex level and at the fragment level just as in the OpenGL Shading Language. An HLSL vertex shader corresponds to an OpenGL vertex shader, and an HLSL pixel shader corresponds to an OpenGL fragment shader.

One of the main differences between the OpenGL Shading Language and HLSL is in the execution environment .The HLSL compiler is really a translator that lives outside DirectX in the sense that HLSL programs are never sent directly to the DirectX 9 API for execution. Instead, the HLSL compiler translates HLSL source into assembly-level source or binary programs called vertex shaders and pixel shaders (in Microsoft DirectX parlance). Various levels of functionality have been defined for these assembly level shaders, and they are differentiated by a version number (e.g., Vertex Shader 1.0, 2.0, 3.0; Pixel Shader 1.1, 1.4, 2.0, 3.0).

**2.9: Shazzam Shader Editor**

The goal of Shazzam is to make it simple to edit and test WPF Pixel Shader Effects. Shazzam compiles HLSL code, auto- generates C#/VB classes and creates a testing page for each effect. The test page allow you to visualize your effect and apply animations to any of the effect properties.

**3.1 Menu Bar**

The first toolbar in our project is the Menu Bar, our Menu Bar consist of File, Edit, Image, Filter, View, Menu I will explain each menu and its item next.

**3.1.1 File Menu**

Using this menu you can open image in different ways, For example you can browse the image from the user computer or you can open it from web pages using the image URL, also you can save the image

1. New Image

This function is used to add canvas to the project; this canvas act as a container to element such as Rectangle, Ellips, line ……etc that added to it dynamically, before you open this canvas you should specify the canvas width and height to determine the canvas size

2. Open Image

To open the image we construct a WriteableBitmap then to display this image in the canvas we use the WriteableBitmapas the source for an imaging control, in our project you can open the image in different way:

* Open Image, you can Browse the image from your computer.
* Open Image URL, this feature make opening image from Webpage easier, instead of save the image from the webpage to your computer then open it, you can basically copy the image URL and paste it to the project then the image will directly open.
* Drag and Drop: the user can drag the image from any place in his computer and drop it directly inside out project.

1. Save the mage

Our project save the image in the user computer in three format Bmp, Jpeg, and Png. we do this using encoder of each format of image.

**3.1.2 Image Menu**

1. Image Size

This function is used to control the height and width of image, by filling the child window which prompted as Image Size clicked.

1. Canvas Size

Like the image the user can control the size of the canvas if the canvas size is smaller than the image size then the image size become as the size of the canvas else the image size stay the same and canvas resized.

1. Rotate Canvas 180

The user rotate the canvas by 180° Clock Wise each time this menu item is clicked, the rotation is accumulated I mean when you click this menu tow time the canvas will rotate 360°.To do this we use the Rotate Transform .

1. Rotate 90CW

The user rotate the image 90°

1. Rotate 90CCW

This function used to rotate the image 90° counter clock wise

1. Flip Canvas Vertical

This function used to reflect the canvas about the X- axis; we use the Scale Transform to implement this using Scale Transform.

1. Flip Canvas Horizontal

This function is used to reflect the canvas in the Y-axis.

**3.1.3 Filter Menu**

We build filter in our project using Microsoft's [High-Level Shading Language](http://msdn.microsoft.com/en-us/library/bb509561(VS.85).aspx) (HLSL) ,when we write filter in this language ,DirectX SDK(Fxc.exe compiler) used to compile the generated file with fx extension, into ps file which included as a resource file in our project ,or we can use tools called t [Shazzam](http://wpfwonderland.wordpress.com/2008/10/08/shazzam-wpf-pixel-shader-effect-testing-tool-now-available/" \t "_blank" \o "Shazzam)  and use it to compile PS files. An added bonus with Shazzam is that you can test and debug the filter effects before compiling them. In our case we use Shazzam for each filter we have a class from providing a default constructor that initializes the filter property with a reference to ps file.

1. Directional Blur

This filter is used to reduce the detail of the image, this filter has tow parameter first one is the Angle which defines the direction of the blur; the second one is the Blue amount which used to identify the scale amount of the blur.

1. Pixelate

This filter is used to Alter the number of pixel in an image, we can change the number of horizontal and vertical pixel block.

1. Old Picture

This filter used to convert the image to black and white color.

1. LightStreek

This filter is used to control the brightness of an image it’s used to decrease/increase the amount of brightness.

5. SketchPencilStrock

This filter used to convert the Picture as Pencil Sketch image.

6. WaveWarper

This filter used to apply wave pattern to the input image we use Timmy Kokke Algorithm to implement this function.

7. Red eye Removal

If the user have perfect picture, expect it’s been ruined by those unsightly, glowing red eyes from camera flash, it’s easy to remove this red eyes without effecting the other part of image, only thing you have to do is to select the area around the red eyes, then apply the filter, we use [Jeff Promise’s](http://www.wintellect.com/CS/blogs/jprosise/default.aspx) Algorithm to do that filter.

8. Magnify

A filter that magnifies a circular region, you can control the radios of this region, and you can change it location in x and y direction.

9. Remove filter

If you make any filter to image you can remove it and return to the original picture.

For the entire above filter before the user applies the filter to image we let him to see the effect of filter, in child window prompted when he clicked in the filter.

**3.1.4View Menu**

This menu allows the user to hide or view the toolbox in our website.

1. Toolbox

Hide or view the Tool bar

2. Advance Brush bar

Hide or view the advanced brush bar.

**3.1.5Edit menu**

1. Undo

We all make mistakes. Having the ability to undo (and then possibly redo) mistakes in any application is extremely convenient because saves us all time from having to start over. In our project for each action we do we save the previous and the current state in a lists for example to draw an element in the canvas, we add the element in the list and the undo action will delete this element from the canvas, the redo we add it again to the canvas.

2. Cut

In our project the user can cut any portion of the image with rectangular or circular form, before do that function the user select the area he want to cut then our function copy the pixel of the selected area and store it in the memory as a Writable Bitmap, after that the function modify the selected area and make it color white.

3. Copy

This faction is similar to cut function ,as I say the function store the pixel of the selected area in the memory, but this time it doesn’t modify the selected area its stay the same.

4. Paste

This function take the pixel array (Writable Bitmap) from memory and make it as a source for image add to the Canvas ,the user can move this image to any part of the canvas.

5. Select all

This function used to select all the area in the working Canvas, so the user can copy or cut the entire part of the Canvas, I store the top left corner point and I get the width and the height of the canvas, if the user choose copy, then I start to copy pixels, row by row until I reach the last row in the canvas.

6. Deselect all

If the user select all area then he like to deselect it he can do it simply using this command.

Important note

The mouse is not the only way to interact with our website, the user can use the keyboard to do many function, to simplify it for the user we use common short keys to handle common command for example, the copy is done when the user press"Ctrl+C" we write the short key name beside the command to let the user know when to use them.

**3.2: Tool Box**

Smart Online Photoshop has a main part which is tool box as shown below in figure 3.2.1.This tool box contains 17 tool for drawing , selection , filling , movement , remove elements , erasing and other many interested elements .We will show them in later sections with more details. All is designed through Silverlight and Expression Blend with XAML .

