CSIT 6910A Independent Project, Spring 2014

Emulator for Flexible Display Devices

Final Report

Supervisor: Prof. David Rossiter

Student: Lee Hin Yu Kevin

Email: hyklee@connect.ust.hk

Table of Contents

1 Introduction

1.1 Background

Flexible display devices have been coming in to the market. A flexible display device, as its name implies, is a device which has a flexible display. Examples of such devices are:

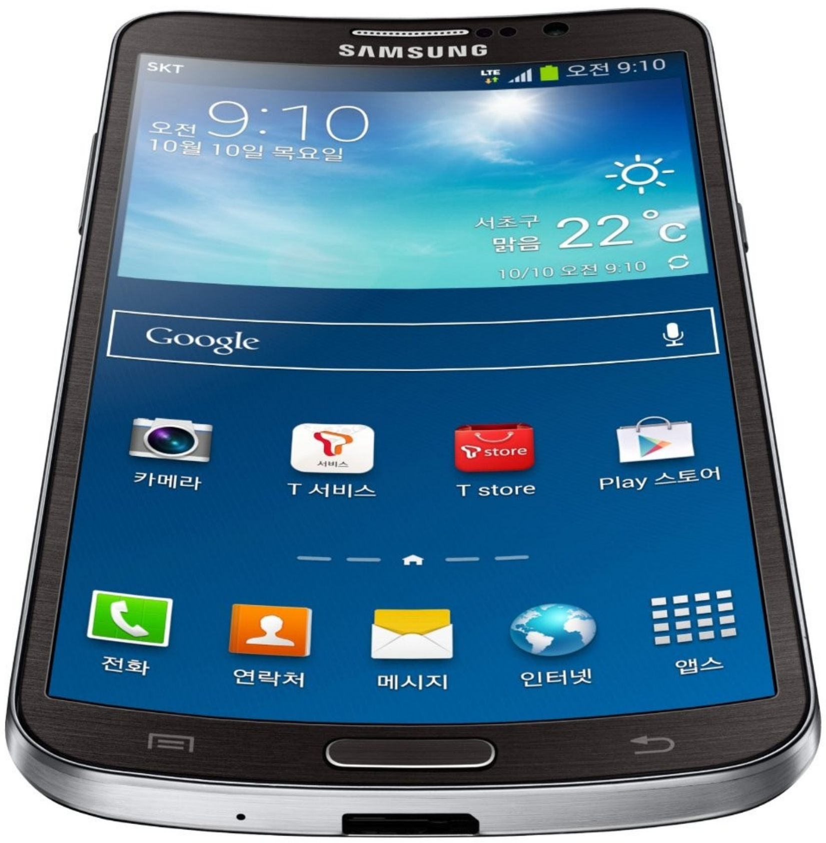


Figure 1: Samsung Galaxy Round. (See: http://www.samsung.com/sec/consumer/mobile-phone/mobile-phone/skt/SM-G910STAESKC)

Figure 2: LG G Flex. (See: http://www.lg.com/us/mobile-phones/gflex)

1.2 Objective

The objective of this project is to build an emulator for flexible mobile devices in order to allow web users to see the effects of such devices have on the web pages they are browsing, without needing the real devices.

The emulator is implemented as a Firefox extension, named Emuflex (short for **emu**lator for **flex**ible display devices). Users can install Emuflex and emulate flexible display devices in the free and open source Firefox browser.

1.3 Technology

The technologies used in the implementation are:

- XUL (XML User Interface Language): Mozilla's XML-based language for building user interfaces[1]
- JavaScript
    - Three.js r66 (a JavaScript library for WebGL)[2]
- HTML5
- CSS

1 https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL
2 http://threejs.org/

2 Project Overview

The user can launch Emuflex while visiting any web pages. Once Emuflex is launched, it will display the current web page in a standard mobile browser size, and open a new window showing the effects of the same web page being displayed on a flexible display device.

Emuflex depends upon the Responsive Design View (RDV) tool[3] that comes with Firefox, which allows the user to change the size and orientation of the device. The RDV tool is primarily designed for web developers who design web pages that automatically "respond" to various browser sizes so that the best layout is chosen for a particular browser.

Since in responsive design, the web pages accustom themselves to browser sizes, rather than the type of browsers such as desktop browser and mobile browser, the RDV tool does not identify itself as a mobile browser. Emuflex extends the RDV tool such that the user is able to *request web pages as a mobile browser* and *run the emulator in four modes*:

1. **Normal mode**. When operating in this mode, the Emuflex shows the web page in a flat plane. The user will be able to zoom in or out and rotate the plane.
2. **Curved mode**. Emuflex will emulate the effects of viewing the web page on a curved device - Samsung Galaxy Round and LG G Flex are two instances of curved devices. The user will be able to zoom in or out, rotate the curved surface, and control the curvature.
3. **Sine wave mode**. Emuflex will show the web page as if the web page was displayed on a sine wave. The picture below shows the effect that Emuflex is trying to mimic in this mode. The user will be able to zoom in or out, rotate the surface, and adjust the amplitude, frequency and offset of the sine wave.

