ONLINE INTERACTIVE LANGUAGE UTILITIES FOR BANGLA AND ENGLISH USING PHP SCRIPT AND ADOBE FLASH TECHNOLOGY

Shiblee Imtiaz Hasan
ID: 07110049 / 09241001

Department of Computer Science and Engineering
April 2010

BRAC University, Dhaka, Bangladesh

DECLARATION

I hereby declare that this thesis is based on the results found by myself. Materials of work found by other researcher are mentioned by reference. This thesis, neither in whole nor in part, has been previously submitted for any degree.

Signature of
Supervisor

Signature of
Author

ACKNOWLEDGMENTS

Special thanks to the Center for Research on Bangla Language Processing (CRBLP) team, who developed the Bang Text-To-Speech and Optical Character Recognition system and the web version of them, for giving me an opportunity to use their system, server and modify their application to customize for its use in the application I am developing. I thank Princeton University for providing a database of English WordNet; University of Edinburgh and Carnegie Mellon University for developing the Festival Speech Synthesizer engine. Thanks to Prof. Mumit Khan and Matin Saad Abdullah for providing guidance in developing this aggregate application consisting of several existing utilities. I also thank the team who built PHP script and Adobe Systems who is constantly developing the Adobe Flash Technology.

TABLE OF CONTENTS

Topic	Page
TITLE	i
DECLARATION	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
TABLE OF FIGURES	ix
CHAPTER ONE. INTRODUCTION	1
1.1 Background	1
1.2 Objectives	1
1.3 Benefits	1
CHAPTER TWO. PROJECT OVERVIEW	3
2.1 Text To Speech	4
2.2 Image To Text/Speech	5
2.3 Speech Recognition	5
2.4 Bangla and English WordNet	6
CHAPTER THREE. EXISTING TECHNOLOGIES	7
3.1 Text To Speech	7
3.2 Image To Text/Speech	8
3.3 Bangla and English WordNet	9
CHAPTER FOUR. LIMITATIONS AND SOLUTIONS	11
4.1 Bangla Glyph Display in Adobe Flash	11
4.2 Cross-Domain Communication in Adobe Flash	11

4.3 Using Unicode Glyphs 11
4.4 File Uploads 11
4.5 Multiple Input 12
4.6 Speech segmentation 12
4.7 Text segmentation 12
4.8 Word sense disambiguation 12
4.9 Syntactic ambiguity 12
4.10 Imperfect or irregular input 13
4.11 Speech acts and plans 13

CHAPTER FIVE. PROJECT DESIGN AND DEVELOPMENTS 15

LITERATURE SURVEY 19

APPENDIX 21

A.1 Source Codes 21
A.2 Open-Source Documentations 57

REFERENCES 75

TABLE OF FIGURES

Code	Title	Page
Figure 2.1	Facebook application screenshot	6
Figure 5.1	Flash file frames in timeline	15
Figure 5.2	Interaction between interface and servers	16
Figure 5.3	Implementation of languages in frames	16
Figure 5.4	The WordNet utility	17
Figure 5.5	Text to Speech utility	17

CHAPTER ONE

INTRODUCTION

1.1 Background

Natural human language processing with computing machines has been a very important area of Computer Science in achieving automation in different parts of our daily life. Recent advances in Natural Language Processing (NLP) have enabled us to talk to and get audio-visual assistance from computers. Computers now-a-days can also automatically summarize certain text depending on its context. Many researches are now focusing on better accuracy of inter-language translations.

1.2 Objectives

The necessity of these language tools are felt by people from all parts of the society. Students can use them to expand their knowledge on different languages and its grammar; the illiterate can get assistance in learning their own languages; even the physically impaired individuals can find these utilities as an aid in different parts of their lives. Due to a very low literacy rate in Bangladesh, these tools have the potential be vital in enabling large crowd people to be literate in Bangla and English in an easy interactive method.

English being an internationally recognized language, most of the researches have been done based on English. However, recently there are tools being developed for Bangla. As a user of such software solutions, a person would want to have more functionality in a single application which is highly portable which is, at the moment, not available in market.

My objective, through this thesis, has been to design and implement a web-based application for Bangla and English which contains most popular and widely used language utilities which will also focus on the accessibility of visually impaired individuals.

A Facebook application version of the utilities has also been made. Through Facebook (www.facebook.com), different people worldwide can gain access to their family's and friend's user profiles and thus allow them to interact with them through photos, updates, notes etc. Facebook has become very popular in different parts of the world enabling its consumers to reach more than a hundred million (and soon, 500 million!) from its advent in 2004. With its positive growth, users are also becoming advanced in terms of their Facebook application usage. Some are not only using Facebook for the sake of social connection. People are now using many educational applications to either educate or get educated.

1.3 Benefits

Improved performance: A 12-year meta-analysis of research by the U.S. Department of Education found that higher education students in online learning generally performed better than those in face-to-face courses.

Convenience and flexibility to learners: in many contexts, eLearning is self-paced and the learning sessions are available 24x7. Learners are not bound to a specific day/time to physically attend classes. They can also pause learning sessions at their convenience.

CHAPTER TWO

PROJECT OVERVIEW

The web application is using Adobe Flash as its front-end user interface. Data manipulation in the back-end is being done by the collaboration between Adobe Flash and PHP Script in the server.

Adobe Flash is a multimedia platform developed by Adobe Systems. It is commonly used to create animation and various web page Flash components, to integrate video into web pages, and more recently, to develop rich Internet applications. Flash can manipulate vector and raster graphics, and supports bidirectional streaming of audio and video. It contains a scripting language called ActionScript. Several software products, systems, and devices are able to create or display Flash content, including Adobe Flash Player, which is available free for most common web browsers, some mobile phones and for other electronic devices (using Flash Lite). Using flash in this application will enable the users to interact with more visual features thus making it more attractive and user friendly.

PHP, or PHP: Hypertext Preprocessor, is a widely used, general-purpose scripting language that is designed for web development, to produce dynamic web pages. It can be embedded into HTML and generally runs on a web server, which needs to be configured to process PHP code and create web page content from it. It can be deployed on most web servers and on almost every operating system and platform free of charge. As PHP runs on every server and the interaction between Adobe Flash and PHP is very easy to achieve, we are choosing PHP as our back-end data processor.

The reason behind building a web application is its portability. Now that Flash can be played in Mobile Devices, a person can access the software/application through internet no matter where or which computer platform that person is.

The application is currently designed to contain 4 different language utilities integrated in it. When a new utility is developed it will be possible to integrate them as well. Followings are the descriptions of the different operations of the application.

2.1 Text To Speech

Text to speech or Speech synthesis is the artificial production of human speech. This text-to-speech (TTS) system converts Bangla and English language text into corresponding speech.

Synthesized speech can be created by concatenating pieces of recorded speech that are stored in a database. Systems differ in the size of the stored speech units; a system that stores phones or diphones provides the largest output range, but may lack clarity. For specific usage domains, the storage of entire words or sentences allows for high-quality output. Alternatively, a synthesizer can incorporate a model of the vocal tract and other human voice characteristics to create a completely "synthetic" voice output.

The quality of a speech synthesizer is judged by its similarity to the human voice and by its ability to be understood. An intelligible text-to-speech program allows people with visual impairments or reading disabilities to listen to written works on a home computer. Many computer operating systems have included speech synthesizers since the early 1980s.

2.2 Image To Text/Speech

The Image To Text/Speech conversion or Optical character recognition (OCR) is the translation of images of handwritten, typewritten or printed text (usually captured by a scanner or camera) into computer-editable text. The text can then be translated to speech.

OCR Software and ICR Software technology are analytical artificial intelligence systems that consider sequences of characters rather than whole words or phrases. Based on the analysis of sequential lines and curves, OCR and ICR make 'best guesses' at characters using database look-up tables to closely associate or match the strings of characters that form words.

2.3 Speech Recognition

Speech recognition (also known as automatic speech recognition or computer speech recognition) converts spoken words to text. There are no tools currently available for Bangla Speech Recognition so its development will be a key objective in this section.

Speech recognition (also known as automatic speech recognition or computer speech recognition) converts spoken words to text. The term "voice recognition" is sometimes used to refer to speech recognition where the recognition system is trained to a particular speaker - as is the case for most desktop recognition software, hence there is an element of speaker recognition, which attempts to identify the person speaking, to better recognize what is being said. Speech recognition is a broad term which means it can recognize almost anybody's speech - such as a call-centre system designed to recognize many voices. Voice recognition is a system trained to a particular user, where it recognizes their speech based on their unique vocal sound.

2.4 Bangla and English WordNet

WordNet is a lexical database for a language. It groups words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable and to support automatic text analysis and artificial intelligence applications. This application will implement Bangla and English WordNet.

Figure 2.1: Facebook application version screenshot

CHAPTER THREE

EXISTING TECHNOLOGIES

Following are the technologies/utilities that will be used to complete this application:

3.1 Text To Speech

CRBLP's "Festival" Bangla speech synthesizer, which is a software capable of making artificial speech in place of a real human. Festival is the most complete open-source multilingual, general-purpose synthesis system available.

A text-to-speech system (or "engine") is composed of two parts: a front-end and a back-end. The front-end has two major tasks. First, it converts raw text containing symbols like numbers and abbreviations into the equivalent of written-out words. This process is often called text normalization, pre-processing, or tokenization. The front-end then assigns phonetic transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation that is output by the front-end. The back-end—often referred to as the synthesizer—then converts the symbolic linguistic representation into sound.

Festival is a general multi-lingual speech synthesis system originally developed at Centre for Speech Technology Research (CSTR) at the

University of Edinburgh. Substantial contributions have also been provided by Carnegie Mellon University and other sites. It is distributed under a free software license similar to the BSD License.

It offers a full text to speech system with various APIs, as well an environment for development and research of speech synthesis techniques. It is written in C++ with a Scheme-like command interpreter for general customization and extension.

Festival is designed to support multiple languages, and comes with support for English (British and American pronunciation), Welsh, and Spanish. Voice packages exist for several other languages, such as Castilian Spanish, Czech, Finnish, Hindi, Italian, Marathi, Polish, Russian and Telugu.

3.2 Image To Text/Speech

CRBLP's BanglaOCR is an Optical Character Recognition tool for Bangla script. It can take a Portable Network Graphics (PNG) image file as an input and output an editable text.