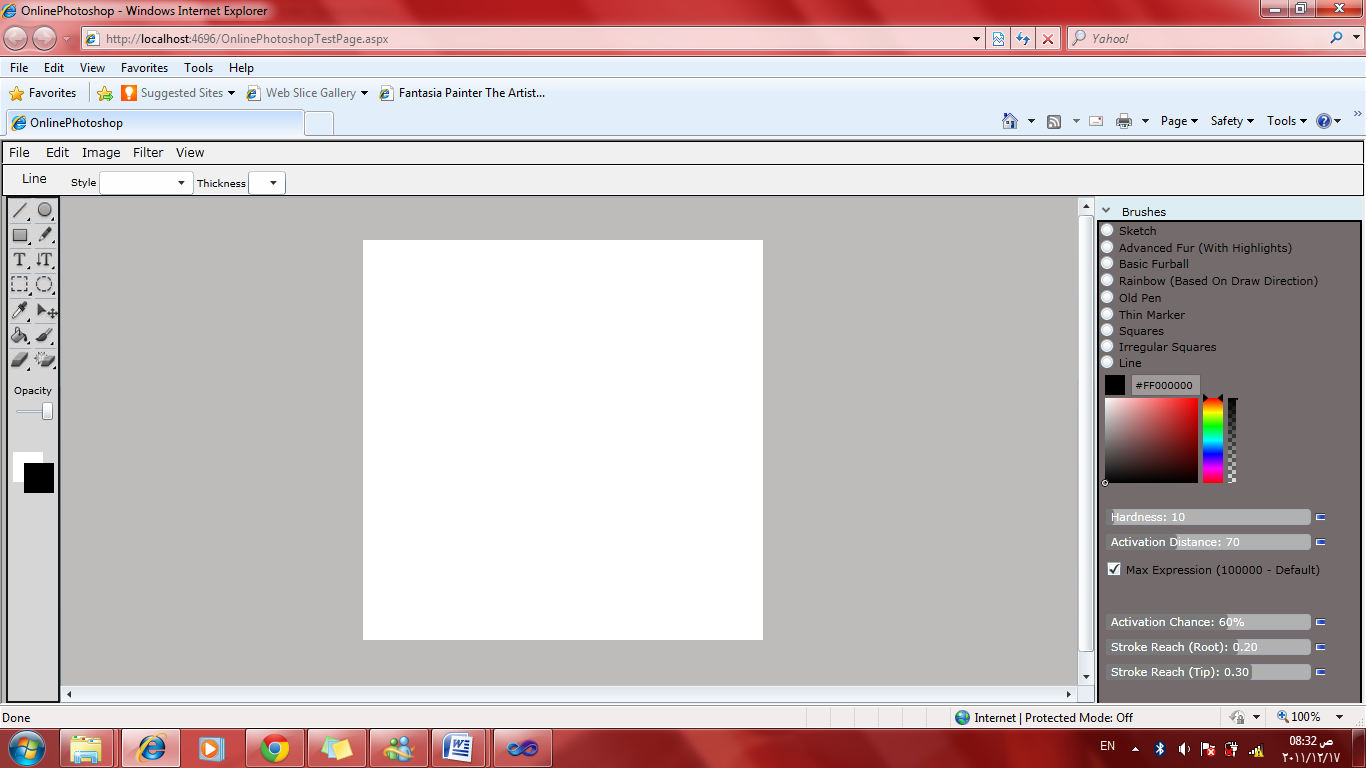
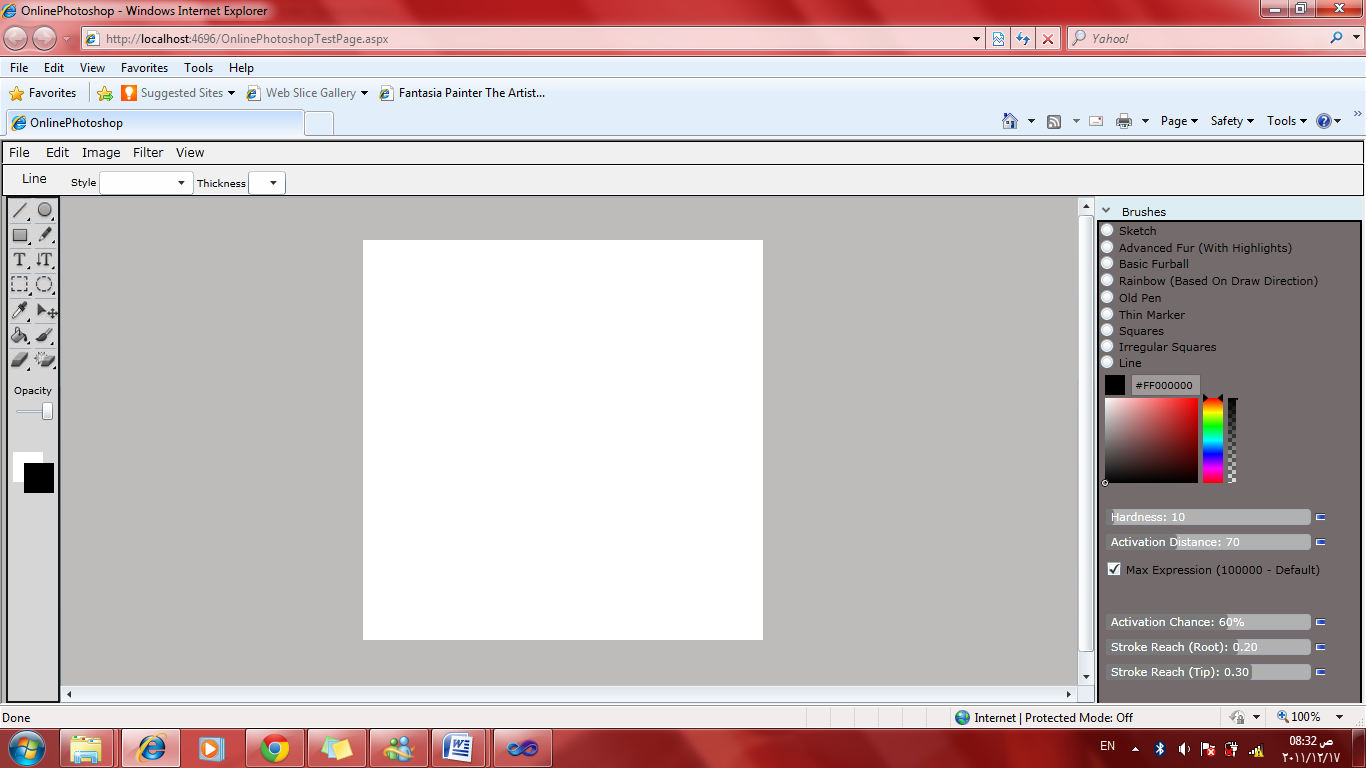
****

Figure 3.2.1 Tool Box

Tool Box

**3.2.1: Drawing Tools**

In this section there is a description for all drawing tools in tool box. As shown in Figure 3.2.1.1 there is four Drawing Tools. Through coding we have used c#.

****

Line

Rectangle

Ellipse

Pencil

Figure 3.2.1.1 Drawing Tools

* **Line Tool**: line tool allow you to draw any straight line with different properties like style and thickness. Element properties shown in properties bar when you click mouse left button on the tool, as shown in Figure 3.2.1.2 .Also the figure shows lines were drawn in different styles normal, dashed and dotted, also different thickness .

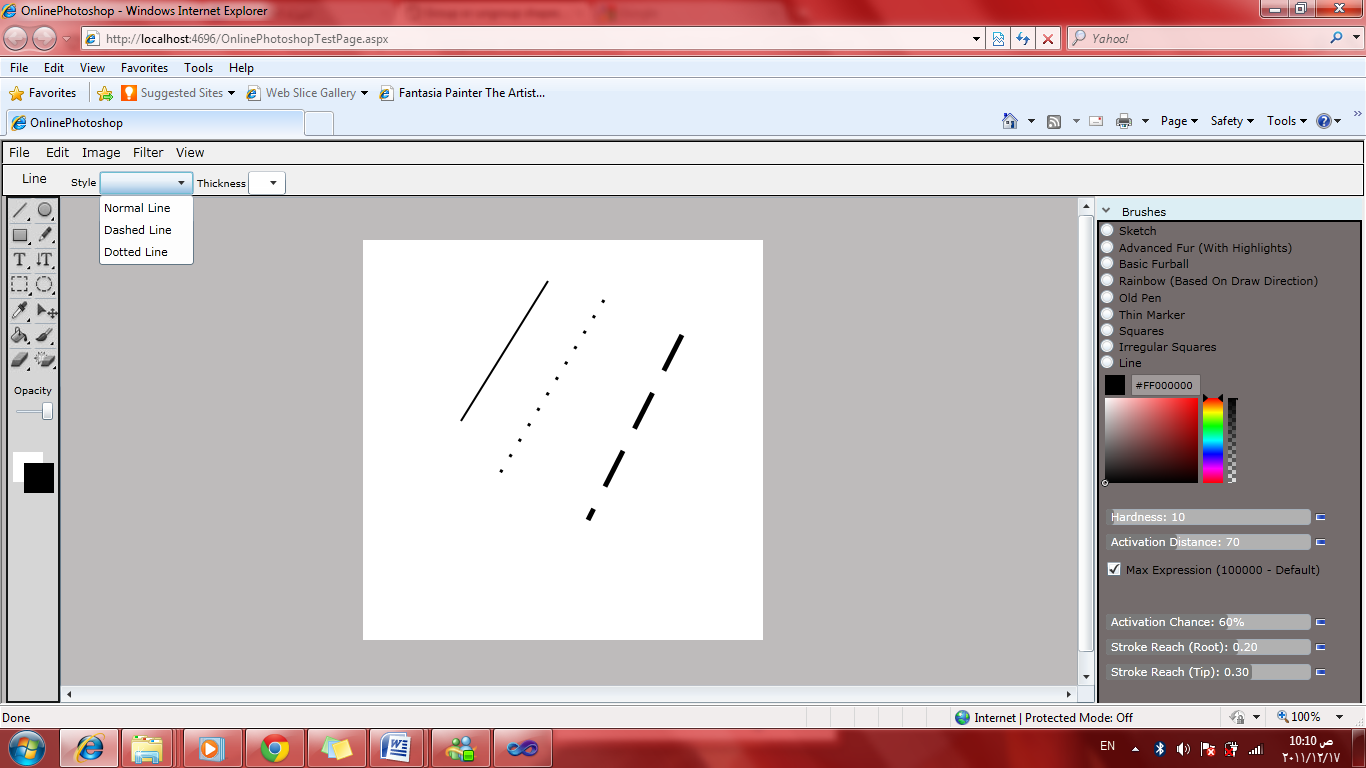


Figure 3.2.1.2 Line Tool

This figure shows different lines with different thickness and styles. Also line properties bar below menu bar. Each properties bar is a separate user control.