3 https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_View

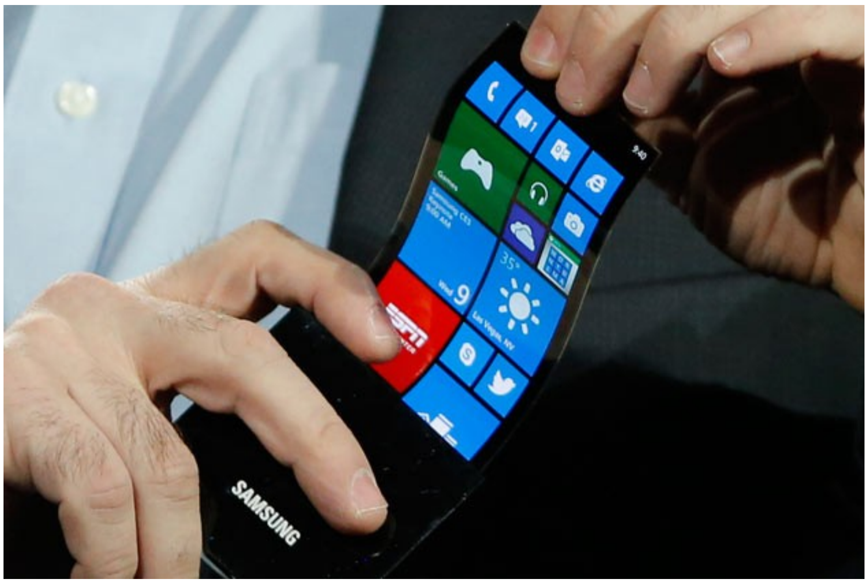


Figure 3: A real device which looks as if it was a sine wave.

4. **Flexible mode**. The emulation in this mode is more dynamic. The user is able to change the shape of the display as they wish, by moving some control points. The user is able to zoom in or out, rotate the surface, and customize the shape of the surface. The surface is much more customizable than the previous three modes in order to emulate the effects such as below.

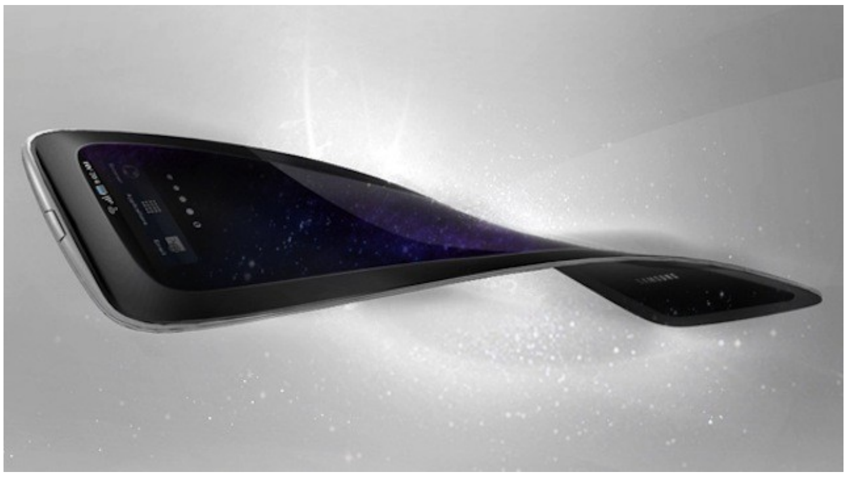


Figure 4: A highly flexible device.

3 Installation

Emuflex is tested on and compatible with Firefox versions 24.0 to 29.0.

3.1 Making an Installable Bundle (.xpi)

If you do not want to make your own installable bundle, skip this section and go to Section 3.2. Before you begin, if you use Windows to make the installable bundle, you will need to have Cygwin (http://cygwin.com/) installed. When you install Cygwin, be sure to install the *make* and *zip* utilities. If you use Mac OS X or Linux, you need to install *make* and *zip* too.

1. Open Cygwin on Windows or the terminal on Mac OS X and Linux.
2. Navigate to the *src* directory.

   ```
   $ cd /cygwin/<where_the_source_code_resides>/emuflex/src
   ```
3. Since there is a *Makefile*, you can simply issue the *make* command. If necessary, you can modify the make file before you issue *make*.

   ```
   $ make
   ```
4. Now the installable bundle is created under <*where_the_source_code_resides*>*/emuflex/bin*.

3.2 Installing Emuflex

Once you have the installable bundle, i.e. *emuflex.xpi*, you can simply drag and drop the .xpi file to Firefox and Firefox will ask you to install it. After the installation, restart Firefox to start using Emuflex.

You may want to install Emuflex in a profile other than the one that you usually use, so that Emuflex will not mess up your browser settings in case something goes wrong.

For more information, see https://support.mozilla.org/en-US/kb/profile-manager-create-and-remove-firefox-profiles.

4 Demonstration

In this section, we will demonstrate the usage of Emuflex. The link to the video demonstration here: http://youtu.be/tAH6E6Svrfg.

4.1 Launching Emuflex

To start Emuflex, click on the menu button on the right hand side. Then click on the Developer icon. Under the developer tools, click on Emuflex.

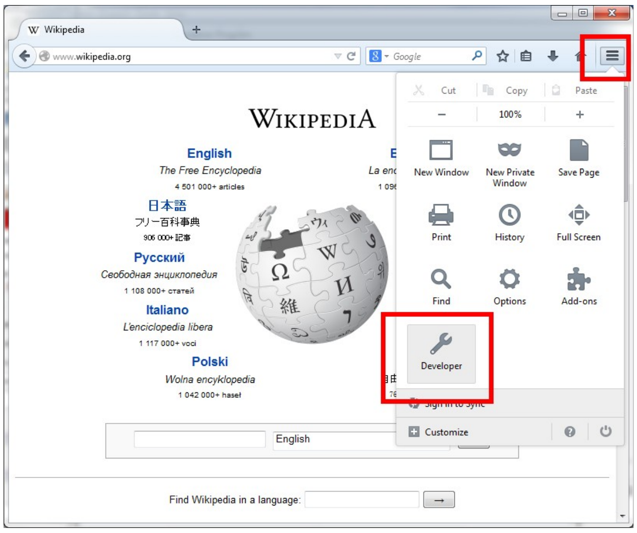


Figure 5: Click on the menu button on the right hand side. Then click on the Developer icon.