Optical character recognition, usually abbreviated to OCR, is the mechanical or electronic translation of scanned images of handwritten, typewritten or printed text into machine-encoded text. It is widely used to convert books and documents into electronic files, to computerize a record-keeping system in an office, or to publish the text on a website. OCR makes it possible to edit the text, search for a word or phrase, store it more compactly, display or print a copy free of scanning artifacts, and apply techniques such as machine translation, text-to-speech and text mining to it. OCR is a field of research in pattern recognition, artificial intelligence and computer vision.

OCR systems require calibration to read a specific font; early versions needed to be programmed with images of each character, and worked on one font at a time. "Intelligent" systems with a high degree of recognition accuracy for most fonts are now common. Some systems are capable of reproducing formatted output that closely approximates the original scanned page including images, columns and other non-textual components.

3.3 Bangla and English WordNet

There has been a Bangla WordNet developed by CRBLP which contains a database of dictionary and thesaurus of Bangla. The English WordNet database being used is the one developed by Princeton University.

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short, general definitions, and records the various semantic relations between these synonym sets. The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications. The database and software tools have been released under a BSD style license and can be downloaded and used freely. The database can also be browsed online. WordNet was created and is being maintained at the Cognitive Science Laboratory of Princeton University under the direction of psychology professor George A. Miller. Development began in 1985. Over the years, the project received funding from government agencies interested in machine translation. George Miller and Christiane Fellbaum were awarded the 2006 Antonio Zampolli Prize for their work with WordNet.

CHAPTER FOUR

LIMITATIONS AND SOLUTIONS

Several limitations exist in the current tools and technologies to be used. A list of limitations and their workarounds are given as following:

4.1 Bangla Glyph Display in Adobe Flash

Despite of being supported by Unicode 4.1 and above, Bangla glyphs are not displayed properly by Adobe Flash in different platforms and browsers. It might be required to add additional functions to enable all users to view Bangla.

4.2 Cross-Domain Communication in Adobe Flash

Due to security issues, Adobe Flash doesn't allow data from different domains to be displayed in its Shockwave Flash files. To solve these limitations, different PHP connectors are being used.

4.3 Using Unicode Glyphs in Query and Other Languages

Although most languages now support almost full Unicode support for string types, it is always required to test input/output to components to make sure the variables are storing the right string.

4.4 File Writing/Uploading in Server

While multiple users are uploading, there is a possibility that the filenames are the same which might trigger an overwrite. To overcome this, the uploaded files were given random non-duplicate string filenames.

4.5 Multiple Input of Similar Data

The server executions are quite expensive in terms of time and memory. This program implements memoization to keep history of processed string and its output audio file URL.

4.6 Speech segmentation

In most spoken languages, the sounds representing successive letters blend into each other, so the conversion of the analog signal to discrete characters can be a very difficult process. Also, in natural speech there are hardly any pauses between successive words; the location of those boundaries usually must take into account grammatical and semantic constraints, as well as the context.

4.7 Text segmentation

Some written languages like Chinese, Japanese and Thai do not have single-word boundaries either, so any significant text parsing usually requires the identification of word boundaries, which is often a non-trivial task.

4.8 Word sense disambiguation

Many words have more than one meaning; we have to select the meaning which makes the most sense in context.

4.9 Syntactic ambiguity

The grammar for natural languages is ambiguous, i.e. there are often multiple possible parse trees for a given sentence. Choosing the most appropriate one usually requires semantic and contextual information. Specific problem

components of syntactic ambiguity include sentence boundary disambiguation.

4.10 Imperfect or irregular input

Includes Foreign or regional accents and vocal impediments in speech; typing or grammatical errors, OCR errors in texts.

4.11 Speech acts and plans

A sentence can often be considered an action by the speaker. The sentence structure alone may not contain enough information to define this action.

CHAPTER FIVE

PROJECT DESIGN AND CURRENT DEVELOPMENTS

The base engine of the application is in Flash and the different components are designed in different frames. This makes sure that during context/feature change the application doesn't use excess processing/memory resources of the computer. After loading the application in browser it prompts for the language preference of the instructions. Each button rollover is associated with the spoken word of the button's titles. The screen reading is turned on by default to make sure that the visually impaired individuals can have an easy access to the application. The application has been hosted at: *http://labs.com.bd/voice*

The application starts with the frame with instructions on how to use the tools available to the user. The PREVIOUS and NEXT buttons located on left and right at the bottom changes the utilities. The "Hot Corner" feature is adopted from Mac to allow users to change the items/utilities easily without having to search for the appropriate button.

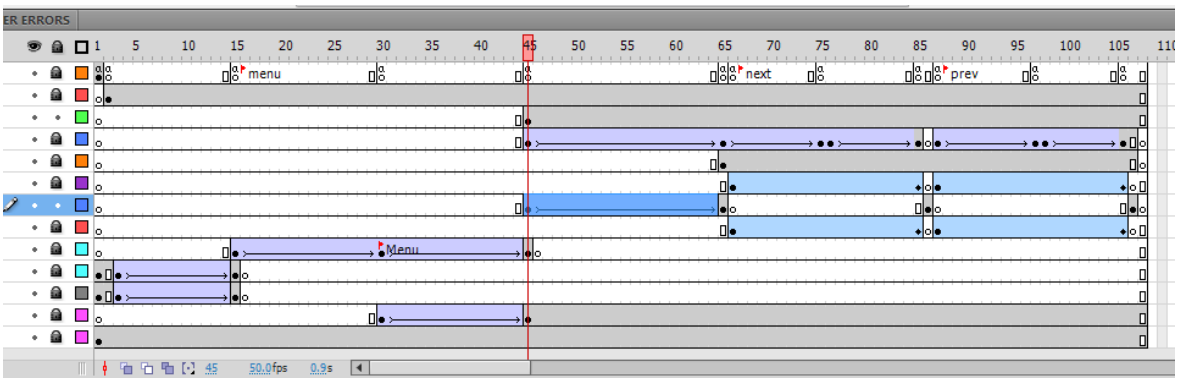


Figure 5.1: The timeline of parent frames

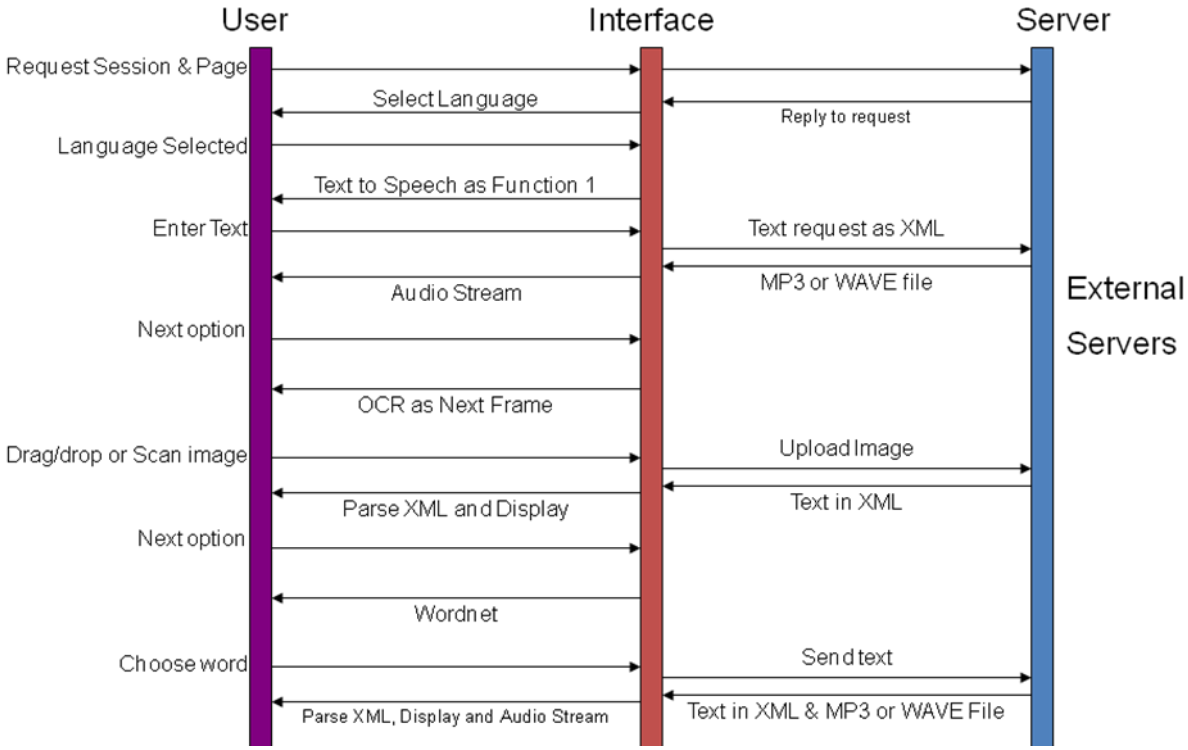
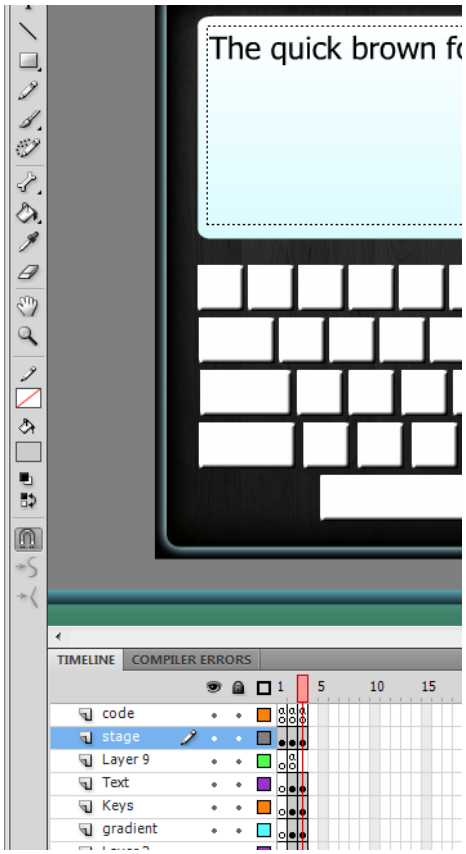


Figure 5.2: Interaction between Interface and Servers



The sound used in the application were planned to be streaming sound, so even if the user inputs a large text, the audio file will be streamed in part by part hence reducing waiting time for the user. Unfortunately the Festival Speech Synthesis System is giving output as WAV file only. If a proper application like LAME is installed in the server, it will be able to convert the WAV to MP3.