**Line coding:**

* For drawing any shape we used mouse events down, move and up.
* For line we declare global variables, origin point to get the location of mouse down click to start drawing. Also declaring current origin to keep the location of mouse move and then take the end point of the line, and then set the properties of line through line control which appear as line bar properties.
* **Ellipse Tool** :you can draw ellipses by this tool with any size you need , also you can change its size and style of its border line as shown in Figure 3.2.1.3

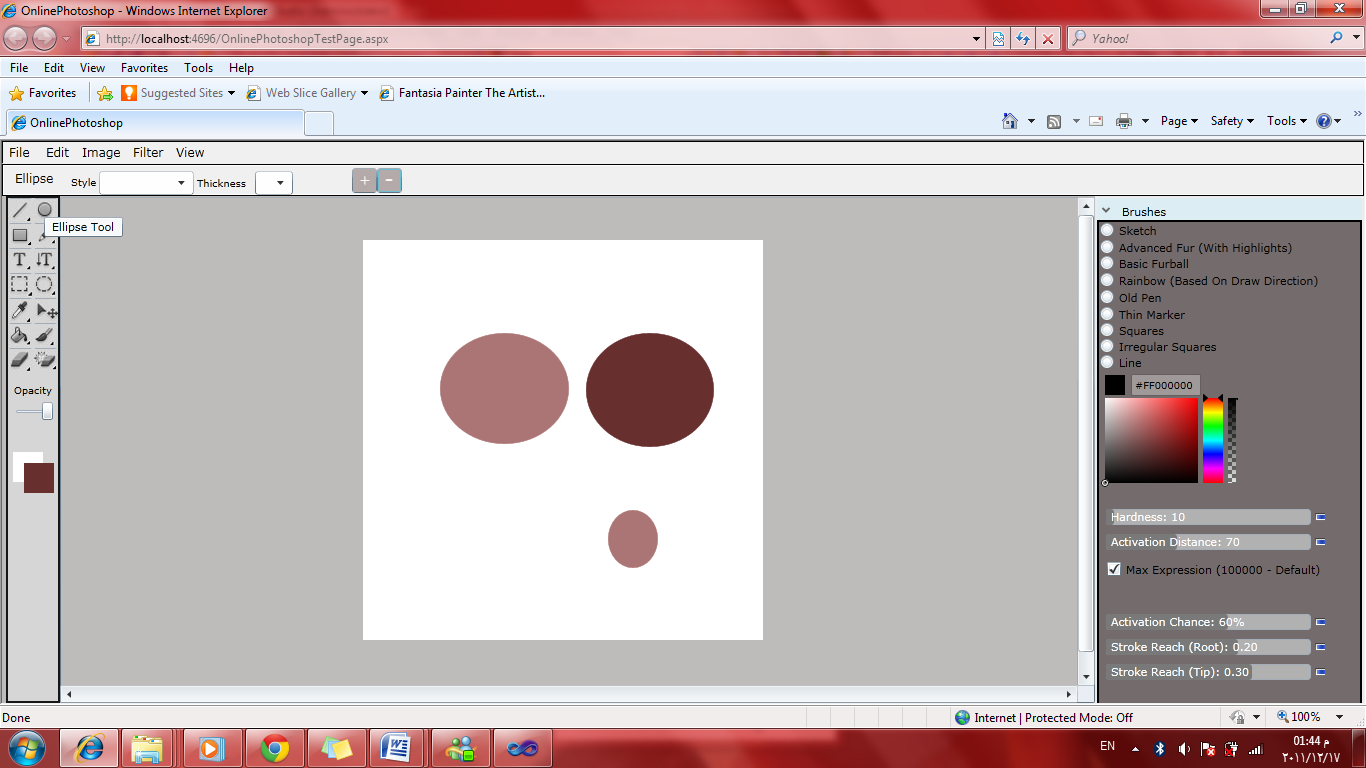


Figure 3.2.1.3 Ellipse Tool

The figure shows different ellipses with ellipse properties bar which contains two buttons for changing size of any ellipse in the canvas, just left mouse click on the ellipse and then change the size as you need.

**Ellipse coding:**

* For drawing ellipse we define also a global variable ellipse. Then using the origin and current points to get the position of mouse down, move and up.
* Take the difference between origin and current points to get the length of the diagonals then set the properties of the ellipse and add it to the canvas.
* For sizing the ellipse we use

ellip.MouseLeftButtonDown +=new MouseButtonEventHandler(FillEllipse\_MouseLeftButtonDown);

Which allow you to change the diagonals length by 1 unit at each click on + , \_ buttons on ellipse properties bar.

* **Rectangle Tool** :you can draw rectangles by this tool with any size as ellipse , also you can change its size and style of its border line as shown in Figure 3.2.1.4

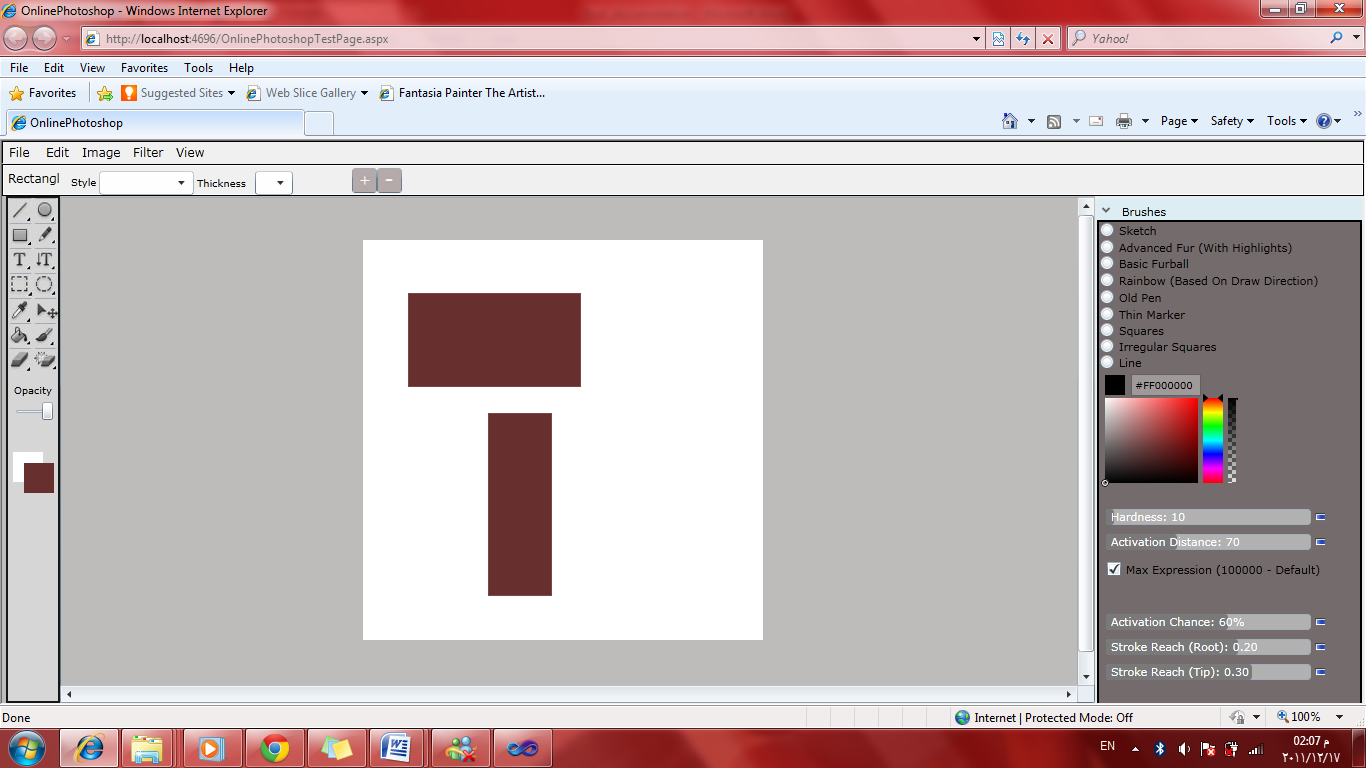


Figure 3.2.1.4 Rectangle Tool

The figure shows different rectangles with rectangle properties bar which contains two buttons for changing size of any rectangle in the canvas, just left mouse click on the rectangle and then change the size as you need.