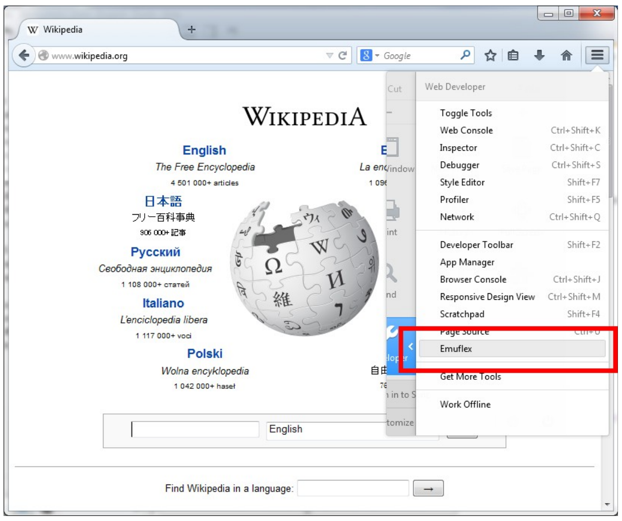


Figure 6: Under the developer tools, click on Emuflex.

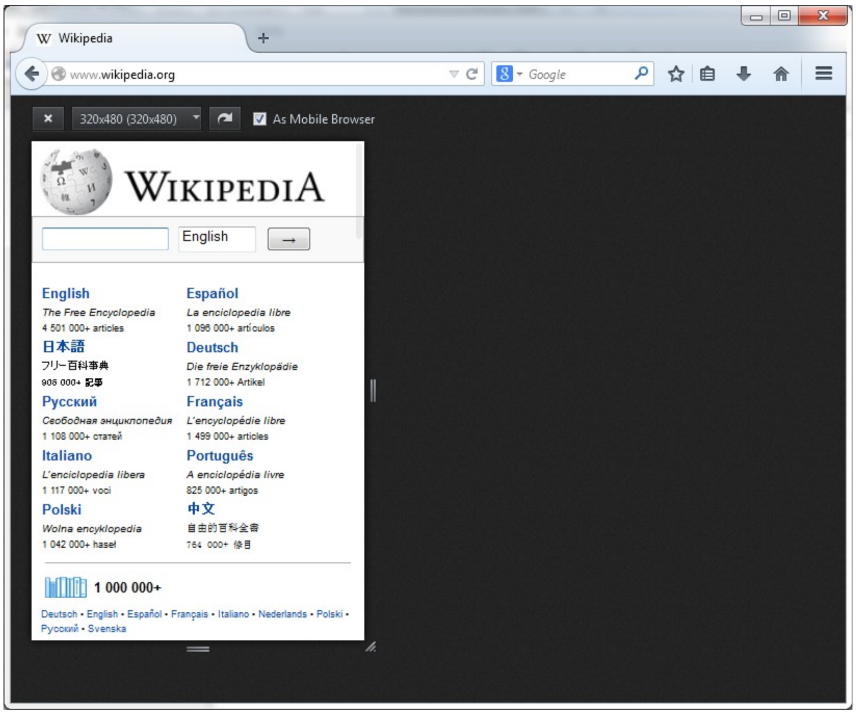
4.2 Normal Mode

Suppose the user is browsing Wikipedia. When the user launched Emuflex, the user will see two windows. One is a modified RDV tool that comes with Firefox. The other one is the emulator window.

With the modified RDV tool, the user can

1. Change the device size;
2. Rotate the device; and
3. Send requests as a mobile browser.

The default mode of the Emuflex is Normal Mode. The user is shown a flat plane and is able to zoom in or out.


Wikipedia

www.wikipedia.org

320x480 (320x480) ☑ As Mobile Browser

WIKIPEDIA

English →

English
The Free Encyclopedia
4 501 000+ articles

Español
La enciclopedia libre
1 096 000+ artículos

日本語
フリー百科事典
906 000+ 記事

Deutsch
Die freie Enzyklopädie
1 712 000+ Artikel

Русский
Свободная энциклопедия
1 108 000+ статей

Français
L'encyclopédie libre
1 499 000+ articles

Italiano
L'enciclopedia libera
1 117 000+ voci

Português
A enciclopédia livre
825 000+ artigos

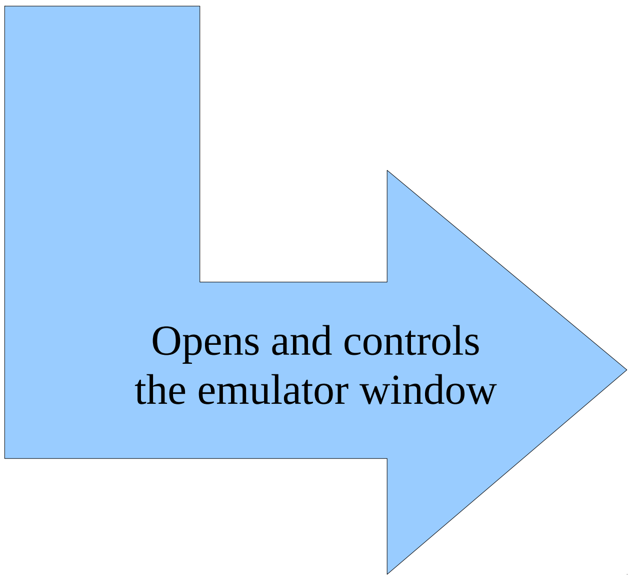
Polski
Wolna encyklopedia
1 042 000+ haseł