Figure 5.3: Implementation of frames for different languages. Frame 1 is preload, Frame 2 is Bangla and Frame 3 is for English

Currently the Text To Speech (TTS), Image to Text (OCR) and WordNet (with limited database) utilities are functional. The TTS feature can load external files and make speech out of it. It also provides an on-screen Bangla Keyboard to write the text and listen to that text. As Cross-domain file loading is still not enabled in Flash, the application copies the WAV to local server and then plays via an iframe. The OCR can upload images in server and process the file to determine the text in it.

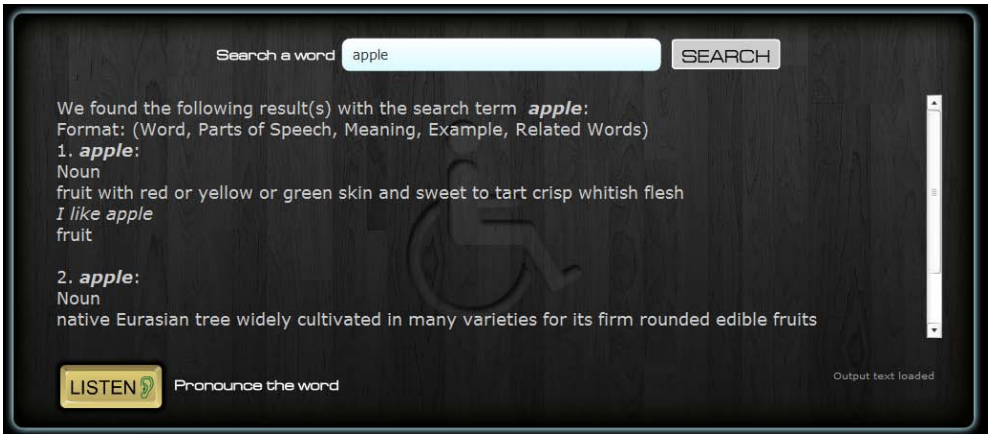


Figure 5.4: The WordNet utility for English

The Microphone capture from computer to application has been implemented to assist the speech recognition utility.

Figure 5.5: A screen preview of the application's Text to Speech utility

LITERATURE SURVEY

Md. Abul Hasnat, Muttakinur Rahman Chowdhury and Mumit Khan. "Integrating Bangla script recognition support in Tesseract OCR", *Proc. of Conference on Language and Technology 2009* (CLT09), Lahore, Pakistan, January 22-24, 2009.

Farhana Faruqe and Mumit Khan. "BWN - A Software Platform for Developing Bengali WordNet", *International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering* (CISSE 08), December 5 - 13, 2008.

Firoj Alam , S. M. Murtoza Habib and Mumit Khan. "Acoustic Analysis of Bangla Consonants", *Spoken Language Technologies for Under-resourced language* (SLTU'08), Hanoi, Vietnam, May 5 - 7, 2008.

Md. Abul Hasnat, S M Murtoza Habib and Mumit Khan. "A High Performance Domain Specific OCR For Bangla Script", *International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering* (CISSE), December 3 - 12, 2007.

Md. Abul Hasnat, Jabir Mowla, Mumit Khan. "Isolated and Continuous Bangla Speech Recognition Implementation, Performance and application perspective", *Seventh International Symposium on Natural Language Processing* (SNLP 2007), December 13 - 15, 2007.

APPENDIX

A.1 Source Codes

The source codes used for this project are given below sorted by type of operation. Some of the variable names have been denoted as “1” for security reasons. The common codes in Flash ActionScript has not been given.

Main file to contain Flash - index.php

```
1.    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
      "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2.    <html xmlns="http://www.w3.org/1999/xhtml">
3.        <head>
4.            <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
5.            <meta name="verify-v1" content="1" >
6.                <title>Interactive Language Tools - LABS.com.bd</title>
7.                <link href="styles.css" rel="stylesheet" type="text/css" />
8.                <script language="javascript"> AC_FL_RunContent = 0; </script>
9.                <script language="javascript"> DetectFlashVer = 0; </script>
10.               <script src="AC_RunActiveContent.js" language="javascript"></script>
11.               <script language="JavaScript" type="text/javascript">
12.                   <!--
13.                   // ---------------------------------------------------------------
      -------------
14.                   // Globals
15.                   // Major version of Flash required
16.                   var requiredMajorVersion = 8;
17.                   // Minor version of Flash required
18.                   var requiredMinorVersion = 0;
19.                   // Revision of Flash required
20.                   var requiredRevision = 24;
21.                   // ---------------------------------------------------------------
      -------------
22.                   // -->
23.               </script>
24.       </head>
25.       <body style="overflow: hidden">
26.   <?php
27.   if ($_REQUEST['uid']) {
28.    $con = mysql_connect('localhost', 1, '1');
29.    if (!$con) {
30.           die('Could not connect: ' . mysql_error());
31.    }
32.    mysql_select_db("1", $con);
33.    $sql = "INSERT INTO fb_language_tools_logs (id, timestamp, uid, first_name,
      middle_name, last_name, sex) VALUES (NULL, UTC_TIMESTAMP(), '".$_REQUEST['uid']."',
      '".$_REQUEST['first_name']."', '".$_REQUEST['middle_name']."',
      '".$_REQUEST['last_name']."', '".$_REQUEST['sex']."');";
34.           mysql_query($sql);
35.    }
36.    mysql_close($con);
37.   }
38.   ?>
39.           <div id="SkipIntroLink">
40.               <iframe src="loadRemoteAudio.php" height="1" width="1" frameborder="0"
      id="loadSoundPage" name="loadSoundPage" scrolling="no" hspace="0" vspace="0"
      ></iframe>
41.           </div>
42.           <!-- All rights reserved by SCULPTORLABS -->
43.           <!-------------- LABS.com.bd -------------->
44.           <script language="JavaScript" type="text/javascript">
45.               <!--
```

```
46.               if (AC_FL_RunContent == 0 || DetectFlashVer == 0) {
47.                   alert("This page requires AC_RunActiveContent.js.");
48.               } else {
49.                   var hasRightVersion = DetectFlashVer(requiredMajorVersion,
      requiredMinorVersion, requiredRevision);
50.                   if(hasRightVersion) {  // if we've detected an acceptable version
51.                       // embed the flash movie
52.                       AC_FL_RunContent(
53.                       'codebase',
      'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=8,0,24
      ,0',
54.                       'width', '100%',
55.                       'height', '100%',
56.                       'src', 'main',
57.                       'quality', 'best',
58.                       'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
59.                       'align', 'middle',
60.                       'play', 'true',
61.                       'loop', 'true',
62.                       'scale', 'showall',
63.                       'wmode', 'transparent',
64.                       'devicefont', 'false',
65.                       'id', 'intro_v3',
66.                       'bgcolor', '#000000',
67.                       'name', 'main',
68.                       'menu', 'true',
69.                       'allowScriptAccess','always',
70.                       'allowFullScreen','true',
71.                       'movie', 'main',
72.                       'salign', ''
73.                   ); //end AC code
74.                   } else {  // flash is too old or we can't detect the plugin
75.                       var alternateContent = '<table width="100%" height="100%"
      border="0" cellpadding="0" cellspacing="0" bgcolor="#000000"
      id="HTML_Global"><tr><td align="center" valign="middle"><table width="100%"
      border="0" cellpadding="0" cellspacing="0" id="HTML_Viewer"><tr><td height="410"
      align="center" valign="middle" background="images/intro_bg_html.jpg"><table
      width="340" border="1" cellpadding="20" cellspacing="0" bordercolor="#111111"
      bgcolor="#070707"
      id="HTML_Message"><tr><td></td></tr></table></td></tr></table></td></tr></table>';
76.                       document.write(alternateContent);  // insert non-flash content
77.                   }
78.               }
79.               // -->
80.           </script>
81.           <noscript>
82.               <table width="100%" height="100%" border="0" cellpadding="0"
      cellspacing="0" bgcolor="#000000" id="HTML_Global">
83.                   <tr>
84.                       <td align="center" valign="middle"><table width="100%"
      border="0" cellpadding="0" cellspacing="0" id="HTML_Viewer">
85.                                 <tr>
86.                                     <td height="410" align="center" valign="middle"
      background="../images/intro_bg_html.jpg"><table width="340" border="1"
      cellpadding="20" cellspacing="0" bordercolor="#111111" bgcolor="#070707"
      id="HTML_Message">
87.                                             <tr>
88.                                                 <td><div align="justify">
89.
      <p><a href="http://labs.com.bd" target="_parent" title="Click here to visit the
      official site of SCULPTORLABS.">SCULPTORLABS</a></p>
90.                                                                 </div></td>
91.                                             </tr>
92.                                         </table></td>
93.                                 </tr>
94.                             </table></td>
95.                   </tr>
96.               </table>
97.           </noscript>
98.           <script type="text/javascript">
99.               var gaJsHost = (("https:" == document.location.protocol) ?
      "https://ssl." : "http://www.");
100.              document.write(unescape("%3Cscript src='" + gaJsHost + "google-
      analytics.com/ga.js' type='text/javascript'%3E%3C/script%3E"));
101.          </script>
```

```
102.          <script type="text/javascript">
103.              try {
104.                  var pageTracker = _gat._getTracker("UA-1500776-6");
105.                  pageTracker._trackPageview();
106.              } catch(err) {}</script>
107.          <script type="text/javascript">
108.              var sc_project=4470472;
109.              var sc_invisible=1;
110.              var sc_partition=55;
111.              var sc_click_stat=1;
112.              var sc_security="5fe7b0ee";
113.          </script>
114.          <script type="text/javascript"
      src="http://www.statcounter.com/counter/counter.js"></script><noscript><div
      class="statcounter"><a href="http://www.statcounter.com/wordpress.org/"
      target="_blank"><img class="statcounter"
      src="http://c.statcounter.com/4470472/0/5fe7b0ee/1/"></a></div></noscript>
115.      </body>
116.  </html>
```