**Rectangle coding:**

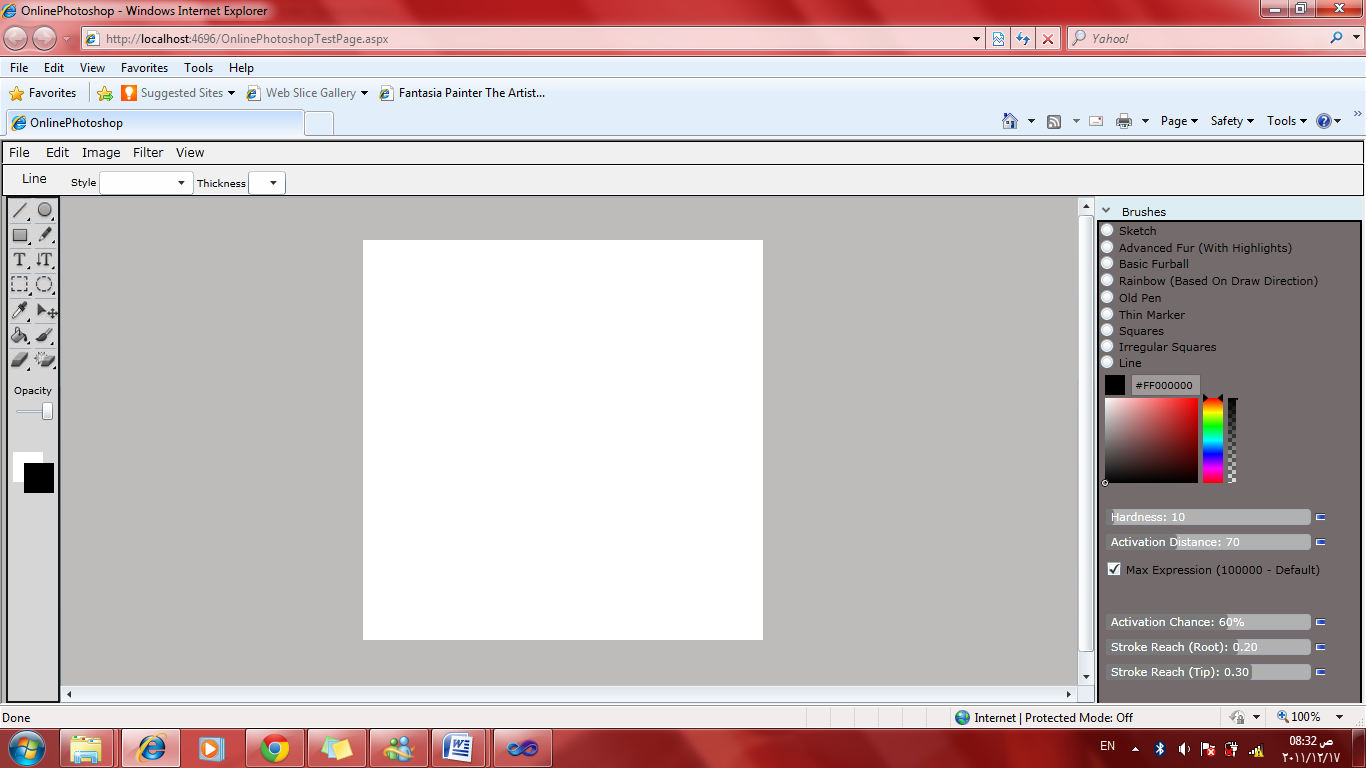
* The same algorithm for Ellipse Tool but the difference of points is calculated to find the width and height. Sizing property is the same in multilayer Photoshop.
* **Pencil Tool**: with this tool you can draw anything you want , so its free drawing tool.

**Pencil coding:**

* The same algorithm for Line Tool but we set the origin point as current point while the mouse move.

**3.2.2: Colors Tools**

Smart Online Photoshop has the main important color tools that enable you to set the color of all shapes and elements of your canvas. The programming language for each tool is c#. Figure 3.2.2.1 shows Colors Tools.

****

Eyedropper Tool

Fill Tool

Brush Tool

Opacity Slider

Foreground Color

Background Color

Figure 3.2.2.1 Colors Tools

* **Background Color :** this tool enable you to choose the back ground for the canvas you draw on, also later when we will talk about selection tools and we will we if we crop photo the background color will appear .when click on Background Color a color picker will appear as shown in Figure 3.2.2.2 and we coded it by c# with these features:
* Palette completely looks like Photoshop color palette.
* Fully Plug n Play Control.
* Event occurs when color changed by user with selected color as event arguments.

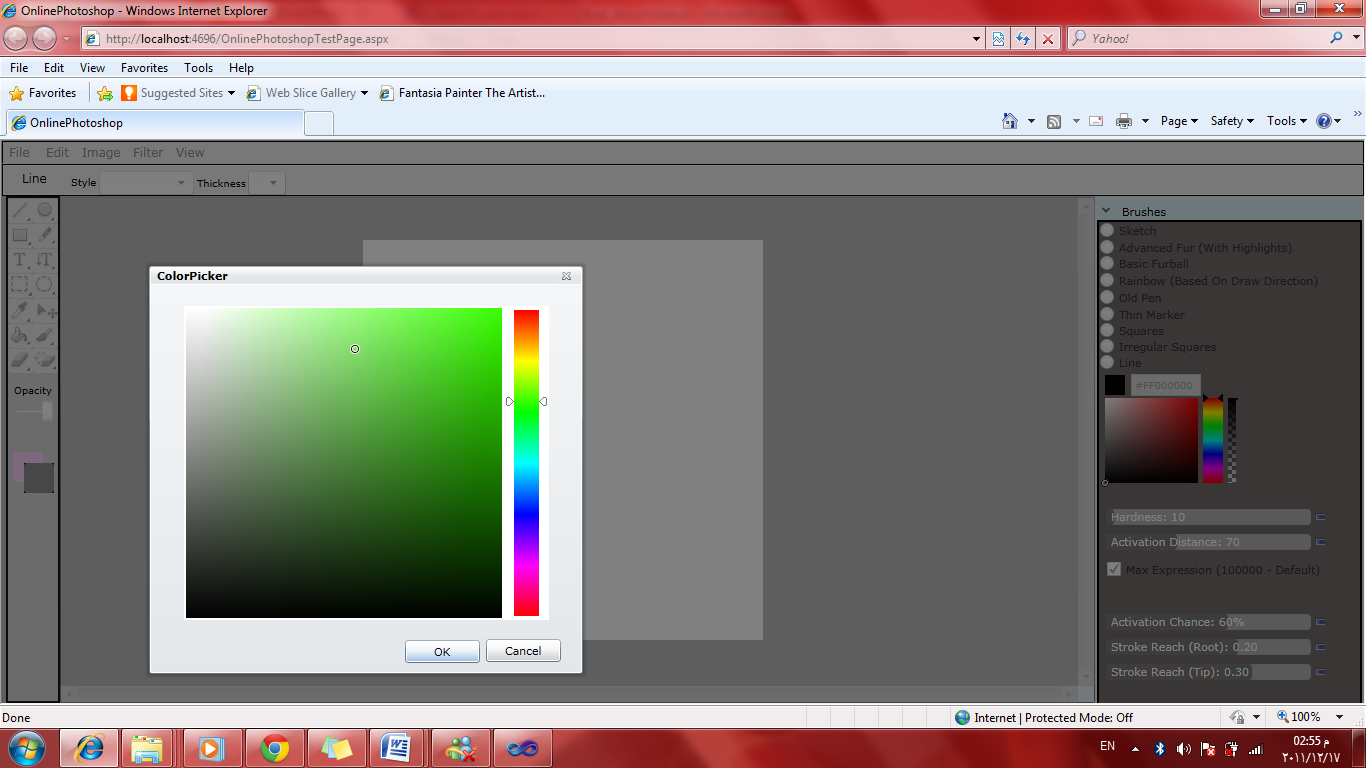


Figure 3.2.2.2 Color Picker

* **Foreground Color** : as Background Color , Foreground Color will provide the color picker as in Figure 3.2.2.2 .Fore ground Color is used as color for texts and shapes or anything will drawn over the canvas.
* **Opacity Slider :** this slider is designed in Silverlight with XAML and we give it two values ,minimum and maximum, then in c# we change the opacity of all elements on the canvas by move the slider through its values and elements opacity property.
* **Brush Tool:** as other tools we build our brush tool .Using this tool will enable you to paint any region or shape .This brush has a properties bar that has thickness and opacity properties of your brush.Figure 3.2.2.3 show the properties bar of Brush Tool.