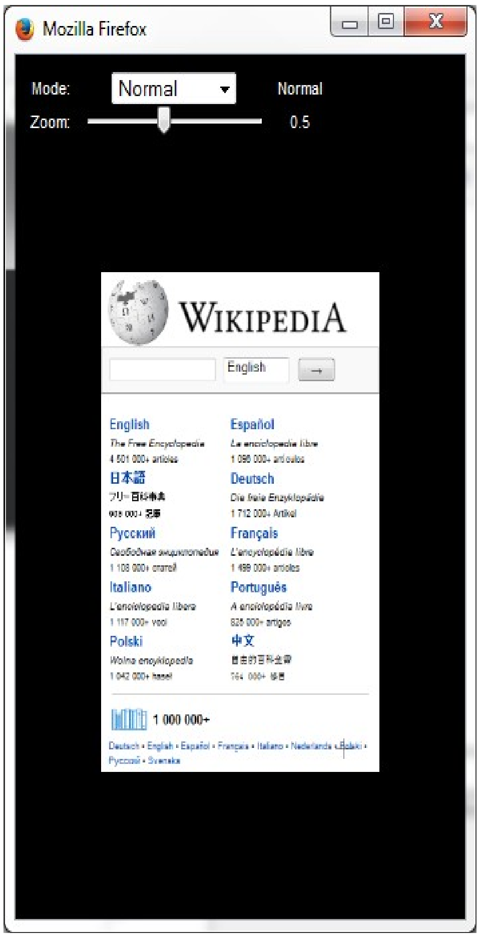
中文
自由的百科全書
764 000+ 條目

1 000 000+

Deutsch • English • Español • Français • Italiano • Nederlands • Polski • Русский • Svenska

Figure 7: The modified RDV tool.




Mozilla Firefox

Mode: Normal — Normal

Zoom: 0.5

WIKIPEDIA

English →

English
The Free Encyclopedia
4 501 000+ articles

Español
La enciclopedia libre
1 096 000+ artículos

日本語
フリー百科事典
906 000+ 記事

Deutsch
Die freie Enzyklopädie
1 712 000+ Artikel

Русский
Свободная энциклопедия
1 108 000+ статей

Français
L'encyclopédie libre
1 499 000+ articles

Italiano
L'enciclopedia libera
1 117 000+ voci

Português
A enciclopédia livre
825 000+ artigos

Polski
Wolna encyklopedia
1 042 000+ haseł

中文
自由的百科全書
764 000+ 條目

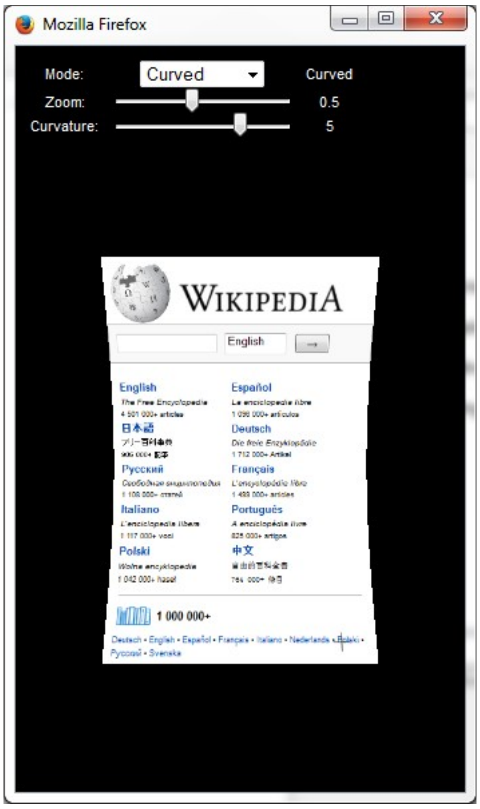
1 000 000+

Deutsch • English • Español • Français • Italiano • Nederlands • Polski • Русский • Svenska

Figure 8: The emulator window

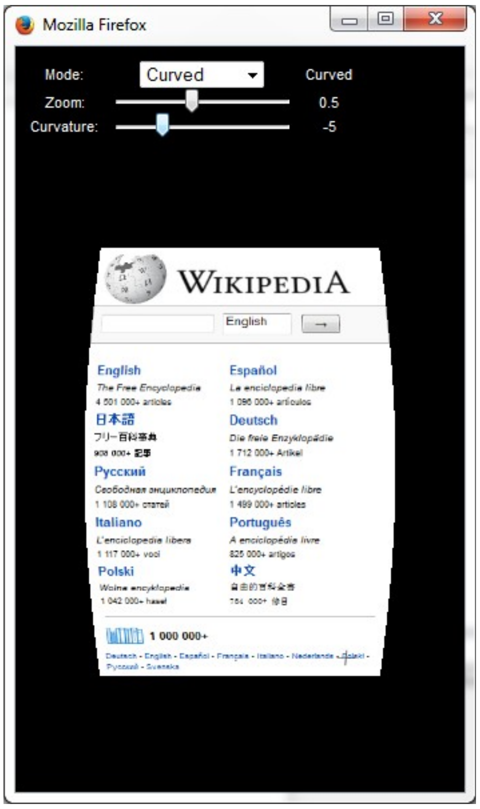
4.3 Curved Mode

Switching to Curved Mode, the user can change the curvature (figures 5 and 6).



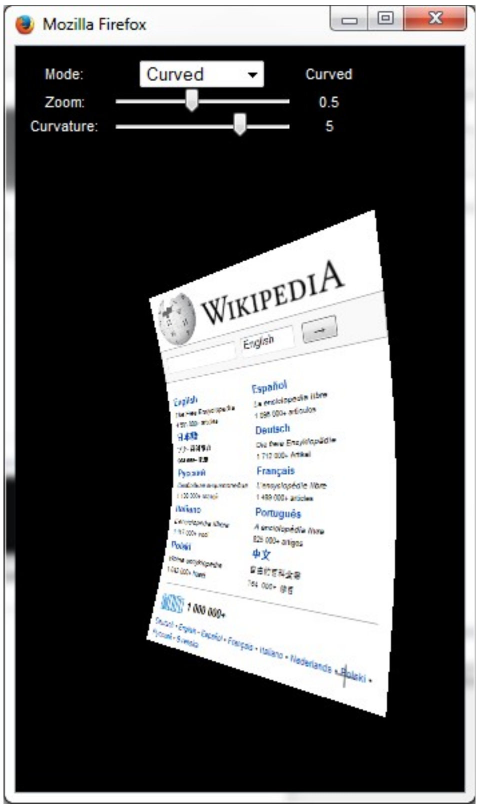
Mode: Curved Curved
Zoom: 0.5
Curvature: 5

Figure 10: Curved mode. Default curvature.