JavaScript Flash Loader - AC_RunActiveContent.js

```
1.    //v1.7
2.    // Flash Player Version Detection
3.    // Detect Client Browser type
4.    // Copyright 2005-2007 Adobe Systems Incorporated.  All rights reserved.
5.    var isIE  = (navigator.appVersion.indexOf("MSIE") != -1) ? true : false;
6.    var isWin = (navigator.appVersion.toLowerCase().indexOf("win") != -1) ? true :
      false;
7.    var isOpera = (navigator.userAgent.indexOf("Opera") != -1) ? true : false;
8.
9.    function ControlVersion()
10.   {
11.    var version;
12.    var axo;
13.    var e;
14.
15.    // NOTE : new ActiveXObject(strFoo) throws an exception if strFoo isn't in the
      registry
16.
17.    try {
18.            // version will be set for 7.X or greater players
19.            axo = new ActiveXObject("ShockwaveFlash.ShockwaveFlash.7");
20.            version = axo.GetVariable("$version");
21.    } catch (e) {
22.    }
23.
24.    if (!version)
25.    {
26.            try {
27.                    // version will be set for 6.X players only
28.                    axo = new ActiveXObject("ShockwaveFlash.ShockwaveFlash.6");
29.
30.                    // installed player is some revision of 6.0
31.                    // GetVariable("$version") crashes for versions 6.0.22 through
      6.0.29,
32.                    // so we have to be careful.
33.
34.                    // default to the first public version
35.                    version = "WIN 6,0,21,0";
36.
37.                    // throws if AllowScripAccess does not exist (introduced in 6.0r47)

38.                    axo.AllowScriptAccess = "always";
39.
40.                    // safe to call for 6.0r47 or greater
41.                    version = axo.GetVariable("$version");
42.
43.            } catch (e) {
44.            }
45.    }
46.
47.    if (!version)
48.    {
```

```
49.             try {
50.                     // version will be set for 4.X or 5.X player
51.                     axo = new ActiveXObject("ShockwaveFlash.ShockwaveFlash.3");
52.                     version = axo.GetVariable("$version");
53.             } catch (e) {
54.             }
55.     }
56.
57.     if (!version)
58.     {
59.             try {
60.                     // version will be set for 3.X player
61.                     axo = new ActiveXObject("ShockwaveFlash.ShockwaveFlash.3");
62.                     version = "WIN 3,0,18,0";
63.             } catch (e) {
64.             }
65.     }
66.
67.     if (!version)
68.     {
69.             try {
70.                     // version will be set for 2.X player
71.                     axo = new ActiveXObject("ShockwaveFlash.ShockwaveFlash");
72.                     version = "WIN 2,0,0,11";
73.             } catch (e) {
74.                     version = -1;
75.             }
76.     }
77.
78.     return version;
79.    }
80.
81.    // JavaScript helper required to detect Flash Player PlugIn version information
82.    function GetSwfVer(){
83.     // NS/Opera version >= 3 check for Flash plugin in plugin array
84.     var flashVer = -1;
85.
86.     if (navigator.plugins != null && navigator.plugins.length > 0) {
87.             if (navigator.plugins["Shockwave Flash 2.0"] ||
       navigator.plugins["Shockwave Flash"]) {
88.                     var swVer2 = navigator.plugins["Shockwave Flash 2.0"] ? " 2.0" :
       "";
89.                     var flashDescription = navigator.plugins["Shockwave Flash" +
       swVer2].description;
90.                     var descArray = flashDescription.split(" ");
91.                     var tempArrayMajor = descArray[2].split(".");
92.                     var versionMajor = tempArrayMajor[0];
93.                     var versionMinor = tempArrayMajor[1];
94.                     var versionRevision = descArray[3];
95.                     if (versionRevision == "") {
96.                             versionRevision = descArray[4];
97.                     }
98.                     if (versionRevision[0] == "d") {
99.                             versionRevision = versionRevision.substring(1);
100.                    } else if (versionRevision[0] == "r") {
101.                            versionRevision = versionRevision.substring(1);
102.                            if (versionRevision.indexOf("d") > 0) {
103.                                    versionRevision = versionRevision.substring(0,
       versionRevision.indexOf("d"));
104.                            }
105.                    }
106.                    var flashVer = versionMajor + "." + versionMinor + "." +
       versionRevision;
107.            }
108.    }
109.    // MSN/WebTV 2.6 supports Flash 4
110.    else if (navigator.userAgent.toLowerCase().indexOf("webtv/2.6") != -1) flashVer =
       4;
111.    // WebTV 2.5 supports Flash 3
112.    else if (navigator.userAgent.toLowerCase().indexOf("webtv/2.5") != -1) flashVer =
       3;
113.    // older WebTV supports Flash 2
114.    else if (navigator.userAgent.toLowerCase().indexOf("webtv") != -1) flashVer = 2;
115.    else if ( isIE && isWin && !isOpera ) {
116.            flashVer = ControlVersion();
```

```
 }
 return flashVer;
}

// When called with reqMajorVer, reqMinorVer, reqRevision returns true if that version or greater is available
function DetectFlashVer(reqMajorVer, reqMinorVer, reqRevision)
{
 versionStr = GetSwfVer();
 if (versionStr == -1 ) {
        return false;
 } else if (versionStr != 0) {
        if(isIE && isWin && !isOpera) {
               // Given "WIN 2,0,0,11"
               tempArray         = versionStr.split(" ");   // ["WIN", "2,0,0,11"]
               tempString        = tempArray[1];                    // "2,0,0,11"
               versionArray      = tempString.split(",");   // ['2', '0', '0', '11']
        } else {
               versionArray      = versionStr.split(".");
        }
        var versionMajor      = versionArray[0];
        var versionMinor      = versionArray[1];
        var versionRevision   = versionArray[2];

        // is the major.revision >= requested major.revision AND the minor version >= requested minor
        if (versionMajor > parseFloat(reqMajorVer)) {
               return true;
        } else if (versionMajor == parseFloat(reqMajorVer)) {
               if (versionMinor > parseFloat(reqMinorVer))
                      return true;
               else if (versionMinor == parseFloat(reqMinorVer)) {
                      if (versionRevision >= parseFloat(reqRevision))
                             return true;
               }
        }
        return false;
 }
}

function AC_AddExtension(src, ext)
{
  if (src.indexOf('?') != -1)
    return src.replace(/\?/, ext+'?');
  else
    return src + ext;
}

function AC_Generateobj(objAttrs, params, embedAttrs)
{
  var str = '';
  if (isIE && isWin && !isOpera)
  {
    str += '<object ';
    for (var i in objAttrs)
    {
      str += i + '="' + objAttrs[i] + '" ';
    }
    str += '>';
    for (var i in params)
    {
      str += '<param name="' + i + '" value="' + params[i] + '" /> ';
    }
    str += '</object>';
  }
  else
  {
    str += '<embed ';
    for (var i in embedAttrs)
    {
      str += i + '="' + embedAttrs[i] + '" ';
    }
    str += '> </embed>';
  }
```

```
189.
190.     document.write(str);
191.   }
192.
193.   function AC_FL_RunContent(){
194.     var ret =
195.       AC_GetArgs
196.       (  arguments, ".swf", "movie", "clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
197.        , "application/x-shockwave-flash"
198.       );
199.     AC_Generateobj(ret.objAttrs, ret.params, ret.embedAttrs);
200.   }
201.
202.   function AC_SW_RunContent(){
203.     var ret =
204.       AC_GetArgs
205.       (  arguments, ".dcr", "src", "clsid:166B1BCA-3F9C-11CF-8075-444553540000"
206.        , null
207.       );
208.     AC_Generateobj(ret.objAttrs, ret.params, ret.embedAttrs);
209.   }
210.
211.   function AC_GetArgs(args, ext, srcParamName, classid, mimeType){
212.     var ret = new Object();
213.     ret.embedAttrs = new Object();
214.     ret.params = new Object();
215.     ret.objAttrs = new Object();
216.     for (var i=0; i < args.length; i=i+2){
217.       var currArg = args[i].toLowerCase();
218.
219.       switch (currArg){
220.         case "classid":
221.           break;
222.         case "pluginspage":
223.           ret.embedAttrs[args[i]] = args[i+1];
224.           break;
225.         case "src":
226.         case "movie":
227.           args[i+1] = AC_AddExtension(args[i+1], ext);
228.           ret.embedAttrs["src"] = args[i+1];
229.           ret.params[srcParamName] = args[i+1];
230.           break;
231.         case "onafterupdate":
232.         case "onbeforeupdate":
233.         case "onblur":
234.         case "oncellchange":
235.         case "onclick":
236.         case "ondblclick":
237.         case "ondrag":
238.         case "ondragend":
239.         case "ondragenter":
240.         case "ondragleave":
241.         case "ondragover":
242.         case "ondrop":
243.         case "onfinish":
244.         case "onfocus":
245.         case "onhelp":
246.         case "onmousedown":
247.         case "onmouseup":
248.         case "onmouseover":
249.         case "onmousemove":
250.         case "onmouseout":
251.         case "onkeypress":
252.         case "onkeydown":
253.         case "onkeyup":
254.         case "onload":
255.         case "onlosecapture":
256.         case "onpropertychange":
257.         case "onreadystatechange":
258.         case "onrowsdelete":
259.         case "onrowenter":
260.         case "onrowexit":
261.         case "onrowsinserted":
262.         case "onstart":
263.         case "onscroll":
```

```
        case "onbeforeeditfocus":
        case "onactivate":
        case "onbeforedeactivate":
        case "ondeactivate":
        case "type":
        case "codebase":
        case "id":
          ret.objAttrs[args[i]] = args[i+1];
          break;
        case "width":
        case "height":
        case "align":
        case "vspace":
        case "hspace":
        case "class":
        case "title":
        case "accesskey":
        case "name":
        case "tabindex":
          ret.embedAttrs[args[i]] = ret.objAttrs[args[i]] = args[i+1];
          break;
        default:
          ret.embedAttrs[args[i]] = ret.params[args[i]] = args[i+1];
      }
    }
    ret.objAttrs["classid"] = classid;
    if (mimeType) ret.embedAttrs["type"] = mimeType;
    return ret;
  }
```