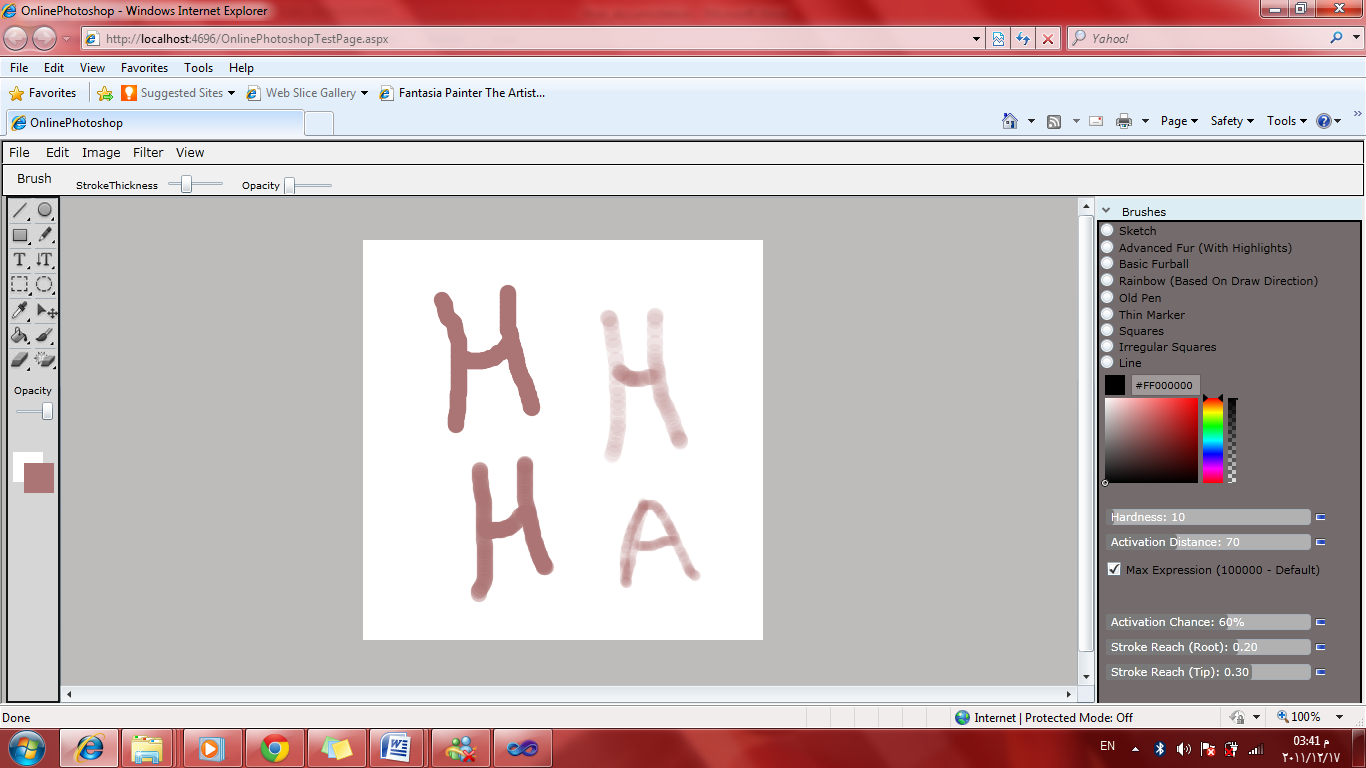
****

Figure 3.2.2.3 Brush Tool

Different thickness and opacity you can use through brush properties bar which contains thickness and opacity sliders.

**Brush coding:**

* The same algorithm of pencil is used to add a brush , but use these two properties of line :

line.StrokeStartLineCap = PenLineCap.Round;

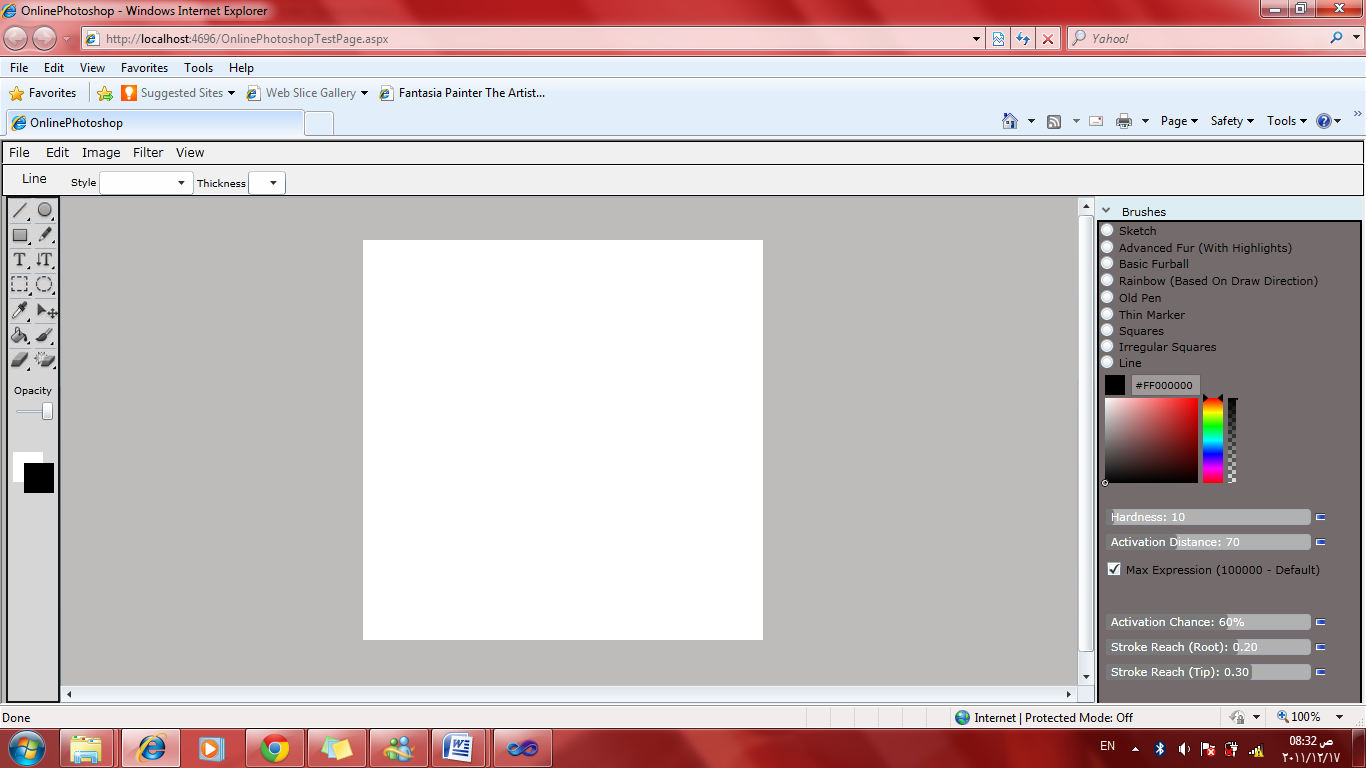
line.StrokeEndLineCap = PenLineCap.Round;

PenLineCap describes the shape at the end of a line or segment, while it’s flat for pencil tool.

* **Fill Tool:** this tool allow choosing any element on the canvas by mouse click and then fill the element like shapes or canvas background with foreground color which is set by color picker. This feature does the same thing that exists in multilayer Photoshop.
* **Eyedropper Tool:** you can select any pixel color and set the Foreground Color as that pixel, its important tool to process any image you want.

**3.2.3: Erasing Tools**

Smart Online Photoshop provides two tools to erase or remove elements on canvas and Figure 3.2.3.1 show them.



Eraser Tool

Erase Element

Figure 3.2.3.1 Erasing Tools

* **Eraser Tool:** eraser enables you to erase any aria by mouse move over that area canvas. Its algorithm is the same for Brush Tool but the difference is just the color of Eraser Tool is the same as Background instead of Foreground in Brush. Also you can change the thickness of eraser as your need. Figure 3.2.3.2 show the eraser with its property bar.