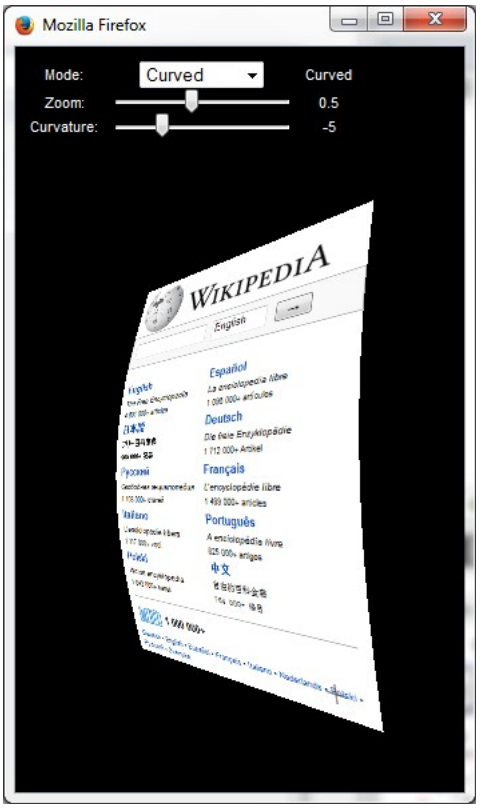
Mode: Curved Curved
Zoom: 0.5
Curvature: -5

Figure 9: Curved mode. Curvature changed.



Mode: Curved Curved
Zoom: 0.5
Curvature: 5

Figure 11: Curved mode. Side view.

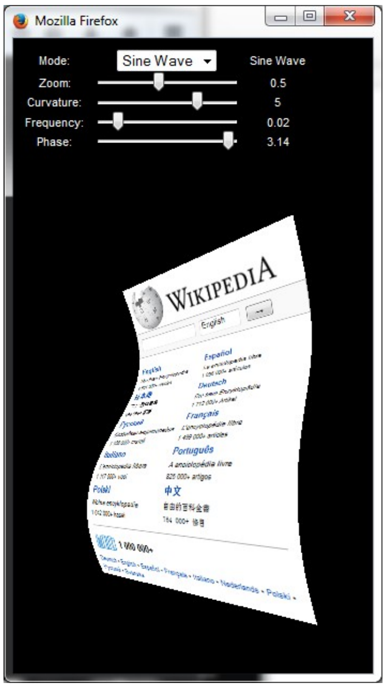


Mode: Curved Curved
Zoom: 0.5
Curvature: -5

Figure 12: Curved mode. Side view.

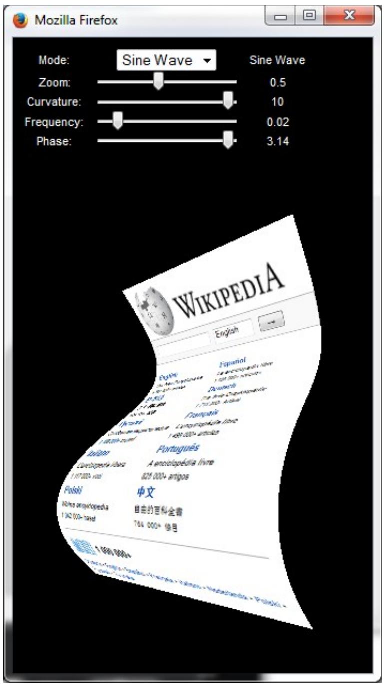
4.4 Sine Wave Mode

The user can change the amplitude (curvature), frequency and phase of the sine wave.



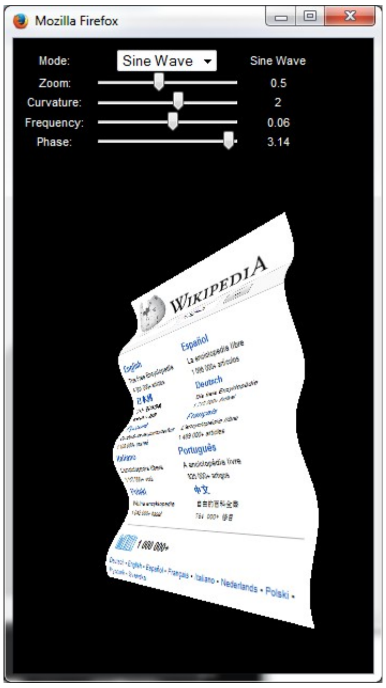
Mode: Sine Wave — Sine Wave
Zoom: 0.5
Curvature: 5
Frequency: 0.02
Phase: 3.14

Figure 13: Sine wave mode.



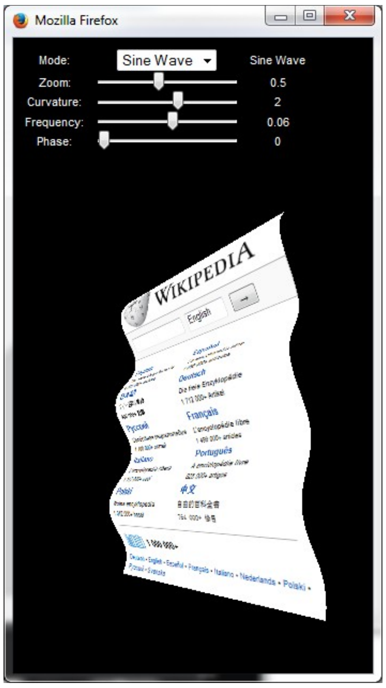
Mode: Sine Wave — Sine Wave
Zoom: 0.5
Curvature: 10
Frequency: 0.02
Phase: 3.14

Figure 14: Sine wave mode. Curvature changed.