Facebook Callback Page – fb.php

```
     <fb:fbml>
        <?php
        if (true) {

             // Get these from http://developers.facebook.com
             $api_key = 1;
             $secret  = 1;

             // Names and links
             $app_name = "Language Tools for Bangla and English";
             $app_url = "language_tools"; // Assumes application is at this
             $invite_href = "fb.php"; // Rename this as needed

             require_once 'php/facebook.php';

             $facebook = new Facebook($api_key, $secret);
             $facebook->require_frame();
             $user = $facebook->require_login();

            try {
                $api = new Facebook($api_key, $secret);

                // If a session key is passed, then grab it
                if (!empty($_REQUEST)) {
                    if (!empty($_REQUEST["fb_sig_session_key"])) {
                        $api->sessionKey = $_REQUEST["fb_sig_session_key"];
                    }
                }

                if (!empty($_REQUEST)) {
                    if (!empty($_REQUEST["fb_sig_canvas_user"])) {
                        $isCanvasUser = true;
                        $uid = $_REQUEST["fb_sig_canvas_user"];
                    } else { $isCanvasUser = false; }
                    if (!empty($_REQUEST["fb_sig_user"])) {
                        $isAddedUser = true;
                        $uid = $_REQUEST["fb_sig_user"];
                    } else { $isAddedUser = false; }
                    if ($isCanvasUser || $isAddedUser) {
                    // Say Hi
                                    $user_details = $api->api_client-
      >users_getInfo($uid, 'last_name, first_name, middle_name, sex');
```

```
42.                         echo "<table width=\"100%\" border=\"0\" align=\"center\"
    cellpadding=\"0\" cellspacing=\"0\" id=\"header\">\n<tr>\n<td height=\"35\"
    width=\"220\" align=\"left\" valign=\"middle\" bgcolor=\"#FFFFFF\"
    style=\"color:#333;\"><img src=\"http://labs.com.bd/voice/Logo.png\" width=\"200\"
    height=\"50\" /></td>\n<td align=\"right\" valign=\"middle\" bgcolor=\"#FFFFFF\"
    style=\"color:#333;\">";
43.                         echo "Hi <strong>" . $user_details[0]['first_name'] .
    "</strong>! Welcome to Language Tools :) - Would you like to visit the <strong><a
    target=\"_parent\"
    href=\"http://www.facebook.com/apps/application.php?id=112003278821867\"
    style=\"font-size:12px; text-decoration:none;\">Application's Page?</a></strong>";
44.                         echo "</td>\n</tr>\n</table>";
45.                     }
46.                 }
47.             } catch (Services_Facebook_Exception $e) {
48.                 echo $e->getLastCall() . "<br />";
49.                 echo $e->getMessage();
50.             }
51.         }
52.         ?>
53.         <fb:iframe
54.          name="flashRemote"
55.             frameborder="0"
56.             scrolling="no"
57.             src="http://labs.com.bd/voice/index.php?uid=<? echo $uid; ?>&first_name=<?
    echo $user_details[0]['first_name']; ?>&middle_name=<? echo
    $user_details[0]['middle_name']; ?>&last_name=<? echo
    $user_details[0]['last_name']; ?>&sex=<? echo $user_details[0]['sex']; ?>"
58.             wmode="window"
59.             scale="showall"
60.             width="760"
61.             height="450"
62.             />
63.
64.         <?php
65.
66.         if(isset($_POST["ids"])) {
67.             echo "<center>Thank you for inviting ".sizeof($_POST["ids"])." of your
    friends on <b><a
    href=\"http://apps.facebook.com/".$app_url."/\">".$app_name."</a></b>.<br><br>\n";
68.             echo "<h2><a href=\"http://apps.facebook.com/".$app_url."/\">Click here to
    invite some more friends to ".$app_name."</a>.</h2></center>";
69.         } else {
70.             // Retrieve array of friends who've already authorized the app.
71.                 $fql = 'SELECT uid FROM user WHERE uid IN (SELECT uid2 FROM friend
    WHERE uid1='.$user.') AND is_app_user = 1';
72.                 $_friends = $facebook->api_client->fql_query($fql);
73.
74.                 // Extract the user ID's returned in the FQL request into a new array.
75.                 $friends = array();
76.                 if (is_array($_friends) && count($_friends)) {
77.                     foreach ($_friends as $friend) {
78.                         $friends[] = $friend['uid'];
79.                     }
80.                 }
81.
82.                 // Convert the array of friends into a comma-delimeted string.
83.                 $friends = implode(',', $friends);
84.
85.                 // Prepare the invitation text that all invited users will receive.
86.                 $content =
87.                     "<fb:name uid=\"".$user."\" firstnameonly=\"true\"
    shownetwork=\"false\"/> has started using <a
    href=\"http://apps.facebook.com/".$app_url."/\">".$app_name."</a> and thought it's
    so cool even you should try it out!\n".
88.                     "<fb:req-choice url=\"".$facebook->get_add_url()."\" label=\"Put
    ".$app_name." on your profile\"/>";
89.
90.                 ?>
91.         <fb:request-form
92.             action="<? echo $invite_href; ?>"
93.             method="post"
94.             type="<? echo $app_name; ?>"
95.             content="<? echo htmlentities($content,ENT_COMPAT,'UTF-8'); ?>">
96.
```

```
97.           <fb:multi-friend-selector
98.               actiontext="Here are your friends who don't have <? echo $app_name; ?>
      yet. Invite whoever you want - it's free and open source!"
99.               exclude_ids="<? echo $friends; ?>" />
100.      </fb:request-form>
101.          <?php
102.
103.          }
104.
105.      ?>
106.  </fb:fbml>
```

getRemoteAudio.php

```
1.      <?php
2.      $audio = $_POST[1];
3.      $copydir = "files/";
4.      $file_base = rand(1000, 1000000) . ".wav";
5.      $data = file_get_contents($audio);
6.      $file = fopen($copydir . $file_base, "w+");
7.      fputs($file, $data);
8.      fclose($file);
9.      echo '&playAudioFile=' . $file_base . '&';
10.     ?>
```

getRemoteContent.php

```
1.      <?php
2.      $url = "http://crblp.bracu.ac.bd/demo/tts/index.php";
3.      $postdata['callFromRemoteLocationForEnglishLanguage'] =
        $_POST['callFromRemoteLocationForEnglishLanguage'];
4.      $postdata['speech'] = $_POST['speech'];
5.      $postdata['volume_scale'] = $_POST['volume_scale'];
6.      $postdata['callFromRemoteLocation'] = "true";
7.      $postdata['make_audio'] = " Listen ";
8.      $useragent= "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)" ;
9.
10.     $ch = curl_init();
11.     //set some cookie details up (depending on the site)
12.     //curl_setopt($ch, CURLOPT_COOKIEJAR, "cookies.txt");
13.     //curl_setopt($ch, CURLOPT_COOKIEFILE, "cookies.txt");
14.     curl_setopt($ch, CURLOPT_USERAGENT, $useragent); //set our user agent
15.     curl_setopt($ch, CURLOPT_POST, 1); //set how many paramaters
16.     curl_setopt($ch, CURLOPT_URL, $url); //set the url we want to use
17.     curl_setopt($ch, CURLOPT_POSTFIELDS, $postdata); //set data to post
18.
19.
20.     $result= curl_exec ($ch); //execute and get the results
21.     curl_close ($ch);
22.     print $result; //display the reuslt
23.     ?>
```

getRemoteText.php

```
1.      <?php
2.      $url = "http://crblp.bracu.ac.bd/demo/bocr/ocr_script_en_bn.php";
3.      $postdata['remoteFileName'] = $_POST['remoteFileName'];
4.      $postdata['remoteFileURL'] = $_POST['remoteFileURL'];
5.      $postdata['remoteLanguage'] = $_POST['remoteLanguage'];
6.      $useragent= "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)" ;
7.
8.      $ch = curl_init();
9.      //set some cookie details up (depending on the site)
10.     //curl_setopt($ch, CURLOPT_COOKIEJAR, "cookies.txt");
11.     //curl_setopt($ch, CURLOPT_COOKIEFILE, "cookies.txt");
12.     curl_setopt($ch, CURLOPT_USERAGENT, $useragent); //set our user agent
13.     curl_setopt($ch, CURLOPT_POST, 1); //set how many paramaters
14.     curl_setopt($ch, CURLOPT_URL, $url); //set the url we want to use
15.     curl_setopt($ch, CURLOPT_POSTFIELDS, $postdata); //set data to post
16.
17.     $result= curl_exec ($ch); //execute and get the results
18.     curl_close ($ch);
19.     print $result; //display the reuslt
```

```
20.     ?>
```

getWord.php

```
1.      <?php
2.      header('Content-Type: text/html; charset=utf-8');
3.      echo '<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"><html
        xmlns="http://www.w3.org/1999/xhtml"><head><meta http-equiv="Content-Type"
        content="text/html; charset=utf-8" /></head><body>';
4.      if ($_REQUEST['searchWord']) {
5.              $con = mysql_connect('localhost', 1_1, 1);
6.              if (!$con) {
7.                      die('Could not connect: ' . mysql_error());
8.              }
9.              mysql_select_db("1_1", $con);
10.
11.             mysql_query("SET NAMES 'utf8' COLLATE 'utf8_unicode_ci'");
12.
13.             $result = mysql_query("SELECT * FROM language_tools_wordnet_db WHERE
        word='".$_REQUEST[1]."';");
14.
15.             $output = "";
16.             $counter = 1;
17.
18.             if (!$result) {
19.             die('Error in query: ' . mysql_error());
20.                     $output = "0";
21.             }
22.             else {
23.                     while($row = mysql_fetch_array($result)) {
24.                             //$row[0]; // ID
25.                             $output .= $counter . ". <b><i>" . $row[1] .
        "</i></b>:<br>"; // Word
26.                             $output .= $row[2] . "<br>"; // Type
27.                             $output .= $row[3] . "<br><i>"; // Meaning
28.                             $output .= $row[4] . "</i><br>"; // Example
29.                             $output .= $row[5] . "<br><br>"; // Related
30.                             $counter++;
31.                     }
32.             }
33.             echo '&outputWordMeaning='.$output.'&';
34.             mysql_close($con);
35.     }
36.     echo '</body></html>';
37.     ?>
```

loadFile.php

```
1.      <?php
2.      $file = $_POST['fileURL'];
3.      $f = fopen($file, "r");
4.      print '&fileContent=';
5.      while ( $line = fgets($f, 1000) ) {
6.              print $line;
7.      }
8.      print '&';
9.      ?>
```

loadRemoteAudio.php

```
1.      <?php
2.      $file_base = $_REQUEST['file_base'];
3.      if ($file_base) {
4.              echo '<html><head><title>Audio Loader</title></head><body>';
5.              echo '<embed src="files/' . $file_base . '" autostart="true" loop="false"
        width="100" height="10"></embed>';
6.              echo '<noembed>';
7.              echo '<bgsound src="files/' . $file_base . '" loop="1">';
8.              echo '</noembed></body></html>';
9.      }
10.     ?>
```