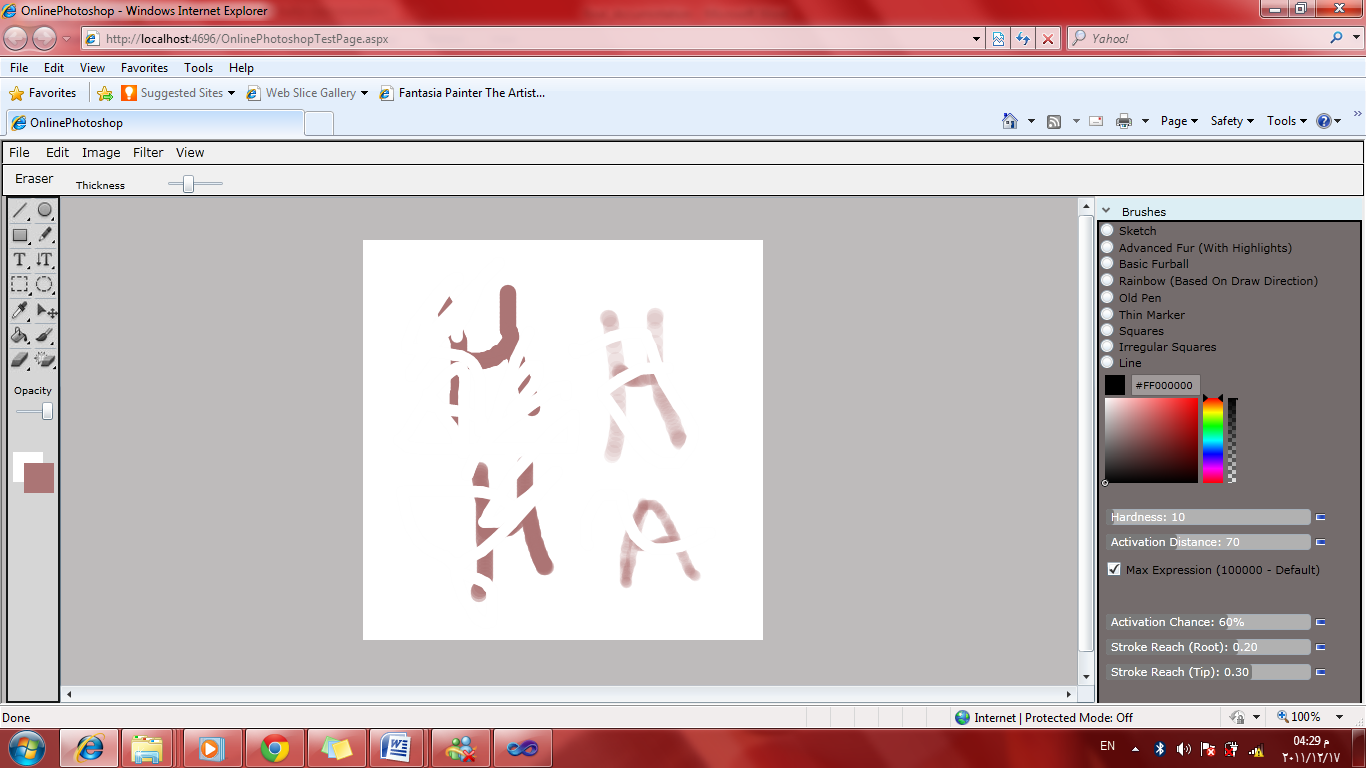


Figure 3.2.3.2 Eraser Tool

The figure show the eraser element with thickness selected by its property bar and it takes the background color to erase areas on canvas.

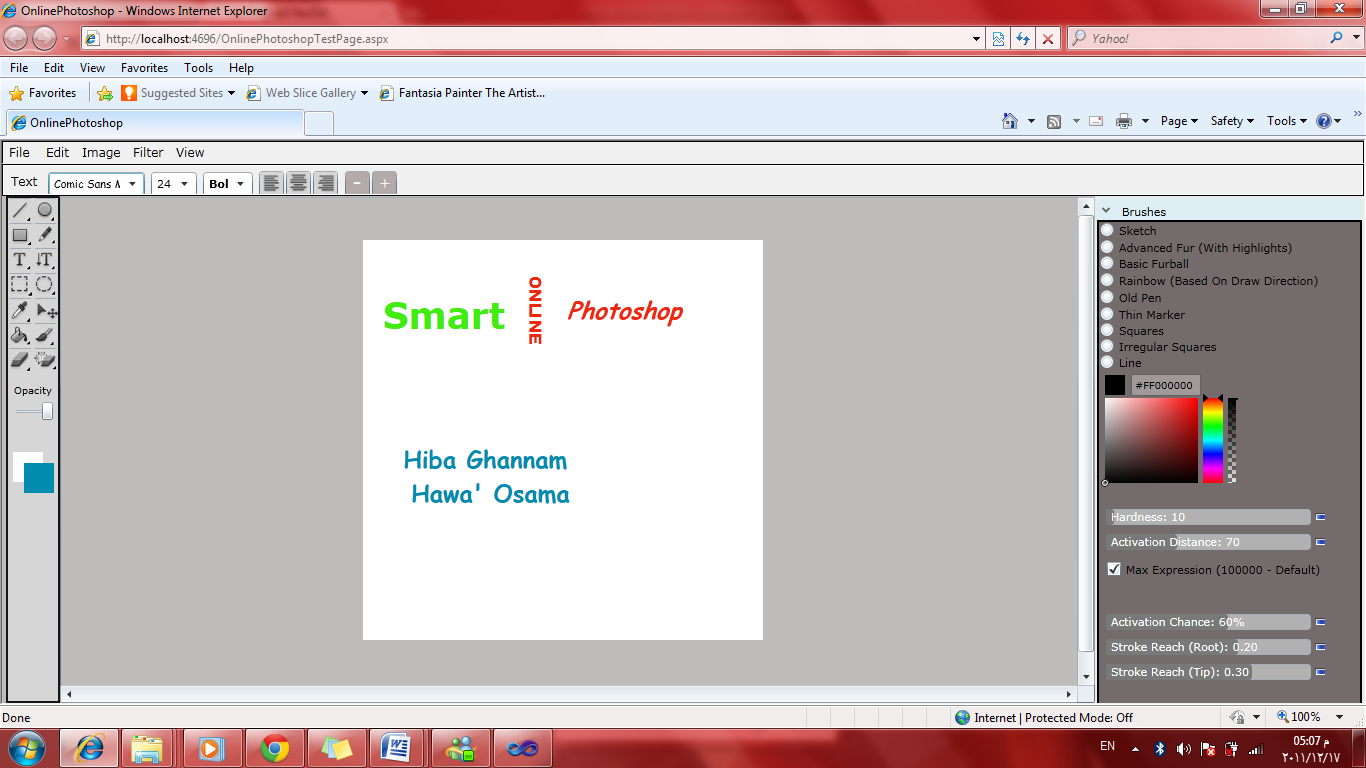
* **Erase Element:** this tool enable you to remove any element on canvas, just mouse click over the element after choose Erase Element tool. We implement this function

void Element\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e) and then FrameworkElement fEle = sender as FrameworkElement;

So we can take any element just by mouse click and then it will be removed as you remove layer in **multi layer Photoshop**.

**3.2.4: Text Tools**

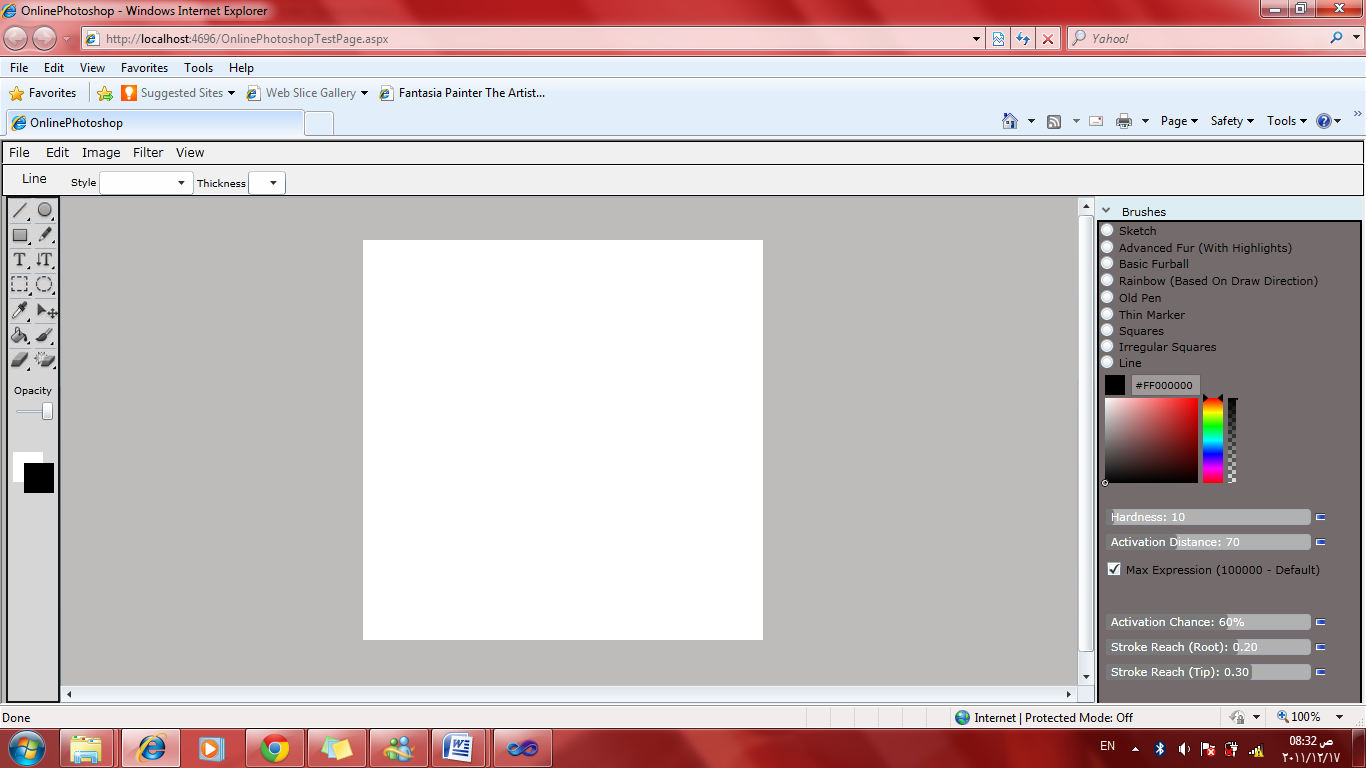
Smart Online Photoshop give two text boxes to write your text with its properties that selected from text bars ,one is horizontal and the other is vertical as shown in Figure 3.2.4.1. The color of text is selected by foreground color.