Mode: Sine Wave — Sine Wave
Zoom: 0.5
Curvature: 2
Frequency: 0.06
Phase: 3.14

Figure 15: Sine wave mode. Frequency changed.

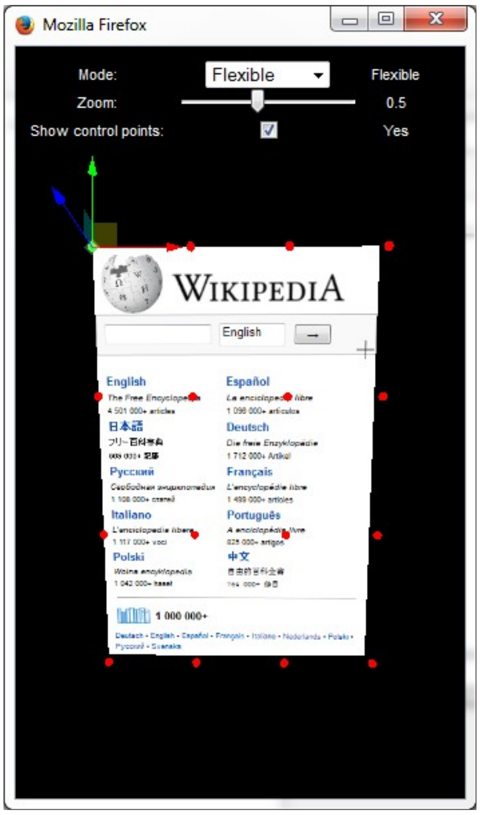


Mode: Sine Wave — Sine Wave
Zoom: 0.5
Curvature: 2
Frequency: 0.06
Phase: 0

Figure 16: Sine wave mode. Phase offset changed.

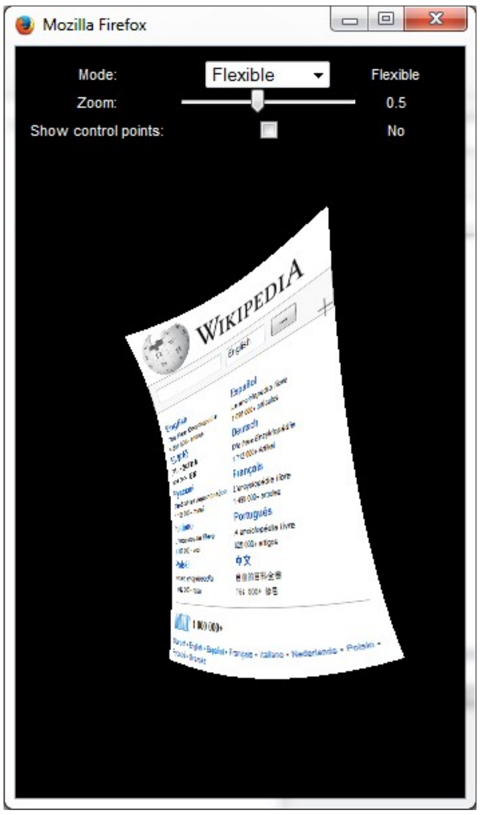
4.5 Flexible Mode

The user can move around control points to customize the shape of the surface.



Mode: Flexible — Flexible
Zoom: 0.5
Show control points: ✓ — Yes

Figure 17: Flexible mode. Moving control points.



Mode: Flexible — Flexible
Zoom: 0.5
Show control points: — No

Figure 18: Flexible mode. Done changing the shape and control points hidden.

4.6 Synchronization

Emuflex will continuously update the emulation. Whatever changes in the browser window also changes in the emulator window.

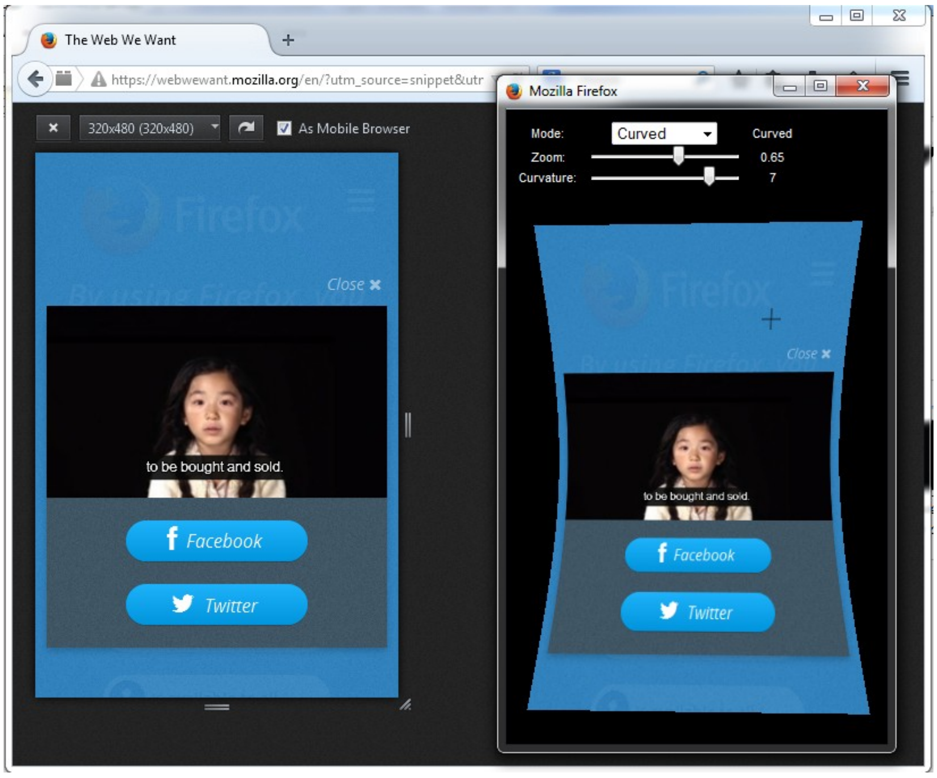


Figure 19: A video is being played.