loadURL.php

```
1.      <?php
2.      $curl_handle=curl_init();
3.      //This is the URL you would like the content grabbed from
4.      curl_setopt($curl_handle,CURLOPT_URL,$_POST['loadURL']);
5.      //This is the amount of time in seconds until it times out, this is useful if the
        server you are requesting data from is down. This way you can offer a "sorry page"
6.      curl_setopt($curl_handle,CURLOPT_CONNECTTIMEOUT,2);
7.
8.      curl_setopt($curl_handle,CURLOPT_RETURNTRANSFER,1);
9.      $buffer = curl_exec($curl_handle);
10.     //This Keeps everything running smoothly
11.     curl_close($curl_handle);
12.
13.     // Change the message bellow as you wish, please keep in mind you must have your
        message within the " " Quotes.
14.     if (empty($buffer)) {
15.             print '&pageContent=দুঃখিত, এই পাতাটি আপাতত দেখানো যাচ্ছে না&';
16.     }
17.     else {
18.             print '&pageContent='.$buffer.'&';
19.     }
20.     ?>
```

ocr_script_en_bn.php

```
1.      <?php
2.      // POST data from LABS.com.bd
3.      $languageOption=trim($_POST[1]);
4.      $image = $_POST[1];
5.      // Copy the remote PNG file to CRBLP server
6.      $randnum = rand(1000, 1000000);
7.      $uploadfile = '/opt/lampp/htdocs/bocr/upload/'.$randnum.'.png';
8.      $data = file_get_contents($image);
9.      $file = fopen($uploadfile, "w+");
10.     fputs($file, $data);
11.     fclose($file);
12.
13.     // Setting directories for temporary files
14.     $tmpfile = '/opt/lampp/htdocs/bocr/upload/'.'t'.$randnum.'.png';
15.     $rimg = '/opt/lampp/htdocs/bocr/upload/'.$randnum.'.tif';
16.     $pbm = '/opt/lampp/htdocs/bocr/upload/'.$randnum.'.pbm';
17.     $txt = '/opt/lampp/htdocs/bocr/upload/'.$randnum;
18.     $bocr = '/opt/lampp/htdocs/bocr/bin/bocrdriver';
19.     $gocr = '/opt/lampp/htdocs/bocr/bin/gocr';
20.     $jar = '/opt/lampp/htdocs/bocr/bin/PostProcessor.jar';
21.     $ppout = '/opt/lampp/htdocs/bocr/upload/out.txt';
22.
23.     // If Bangla, then execute bocr
24.     if ($languageOption=='BN') {
25.             exec("$bocr $uploadfile $tmpfile");
26.             exec("convert $tmpfile $rimg");
27.             exec("tesseract $rimg $txt -l ban");
28.             $txt=$txt.'.txt';
29.             exec("java -jar $jar $txt $ppout");
30.             $fh = fopen($ppout, 'r');
31.             $theData = fread($fh, filesize($ppout));
32.
33.             fclose($fh);
34.
35.             unlink($uploadfile);
36.             unlink($tmpfile);
37.             unlink($rimg);
38.             unlink($txt);
39.             unlink($ppout);
40.             echo "&convertedText=".$theNewData."&";
41.     }
42.     else { // If English, then execute gocr
43.             exec("convert $uploadfile $pbm");
44.             exec("$gocr $pbm", $theNewData);
45.
46.             unlink($uploadfile);
47.             unlink($pbm);
```

```
48.             echo "&convertedText=".$theNewData."&";
49.     }
50.     ?>
```

uploadImg.php

```
1.      <?php
2.      move_uploaded_file($_FILES['1'][1], "files/".$_FILES[1][1]);
3.      chmod("files/".$_FILES[1][1], 0777);
4.      ?>
```

uploadTxt.php

```
1.      <?php
2.      move_uploaded_file($_FILES['1'][1], "txt_file/".$_FILES[1][1]);
3.      chmod("files/".$_FILES[1][1], 0777);
4.      ?>
```