Horizontal Text

Vertical Text

Figure 3.2.4.1 Text Tools



**Text Tools coding**: for horizontal text just declare a global variable on c# and set its properties by its bar. But, for vertical text we use transformation methods that used to transform images, so we use rotate function to rotate each character .Also the removal of text box is possible using erase element tool, and then the properties of text on multilayer Photoshop are set here.

**3.2.5: Move Tool**

Our smart application give you a movement tool that enable you to move any element you want like text, photo, shape. Figure 3.2.5.1 show it.

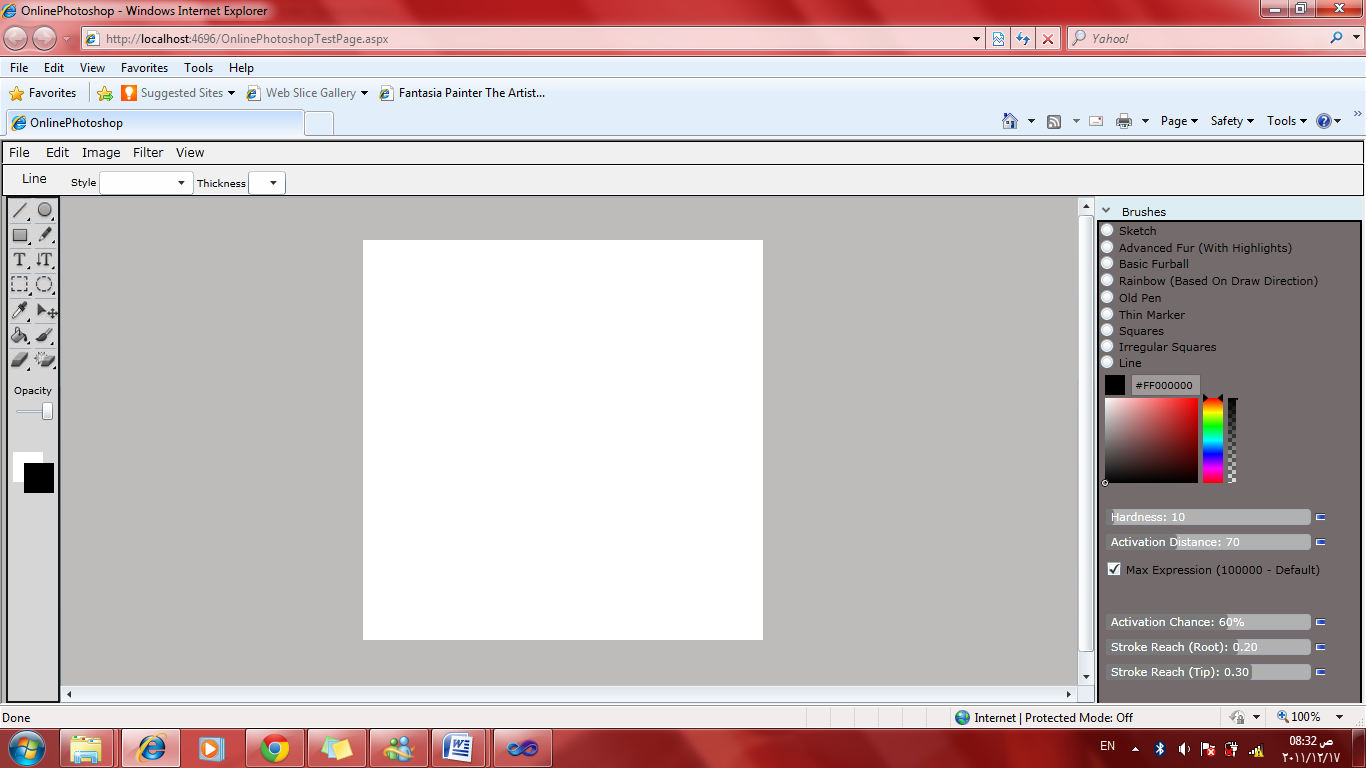
This tool is also a feature that exists in multilayer Photoshop. But we apply it to our single layer application to enable you to change the location of any element on the canvas by drag the element using mouse.

Move Tool coding: we implement three methods for this tool:

void Element\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

void Element\_MouseMove(object sender, MouseEventArgs e)

void Element\_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)

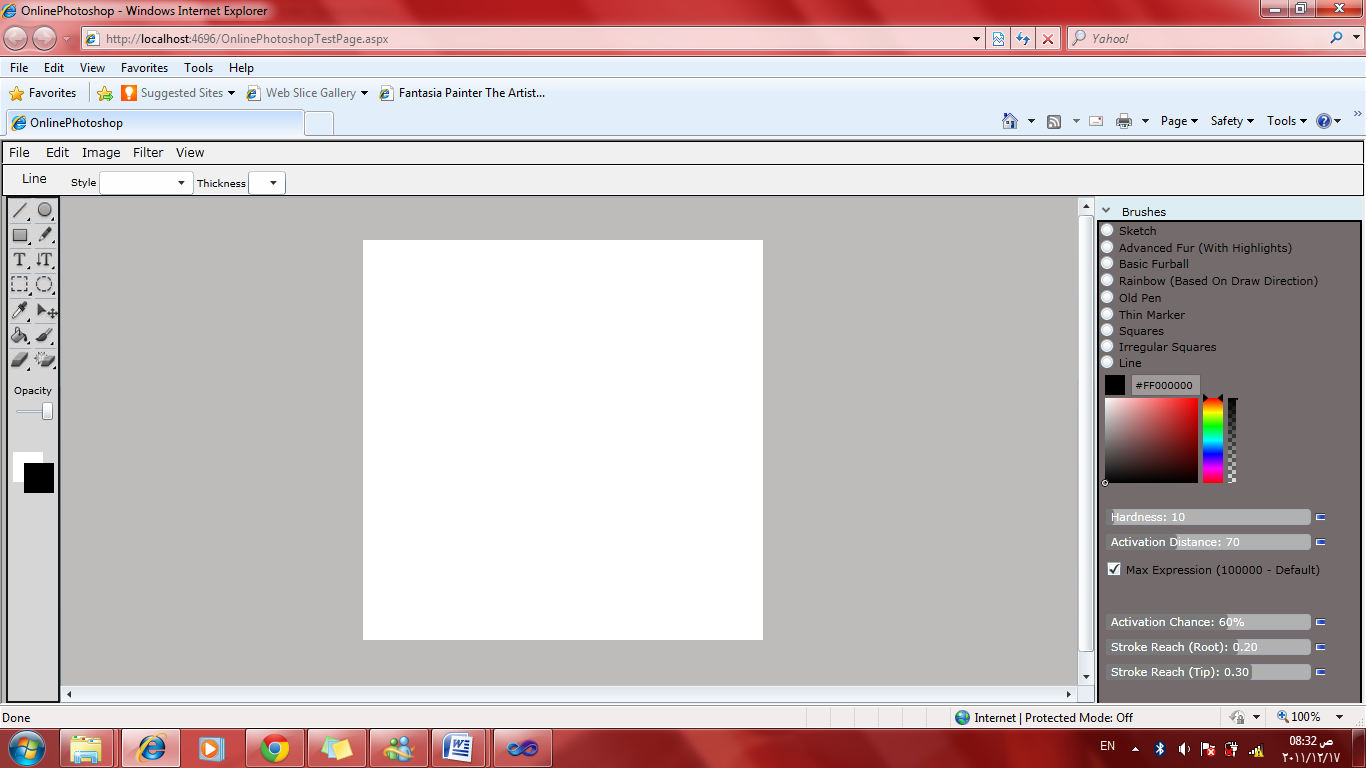


Move Tool

Figure 3.2.5.1

Move Tool

**3.2.6: Selection Tools**

 Our application has two selection tools, rectangle and elliptical. Surly you can cut, copy , paste or fill the selected part of your need . Figure 3.2.6.1 show these tools.