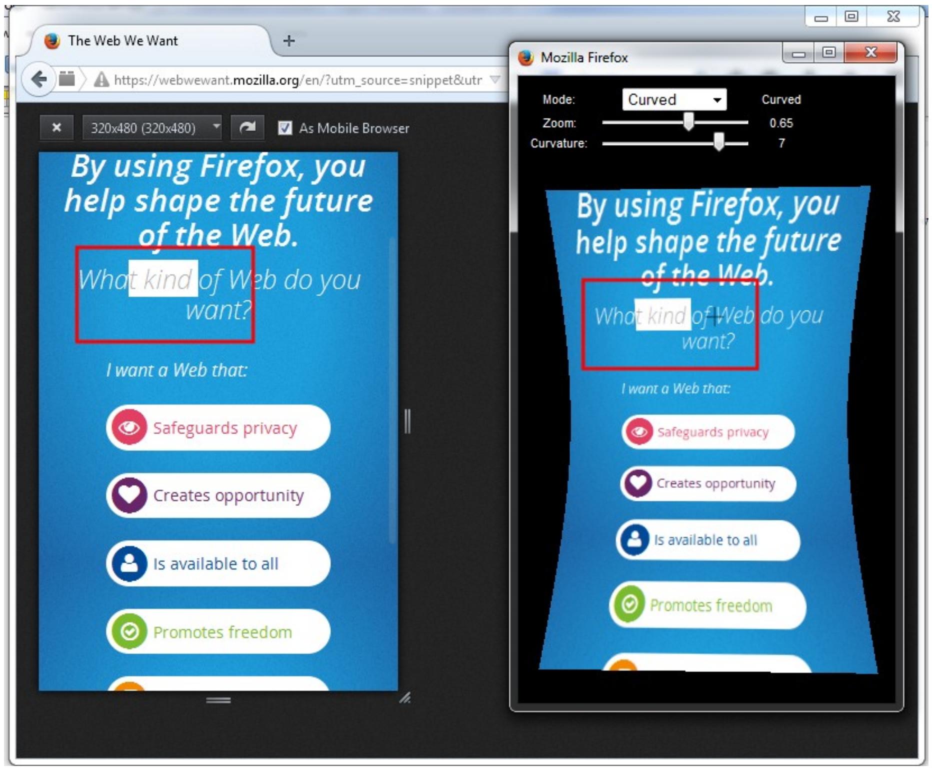


Figure 20: The cursor in the browser window is shown as a cross in the emulator window.

5 Implementation

In this section, we will briefly discuss the implementation of the four modes. In the implementation of the four modes, two mathematical tools are used, namely, the sine function and the Bezier surface[4]. Since the sine function is given by the JavaScript Math object[5], the code for the sine function is not shown. Emuflex has its own implementation of the Bezier surface, so the code is attached in the end of this section.

5.1 Normal Mode

Once Emuflex is launched, the browser window opens the emulator window and keeps drawing its content to a hidden canvas in the emulator window. The hidden canvas then serves as a texture to the flat surface which is an instance of PlaneGeometry of three.js. Since the browser window keeps updating the emulator window, the two windows are synchronized.

5.2 Curved Mode

In Curved Mode, the shape of the surface is defined by a sine function, i.e. $a \sin(2\pi \frac{1}{\lambda} t - \pi)$, where a is the amplitude, t is time, and λ is half the height of the browser in portrait or half the width of the browser in landscape.

Changing the curvature is indeed changing the amplitude. If the dimensions of the browser change, since λ also changes with the dimensions, the same portion of the sine wave is always used, as depicted in the figure below.

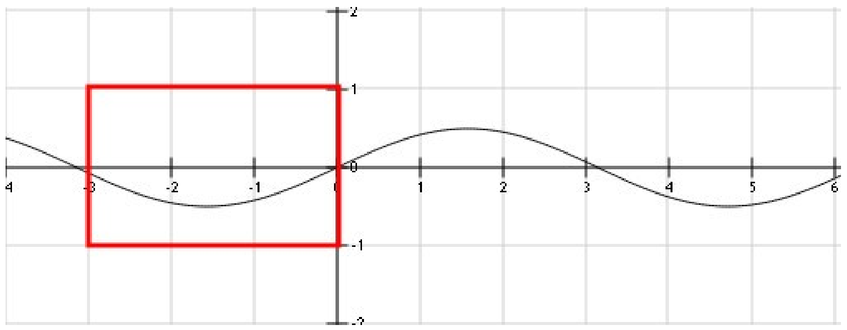


Figure 21: The red part of the sine curve is used to define the shape of the plane geometry.

4 https://en.wikipedia.org/wiki/B%C3%A9zier_surface

5 https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

5.3 Sine Wave Mode

Similar to Curved Mode, the shape of the surface is also defined by a sine function, i.e. $a \sin(2\pi f t - t_0)$ where a is the amplitude, f is the frequency, t is time, t_0 is the phase offset.

The user can change the amplitude, frequency and the phase offset.