Adobe Flash Source Code – main.swf

```
1.      Frame 1
2.      var lan:String = "EN";
3.      var fontEmbedding:String =
        "।ঃ ং ঃ অআইঈউঊঋঌএঐওঔকখগঘঙচছজঝঞটঠডঢণতথদধনপফবভমযরলশষসহাাি ু ূ ৃ ৄ ে ৈ ো ৌ ্ ড় ঢ় য় ৗ ০১২৩৪৫৬
        ৭৮৯৲৳৴৵৶৷৸৹৺৻";
4.      var pageNow:Number = 1;
5.      var mainMenu:ContextMenu = new ContextMenu();
6.      mainMenu.hideBuiltInItems();
7.      _root.menu = mainMenu;
8.      function deadClick() {
9.      }
10.     function contactSupport() {
11.             getURL("http://labs.com.bd/contact.html", "_blank");
12.     }
13.     var SCULPTORMenu:ContextMenu = new ContextMenu();
14.     SCULPTORMenu.hideBuiltInItems();
15.     var copyrightNotice:ContextMenuItem = new ContextMenuItem("2010 © Shiblee Imtiaz
        Hasan", deadClick);
16.     var contactLink:ContextMenuItem = new ContextMenuItem("Contact Support",
        contactSupport);
17.     copyrightNotice.separatorBefore = true;
18.     SCULPTORMenu.customItems.push(contactLink,copyrightNotice);
19.     _root.menu = SCULPTORMenu;
20.     fscommand("fullscreen", "true");
21.     fscommand("allowscale", "true");
22.     function toggleFullScreen(){
23.       //if normal size, go to fullscreen, else go to normal size
24.       if(Stage["displayState"]=="normal"){
25.         Stage["displayState"]="fullScreen";
26.       }else{
27.         Stage["displayState"]="normal";
28.       }
29.     }
30.     function doFullScreen(){
31.         Stage["displayState"]="fullScreen";
32.     }
33.     //Create a listener for each time the Stage is resized
34.     var resizeListener:Object = new Object();
35.     //Called each time the stage is resized
36.     resizeListener.onResize = function () {
37.       //Move the button to the center of the screen
38.       toggleFullScreenButton._x=Stage.width/2;
39.       toggleFullScreenButton._y=Stage.height/2;
40.     }
41.     //Add the listener to Stage
42.     Stage.addListener(resizeListener);
43.     var clicked:Boolean = false;
44.     removeChild(this["font_embedder"]);
45.     play();
```

```
46.
47.     _root.clicked = false;
48.
49.     tellTarget("currentPage") {
50.             gotoAndStop(_root.pageNow);
51.     }
52.     tellTarget("pageTitle") {
53.             gotoAndStop(_root.pageNow);
54.     }
55.     _root.clicked = false;
56.
57.     stop();
58.     tellTarget("currentPage") {
59.             gotoAndStop(_root.pageNow);
60.     }
61.     tellTarget("pageTitle") {
62.             gotoAndStop(_root.pageNow);
63.     }
64.     if (_root.lan == "EN") {
65.             currentPage.mainContent.gotoAndStop(3);
66.     }
67.     else {
68.             currentPage.mainContent.gotoAndStop(2);
69.     }
70.
71.     tellTarget("currentPage") {
72.             gotoAndStop(_root.pageNow);
73.     }
74.     tellTarget("nextPage") {
75.             if (_root.pageNow==5) {
76.                     gotoAndStop(1);
77.             }
78.             else {
79.                     gotoAndStop(_root.pageNow+1);
80.             }
81.     }
82.
83.     tellTarget("pageTitle") {
84.             if (_root.pageNow==5) {
85.                     gotoAndStop(1);
86.             }
87.             else {
88.                     gotoAndStop(_root.pageNow+1);
89.             }
90.     }
91.
92.     if (_root.pageNow==5) {
93.             _root.pageNow = 1;
94.     }
95.     else {
96.             _root.pageNow++;
97.     }
98.     gotoAndStop("page");
99.
100.    tellTarget("currentPage") {
101.            gotoAndStop(_root.pageNow);
102.    }
103.    tellTarget("prevPage") {
104.            if (_root.pageNow==1) {
105.                    gotoAndStop(5);
106.            }
107.            else {
108.                    gotoAndStop(_root.pageNow-1);
109.            }
110.    }
111.    tellTarget("pageTitle") {
112.            gotoAndStop(_root.pageNow);
113.    }
114.
115.    tellTarget("pageTitle") {
116.            if (_root.pageNow==1) {
117.                    gotoAndStop(5);
118.            }
119.            else {
120.                    gotoAndStop(_root.pageNow-1);
```

```
121.            }
122.    }
123.
124.    if (_root.pageNow==1) {
125.            _root.pageNow = 5;
126.    }
127.    else {
128.            _root.pageNow--;
129.    }
130.    gotoAndStop("page");
131.
132.    on(release){
133.      _root.toggleFullScreen();
134.    }
135.
136.    on(release){
137.      _root.gotoAndPlay("menu");
138.    }
139.
140.    stop();
141.    if (_root.lan == "BN") {
142.            gotoAndStop(2);
143.    }
144.
145.    stop();
146.    if (_root.lan == "EN") {
147.            gotoAndStop(1);
148.    }
149.
150.    stop();
151.    //Allow this domain
152.    System.security.allowDomain("http://localhost/");
153.    System.security.allowDomain("http://crblp.bracu.ac.bd/");
154.    System.security.allowDomain("http://www.bracu.ac.bd/");
155.    System.security.allowDomain("http://bracu.ac.bd/");
156.    import flash.net.FileReference;
157.    // The listener object listens for FileReference events.
158.    var listener:Object = new Object();
159.
160.    // When the user selects a file, the onSelect() method is called, and
161.    // passed a reference to the FileReference object.
162.    listener.onSelect = function(selectedFile:FileReference):Void {
163.      //clean statusArea and details area
164.      statusArea.text = details.text = ""
165.      // Flash is attempting to upload the image.
166.      statusArea.text += "Attempting to upload " + selectedFile.name + "\n";
167.      // Upload the file to the PHP script on the server.
168.      selectedFile.upload("uploadTxt.php");
169.    };
170.
171.    // the file is starting to upload.
172.    listener.onOpen = function(selectedFile:FileReference):Void {
173.      statusArea.text += "Uploading " + selectedFile.name + "\n";
174.    };
175.    //Possible file upload errors
176.    listener.onHTTPError = function(file:FileReference, httpError:Number):Void {
177.            details.text = "HTTPError number: "+httpError +"\nFile: "+ file.name;
178.    }
179.
180.    listener.onIOError = function(file:FileReference):Void {
181.            details.text = "IOError: "+ file.name;
182.    }
183.
184.    listener.onSecurityError = function(file:FileReference, errorString:String):Void {
185.            details.text = "SecurityError: "+SecurityError+"\nFile: "+ file.name;
186.    }
187.
188.    // the file has uploaded
189.    listener.onComplete = function(selectedFile:FileReference):Void {
190.      // Notify the user that Flash is starting to download the image.
191.      statusArea.text += "Upload finished.\nNow downloading " + selectedFile.name + "
        to text field\n";
192.      //Show file details
193.      details.text = ""
194.      for(i in selectedFile) details.text +="<b>"+i+":</b> "+selectedFile[i]+"\n"
```

```
195.      // Call the custom downloadImage() function.
196.      downloadImage(selectedFile.name);
197.    };
198.
199.    var imageFile:FileReference = new FileReference();
200.    imageFile.addListener(listener);
201.
202.    uploadBtn.onPress = uploadImage;
203.    imagePane.addEventListener("complete", imageDownloaded);
204.
205.    // Call the uploadImage() function, opens a file browser dialog.
206.    function uploadImage(event:Object):Void {
207.      imageFile.browse([{description:"Text Files (.txt)", extension:"*.txt;*.doc"}]);
208.    }
209.
210.    // If the image does not download, the event object's total property
211.    // will equal -1. In that case, display am error message
212.    function imageDownloaded(event:Object):Void {
213.      if(event.total == -1) {
214.        details.text = "Error getting the file! Please try again.";
215.      }
216.    }
217.
218.    // show uploaded image in scrollPane
219.    function downloadImage(file:Object):Void {
220.            varsHolderIn = new LoadVars();
221.            varsHolderOut = new LoadVars();
222.
223.            varsHolderIn.onLoad = function(success) {
224.                    mainText.text = varsHolderIn.fileContent;
225.                    trace(varsHolderIn.pageContent);
226.            }
227.
228.            varsHolderOut.fileURL = "files/" + file;
229.            varsHolderOut.sendAndLoad("loadFile.php", varsHolderIn, "POST");
230.    }
231.    stop();
232.
233.    stop();
234.    //Allow this domain
235.    System.security.allowDomain("http://localhost/");
236.    System.security.allowDomain("http://crblp.bracu.ac.bd/");
237.    System.security.allowDomain("http://www.bracu.ac.bd/");
238.    System.security.allowDomain("http://bracu.ac.bd/");
239.    import flash.net.FileReference;
240.    // The listener object listens for FileReference events.
241.    var listener:Object = new Object();
242.
243.    // When the user selects a file, the onSelect() method is called, and
244.    // passed a reference to the FileReference object.
245.    listener.onSelect = function(selectedFile:FileReference):Void {
246.      //clean statusArea and details area
247.      statusArea.text = details.text = ""
248.      // Flash is attempting to upload the image.
249.      statusArea.text += "Attempting to upload " + selectedFile.name + "\n";
250.      // Upload the file to the PHP script on the server.
251.      selectedFile.upload("uploadTxt.php");
252.    };
253.
254.    // the file is starting to upload.
255.    listener.onOpen = function(selectedFile:FileReference):Void {
256.      statusArea.text += "Uploading " + selectedFile.name + "\n";
257.    };
258.    //Possible file upload errors
259.    listener.onHTTPError = function(file:FileReference, httpError:Number):Void {
260.            details.text = "HTTPError number: "+httpError +"\nFile: "+ file.name;
261.    }
262.
263.    listener.onIOError = function(file:FileReference):Void {
264.            details.text = "IOError: "+ file.name;
265.    }
266.
267.    listener.onSecurityError = function(file:FileReference, errorString:String):Void {
268.            details.text = "SecurityError: "+SecurityError+"\nFile: "+ file.name;
269.    }
```

```
270.
271.    // the file has uploaded
272.    listener.onComplete = function(selectedFile:FileReference):Void {
273.      // Notify the user that Flash is starting to download the image.
274.      statusArea.text += "Upload finished.\nNow downloading " + selectedFile.name + "
        to text field\n";
275.      //Show file details
276.      details.text = ""
277.      for(i in selectedFile) details.text +="<b>"+i+":</b> "+selectedFile[i]+"\n"
278.      // Call the custom downloadImage() function.
279.      downloadImage(selectedFile.name);
280.    };
281.
282.    var imageFile:FileReference = new FileReference();
283.    imageFile.addListener(listener);
284.
285.    uploadBtn.onPress = uploadImage;
286.    imagePane.addEventListener("complete", imageDownloaded);
287.
288.    // Call the uploadImage() function, opens a file browser dialog.
289.    function uploadImage(event:Object):Void {
290.      imageFile.browse([{description:"Text Files (.txt)", extension:"*.txt;*.doc"}]);
291.    }
292.
293.    // If the image does not download, the event object's total property
294.    // will equal -1. In that case, display am error message
295.    function imageDownloaded(event:Object):Void {
296.      if(event.total == -1) {
297.        details.text = "Error getting the file! Please try again.";
298.      }
299.    }
300.
301.    // show uploaded image in scrollPane
302.    function downloadImage(file:Object):Void {
303.            varsHolderIn = new LoadVars();
304.            varsHolderOut = new LoadVars();
305.
306.            varsHolderIn.onLoad = function(success) {
307.                    mainText.text = varsHolderIn.fileContent;
308.                    trace(varsHolderIn.pageContent);
309.            }
310.
311.            varsHolderOut.fileURL = "files/" + file;
312.            varsHolderOut.sendAndLoad("loadFile.php", varsHolderIn, "POST");
313.    }
314.    stop();
315.
316.    on (press, keyPress "u") {
317.            varsHolderIn = new LoadVars();
318.            varsHolderOut = new LoadVars();
319.
320.            varsHolderIn.onLoad = function(success) {