Rectangular selection

Elliptical selection

Figure 3.2.6.1 Selection Tools

**Selection Tools coding :** the algorithm is the same for drawing ellipse and rectangle but the work was with cut, copy, paste, filling and filter the selected part.

**3.3Advance Brushes**

Brushes are a great way to add that extra little embellishment to you’re the user images, we have very nice brushes in our project like:

1. Sketch Brush

As you see in Figure 3.3.1 this brush is used to draw sketch line the user can change the thickness of the brush and the color of the brush from the color picker.



Figure3.3.1: Sketch Brush

2. Basic fur ball

This brush is used to draw lines which are similar to fur as the previous brush the user can change it line and thickness.

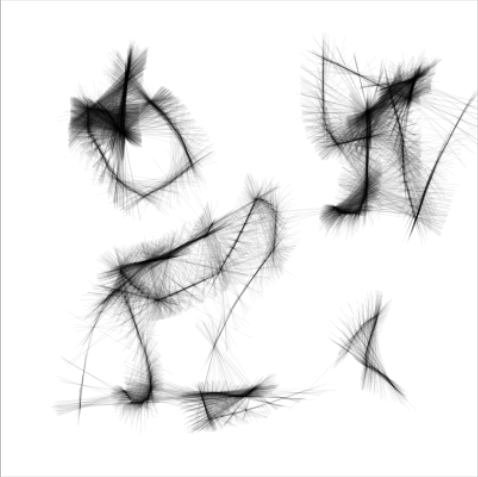


Figure3.3.2: Basic Fur Brush

2. Thin marker

This brush used to draw line with multiple colors as below.



Figure3.3.3: Thin Marker brush

4. Square brush

This brush is used to draw asset of rectangle in the canvas the size of this rectangle change according to the seed of the mouse when the speed increase the square size increase

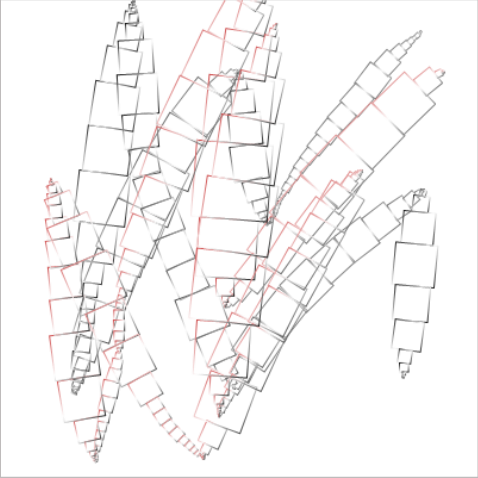


Figure3.3.4: Square Brush

Using this brush you can draw any thing you need like Figure 3.3.5



Figure 3.3.5: Brush effect

**3.4 Scenario**

****

Figure 3.4.1:UML Diagram

****

Figure 3.4.2: Sequence diagram

4. Problems and Development

4.1Problems

The first problem was to determine which programming language to choose. Because we have smart client application, and all of its functionality must be on the client side, we have many options, first PHP and AJAX, second ASP.Net and AJAX, finally Silverlight. After long study and research we see that Silverlight is the best choice for many reason, it is easy to design website that is similar to a desktop program. In addition all function implemented by Silverlight run on the client side not on the server meaning that it does not exhausts the server. Finally it improves graphic rendering because its uses the GPU of user computer.

The second problem is the design problem. During our study we learned to build conventional WebPages, this was the first time we designed a website that was similar to desktop programs, we tried as much as possible to make our project interface similar to Photoshop program, and we spent a lot of time to do this interface.

Finally, the biggest problem was the image processing, for us this was the first time editing images; through our research we found that Silverlight supports HLSL. Then we started writing filters using this language, compiling them, and then including the compiled file in our project.

4.2Future development

In the future we would like to make our website multi- touch website, This is possible because multi-touch capabilities are one of the most important features to [Silverlight](http://silverlight.net/) Users will no longer be limited by a simple keyboard and mouse; they simply interact with the website using their hand.

Another development to our project, is applying filters to videos and individual frames in the video. We also want to apply parallel programming, to improve the performance and speed of the website.

We hope to use cloud computing applications in our project. This gives the user a very large space to store images, and enables the user to work on them any time and any place.

Now we developing a tool that use the local camera of the user, in the website, allow the user to apply multiple filter to it, and finally save it as photo.

5. Conclusion

Our project is combination of Desktop application feature such as fast ,and using the local resource of the user computer ,and web based application because the project doesn’t need to buy the program, only computer connected to internet is needed to use the program, in our project we try as much as we can to do the Photoshop program functionallity, we do a lot but not all because we have small period of time to the graduation project and we have a lot thing to learn before we start, in the future we hope to include all the Photoshop feature in our project, also we hope to allow user to have large space for storing his image using Cloud computing feature

**6. References**

1. Lynda\_Silverlight.3.Essential.Training Videos.
2. Professional\_Silverlight\_4 Jason Beres
3. Expression.Blend.4 Andrew Troelsen
4. Pro.Silverlight.4.in.CSharp Matthew MacDonald
5. [jeff prosise's blog](http://www.wintellect.com/CS/blogs/jprosise/default.aspx)
6. [http://www.silverlightshow.net](http://www.silverlightshow.net/)
7. [http://www.silverlight.net](http://www.silverlight.net/)
8. [http://programmerpayback.com](http://programmerpayback.com/)
9. [http://www.neatware.com](http://www.neatware.com/)
10. [http://msdn.microsoft.com](http://msdn.microsoft.com/)
11. [http://ar.wikipedia.org](http://ar.wikipedia.org/)
12. http://blog.shazzam-tool.com
13. http://www.directxtutorial.com
14. http://www.wynapse.com
15. <http://expression.microsoft.com/en-us/cc184874>
16. http://www.c-sharpcorner.com/
17. http://www.neatware.com/lbstudio/web/hlsl.html

7. Appendix

1. Shader, in the field of [computer graphics](http://en.wikipedia.org/wiki/Computer_graphics), a shader is a [computer program](http://en.wikipedia.org/wiki/Computer_program) that is used primarily to calculate [rendering](http://en.wikipedia.org/wiki/Rendering_(computer_graphics)) effects on graphics hardware with a high degree of flexibility. Shaders are used to program the [graphics processing unit](http://en.wikipedia.org/wiki/Graphics_processing_unit) (GPU) programmable [rendering pipeline](http://en.wikipedia.org/wiki/Rendering_pipeline), which has mostly superseded the fixed-function pipeline that allowed only common geometry transformation and pixel-shading functions; with shaders, customized effects can be used.

2. SOAP, originally defined as Simple Object Access Protocol, is a [protocol](http://en.wikipedia.org/wiki/Protocol_(computing)) specification for exchanging structured information in the implementation of [Web Services](http://en.wikipedia.org/wiki/Web_Service) in [computer networks](http://en.wikipedia.org/wiki/Computer_network). It relies on [Extensible Markup Language](http://en.wikipedia.org/wiki/XML) (XML) for its message format

3.XML, Extensible Markup Language (XML) is a [markup language](http://en.wikipedia.org/wiki/Markup_language) which defines a set of rules for encoding documents in a [format](http://en.wikipedia.org/wiki/File_format) which is both [human-readable](http://en.wikipedia.org/wiki/Human-readable_medium) and [machine-readable](http://en.wikipedia.org/wiki/Machine-readable), The design goals of XML emphasize simplicity, generality, and usability over the [Internet](http://en.wikipedia.org/wiki/Internet)

4. WPF Developed by [Microsoft](http://en.wikipedia.org/wiki/Microsoft), the Windows Presentation Foundation (or WPF) is a [computer-software](http://en.wikipedia.org/wiki/Computer_software) graphical subsystem for rendering user interfaces in Windows-based applications.

5. Writeable Bitmap: is a class used to get pixel array from image, Loop through the array, setting the individual pixel values as integer values that are evaluated as premultiplied ARGB32

6. Canvas: Defines an area within which you can explicitly position child elements by using coordinates that are relative to the area.

7. Rotate Transform: it is a class used to rotate an object about a specified point in a two-dimensional x-y coordinate system.

8. Scale Transform: a class used to scales an object in the two-dimensional x-y coordinate system

9. [Opacity](http://en.wikipedia.org/wiki/Opacity_(optics)), the degree to which light is not allowed to travel through