5.4 Flexible Mode

The shape of the surface is implemented as the Bezier surface which is given by

$$p(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u) B_j^m(v) k_{i,j} \text{ where } B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad .$$

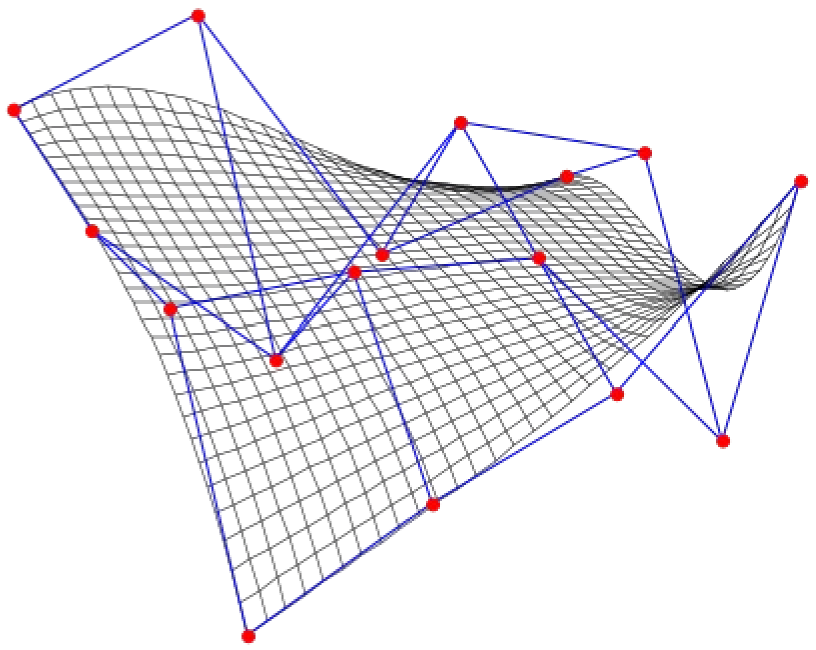


Figure 22: A Bezier surface

The Bezier surface is of degree(n , m) and is defined by $(n+1)(m+1)$ control points $k_{i,j}$. p is a point on the surface. In the implementation of Emuflex, u is the y direction and v is the x direction on the surface.

The Bezier surface implemented is of degree(3, 3). So there are 4 by 4 control points. The implementation of the Bezier surface is as follows.

```
// degree = {n, m}
// t = {u, v}
// ctrlPts is an array of control points
var getBezier = function(degree, t, ctrlPts) {

    // binomial coefficient
    var choose = function(n, k) {

      var factorial = function fact(k) {
        if (k < 2) {
          return 1;
        }
        return k * fact(k - 1);
      };

      return factorial(n) / factorial(k) / factorial(n - k);
    }

    var C = ctrlPts;
    var i, j, len_i, len_j;
    var n = degree.n, m = degree.m, u = t.u, v = t.v;
    var B_n = [], B_m = [];

    for (i = 0, len_i = n + 1; i < len_i; i++) {
      B_n.push(
        function(u) {
          return choose(n, i) * Math.pow(1 - u, n - i) * Math.pow(u, i);
        }
      );
    }

    for (j = 0, len_j = m + 1; j < len_j; j++) {
      B_m.push(
        function(v) {
          return choose(m, j) * Math.pow(1 - v, m - j) * Math.pow(v, j);
        }
      );
    }

    var x = 0, y = 0, z = 0, b_i, b_j;
```

```
  for (i = 0, len_i = n + 1; i < len_i; i++) {

    b_i = B_n[i](u);

    for (j = 0, len_j = m + 1; j < len_j; j++) {

      b_j = B_m[j](v);

      x += C[i][j].x * b_i * b_j;
      y += C[i][j].y * b_i * b_j;
      z += C[i][j].z * b_i * b_j;

    }

  }

  return new THREE.Vector3(x, y, z);
};
```

6 Meeting Minutes

Minutes of the 1st Project Meeting

Date: 21 Feb 2014
Time: 1:30 PM
Place: Rm. 3512
Attending: Kevin Lee, Prof. ROSSITER
Absent: None
Recorder: Kevin Lee

1. Things to do by the next meeting (in 3 weeks)
   - Implement the user interface
   - Have some working emulation
2. Meeting adjournment and next meeting
   The meeting was adjourned at 1:40 PM. The next meeting will be held in March.

Minutes of the 2nd Project Meeting

Date: 21 March 2014
Time: 1:30 PM
Place: Rm. 3512
Attending: Kevin Lee, Prof. ROSSITER
Absent: None
Recorder: Kevin Lee

1. Things to do by the next meeting (in 3 weeks)
   - Currently, we are working in 2D. Implement the work in 3D.
   - More user interactions will be programmed.
2. Meeting adjournment and next meeting
   The meeting was adjourned at 1:40 PM. The next meeting will be held in April.

Minutes of the 3rd Project Meeting

Date: 22 April 2014
Time: 2:30 PM
Place: Rm. 3512
Attending: Kevin Lee, Prof. ROSSITER
Absent: None
Recorder: Kevin Lee

1. Things to do
   - Submit the first draft of the report by May 3rd
   - Complete “Flexible Mode”
2. Meeting adjournment and next meeting
   The meeting was adjourned at 1:45 PM. The next meeting will be held in May.

Minutes of the 4th Project Meeting

Date: 14 May 2014
Time: 3:30 PM
Place: Rm. 3512
Attending: Kevin Lee, Prof. ROSSITER
Absent: None
Recorder: Kevin Lee

1. Things to do
    - Specify the version of Firefox used in the development
    - Shorten the video
    - Rearrange/ remove items in the video
    - Add installation instructions to the report
    - Make some small changes to the report
    - Submit the final report, video and the installable bundle in a few days
2. Meeting adjournment
   The meeting was adjourned at 3:45 PM.