321.                    mainText.text = varsHolderIn.pageContent;
322.                    trace(varsHolderIn.pageContent);
323.            }
324.
325.            varsHolderOut.loadURL = loadURL.text;
326.            varsHolderOut.sendAndLoad("loadURL.php", varsHolderIn, "POST");
327.    }
328.
329.    on (press, keyPress "u") {
330.            varsHolderIn = new LoadVars();
331.            varsHolderOut = new LoadVars();
332.
333.            varsHolderIn.onLoad = function(success) {
334.                    mainText.text = varsHolderIn.pageContent;
335.                    trace(varsHolderIn.pageContent);
336.            }
337.
338.            varsHolderOut.loadURL = loadURL.text;
339.            varsHolderOut.sendAndLoad("loadURL.php", varsHolderIn, "POST");
340.    }
341.
342.    //-- 0981 -------------------------------//
343.    u0981.cha.r.text = "\u0981";
```

```
344.    u0981.thisChar = "\u0981";
345.
346.    //-- 0982 ------------------------------//
347.    u0982.cha.r.text = "\u0982";
348.    u0982.thisChar = "\u0982";
349.
350.    //-- 0983 ------------------------------//
351.    u0983.cha.r.text = "\u0983";
352.    u0983.thisChar = "\u0983";
353.
354.    //-- 0984 ------------------------------//
355.    u0984.cha.r.text = "\u0984";
356.    u0984.thisChar = "\u0984";
357.
358.    //-- 0985 ------------------------------//
359.    u0985.cha.r.text = "\u0985";
360.    u0985.thisChar = "\u0985";
361.
362.    //-- 0986 ------------------------------//
363.    u0986.cha.r.text = "\u0986";
364.    u0986.thisChar = "\u0986";
365.
366.    //-- 0987 ------------------------------//
367.    u0987.cha.r.text = "\u0987";
368.    u0987.thisChar = "\u0987";
369.
370.    //-- 0988 ------------------------------//
371.    u0988.cha.r.text = "\u0988";
372.    u0988.thisChar = "\u0988";
373.
374.    //-- 0989 ------------------------------//
375.    u0989.cha.r.text = "\u0989";
376.    u0989.thisChar = "\u0989";
377.
378.    //-- 098A ------------------------------//
379.    u098A.cha.r.text = "\u098A";
380.    u098A.thisChar = "\u098A";
381.
382.    //-- 098B ------------------------------//
383.    u098B.cha.r.text = "\u098B";
384.    u098B.thisChar = "\u098B";
385.
386.    //-- 098C ------------------------------//
387.    u098C.cha.r.text = "\u098C";
388.    u098C.thisChar = "\u098C";
389.
390.    //-- 098D ------------------------------//
391.    u098D.cha.r.text = "\u098D";
392.    u098D.thisChar = "\u098D";
393.
394.    //-- 098E ------------------------------//
395.    u098E.cha.r.text = "\u098E";
396.    u098E.thisChar = "\u098E";
397.
398.    //-- 098F ------------------------------//
399.    u098F.cha.r.text = "\u098F";
400.    u098F.thisChar = "\u098F";
401.
402.    //-- 0990 ------------------------------//
403.    u0990.cha.r.text = "\u0990";
404.    u0990.thisChar = "\u0990";
405.
406.    //-- 0991 ------------------------------//
407.    u0991.cha.r.text = "\u0991";
408.    u0991.thisChar = "\u0991";
409.
410.    //-- 0992 ------------------------------//
411.    u0992.cha.r.text = "\u0992";
412.    u0992.thisChar = "\u0992";
413.
414.    //-- 0993 ------------------------------//
415.    u0993.cha.r.text = "\u0993";
416.    u0993.thisChar = "\u0993";
417.
418.    //-- 0994 ------------------------------//
```

```
419.    u0994.cha.r.text = "\u0994";
420.    u0994.thisChar = "\u0994";
421.
422.    //-- 0995 -------------------------------//
423.    u0995.cha.r.text = "\u0995";
424.    u0995.thisChar = "\u0995";
425.
426.    //-- 0996 -------------------------------//
427.    u0996.cha.r.text = "\u0996";
428.    u0996.thisChar = "\u0996";
429.
430.    //-- 0997 -------------------------------//
431.    u0997.cha.r.text = "\u0997";
432.    u0997.thisChar = "\u0997";
433.
434.    //-- 0998 -------------------------------//
435.    u0998.cha.r.text = "\u0998";
436.    u0998.thisChar = "\u0998";
437.
438.    //-- 0999 -------------------------------//
439.    u0999.cha.r.text = "\u0999";
440.    u0999.thisChar = "\u0999";
441.
442.    //-- 099A -------------------------------//
443.    u099A.cha.r.text = "\u099A";
444.    u099A.thisChar = "\u099A";
445.
446.    //-- 099B -------------------------------//
447.    u099B.cha.r.text = "\u099B";
448.    u099B.thisChar = "\u099B";
449.
450.    //-- 099C -------------------------------//
451.    u099C.cha.r.text = "\u099C";
452.    u099C.thisChar = "\u099C";
453.
454.    //-- 099D -------------------------------//
455.    u099D.cha.r.text = "\u099D";
456.    u099D.thisChar = "\u099D";
457.
458.    //-- 099E -------------------------------//
459.    u099E.cha.r.text = "\u099E";
460.    u099E.thisChar = "\u099E";
461.
462.    //-- 099F -------------------------------//
463.    u099F.cha.r.text = "\u099F";
464.    u099F.thisChar = "\u099F";
465.
466.    //-- 09A0 -------------------------------//
467.    u09A0.cha.r.text = "\u09A0";
468.    u09A0.thisChar = "\u09A0";
469.
470.    //-- 09A1 -------------------------------//
471.    u09A1.cha.r.text = "\u09A1";
472.    u09A1.thisChar = "\u09A1";
473.
474.    //-- 09A2 -------------------------------//
475.    u09A2.cha.r.text = "\u09A2";
476.    u09A2.thisChar = "\u09A2";
477.
478.    //-- 09A3 -------------------------------//
479.    u09A3.cha.r.text = "\u09A3";
480.    u09A3.thisChar = "\u09A3";
481.
482.    //-- 09A4 -------------------------------//
483.    u09A4.cha.r.text = "\u09A4";
484.    u09A4.thisChar = "\u09A4";
485.
486.    //-- 09A5 -------------------------------//
487.    u09A5.cha.r.text = "\u09A5";
488.    u09A5.thisChar = "\u09A5";
489.
490.    //-- 09A6 -------------------------------//
491.    u09A6.cha.r.text = "\u09A6";
492.    u09A6.thisChar = "\u09A6";
493.
```

```
494.    //-- 09A7 --------------------------------//
495.    u09A7.cha.r.text = "\u09A7";
496.    u09A7.thisChar = "\u09A7";
497.
498.    //-- 09A8 --------------------------------//
499.    u09A8.cha.r.text = "\u09A8";
500.    u09A8.thisChar = "\u09A8";
501.
502.    //-- 09A9 --------------------------------//
503.    u09A9.cha.r.text = "\u09A9";
504.    u09A9.thisChar = "\u09A9";
505.
506.    //-- 09AA --------------------------------//
507.    u09AA.cha.r.text = "\u09AA";
508.    u09AA.thisChar = "\u09AA";
509.
510.    //-- 09AB --------------------------------//
511.    u09AB.cha.r.text = "\u09AB";
512.    u09AB.thisChar = "\u09AB";
513.
514.    //-- 09AC --------------------------------//
515.    u09AC.cha.r.text = "\u09AC";
516.    u09AC.thisChar = "\u09AC";
517.
518.    //-- 09AD --------------------------------//
519.    u09AD.cha.r.text = "\u09AD";
520.    u09AD.thisChar = "\u09AD";
521.
522.    //-- 09AE --------------------------------//
523.    u09AE.cha.r.text = "\u09AE";
524.    u09AE.thisChar = "\u09AE";
525.
526.    //-- 09AF --------------------------------//
527.    u09AF.cha.r.text = "\u09AF";
528.    u09AF.thisChar = "\u09AF";
529.
530.    //-- 09B0 --------------------------------//
531.    u09B0.cha.r.text = "\u09B0";
532.    u09B0.thisChar = "\u09B0";
533.
534.    //-- 09B1 --------------------------------//
535.    u09B1.cha.r.text = "\u09B1";
536.    u09B1.thisChar = "\u09B1";
537.
538.    //-- 09B2 --------------------------------//
539.    u09B2.cha.r.text = "\u09B2";
540.    u09B2.thisChar = "\u09B2";
541.
542.    //-- 09B3 --------------------------------//
543.    u09B3.cha.r.text = "\u09B3";
544.    u09B3.thisChar = "\u09B3";
545.
546.    //-- 09B4 --------------------------------//
547.    u09B4.cha.r.text = "\u09B4";
548.    u09B4.thisChar = "\u09B4";
549.
550.    //-- 09B5 --------------------------------//
551.    u09B5.cha.r.text = "\u09B5";
552.    u09B5.thisChar = "\u09B5";
553.
554.    //-- 09B6 --------------------------------//
555.    u09B6.cha.r.text = "\u09B6";
556.    u09B6.thisChar = "\u09B6";
557.
558.    //-- 09B7 --------------------------------//
559.    u09B7.cha.r.text = "\u09B7";
560.    u09B7.thisChar = "\u09B7";
561.
562.    //-- 09B8 --------------------------------//
563.    u09B8.cha.r.text = "\u09B8";
564.    u09B8.thisChar = "\u09B8";
565.
566.    //-- 09B9 --------------------------------//
567.    u09B9.cha.r.text = "\u09B9";
568.    u09B9.thisChar = "\u09B9";
```

```
569.
570.    //-- 09BA --------------------------------//
571.    u09BA.cha.r.text = "\u09BA";
572.    u09BA.thisChar = "\u09BA";
573.
574.    //-- 09BB --------------------------------//
575.    u09BB.cha.r.text = "\u09BB";
576.    u09BB.thisChar = "\u09BB";
577.
578.    //-- 09BC --------------------------------//
579.    u09BC.cha.r.text = "\u09BC";
580.    u09BC.thisChar = "\u09BC";
581.
582.    //-- 09BD --------------------------------//
583.    u09BD.cha.r.text = "\u09BD";
584.    u09BD.thisChar = "\u09BD";
585.
586.    //-- 09BE --------------------------------//
587.    u09BE.cha.r.text = "\u09BE";
588.    u09BE.thisChar = "\u09BE";
589.
590.    //-- 09BF --------------------------------//
591.    u09BF.cha.r.text = "\u09BF";
592.    u09BF.thisChar = "\u09BF";
593.
594.    //-- 09C0 --------------------------------//
595.    u09C0.cha.r.text = "\u09C0";
596.    u09C0.thisChar = "\u09C0";
597.
598.    //-- 09C1 --------------------------------//
599.    u09C1.cha.r.text = "\u09C1";
600.    u09C1.thisChar = "\u09C1";
601.
602.    //-- 09C2 --------------------------------//
603.    u09C2.cha.r.text = "\u09C2";
604.    u09C2.thisChar = "\u09C2";
605.
606.    //-- 09C3 --------------------------------//
607.    u09C3.cha.r.text = "\u09C3";
608.    u09C3.thisChar = "\u09C3";
609.
610.    //-- 09C4 --------------------------------//
611.    u09C4.cha.r.text = "\u09C4";
612.    u09C4.thisChar = "\u09C4";
613.
614.    //-- 09C5 --------------------------------//
615.    u09C5.cha.r.text = "\u09C5";
616.    u09C5.thisChar = "\u09C5";
617.
618.    //-- 09C6 --------------------------------//
619.    u09C6.cha.r.text = "\u09C6";
620.    u09C6.thisChar = "\u09C6";
621.
622.    //-- 09C7 --------------------------------//
623.    u09C7.cha.r.text = "\u09C7";
624.    u09C7.thisChar = "\u09C7";
625.
626.    //-- 09C8 --------------------------------//
627.    u09C8.cha.r.text = "\u09C8";
628.    u09C8.thisChar = "\u09C8";
629.
630.    //-- 09C9 --------------------------------//
631.    u09C9.cha.r.text = "\u09C9";
632.    u09C9.thisChar = "\u09C9";
633.
634.    //-- 09CA --------------------------------//
635.    u09CA.cha.r.text = "\u09CA";
636.    u09CA.thisChar = "\u09CA";
637.
638.    //-- 09CB --------------------------------//
639.    u09CB.cha.r.text = "\u09CB";
640.    u09CB.thisChar = "\u09CB";
641.
642.    //-- 09CC --------------------------------//
643.    u09CC.cha.r.text = "\u09CC";
```

```
644.    u09CC.thisChar = "\u09CC";
645.
646.    //-- 09CD --------------------------------//
647.    u09CD.cha.r.text = "\u09CD";
648.    u09CD.thisChar = "\u09CD";
649.
650.    //-- 09CE --------------------------------//
651.    u09CE.cha.r.text = "\u09CE";
652.    u09CE.thisChar = "\u09CE";
653.
654.    //-- 09CF --------------------------------//
655.    u09CF.cha.r.text = "\u09CF";
656.    u09CF.thisChar = "\u09CF";
657.
658.    //-- 09D0 --------------------------------//
659.    u09D0.cha.r.text = "\u09D0";
660.    u09D0.thisChar = "\u09D0";
661.
662.    //-- 09D1 --------------------------------//
663.    u09D1.cha.r.text = "\u09D1";
664.    u09D1.thisChar = "\u09D1";
665.
666.    //-- 09D2 --------------------------------//
667.    u09D2.cha.r.text = "\u09D2";
668.    u09D2.thisChar = "\u09D2";
669.
670.    //-- 09D3 --------------------------------//
671.    u09D3.cha.r.text = "\u09D3";
672.    u09D3.thisChar = "\u09D3";
673.
674.    //-- 09D4 --------------------------------//
675.    u09D4.cha.r.text = "\u09D4";
676.    u09D4.thisChar = "\u09D4";
677.
678.    //-- 09D5 --------------------------------//
679.    u09D5.cha.r.text = "\u09D5";
680.    u09D5.thisChar = "\u09D5";
681.
682.    //-- 09D6 --------------------------------//
683.    u09D6.cha.r.text = "\u09D6";
684.    u09D6.thisChar = "\u09D6";
685.
686.    //-- 09D7 --------------------------------//
687.    u09D7.cha.r.text = "\u09D7";
688.    u09D7.thisChar = "\u09D7";
689.
690.    //-- 09D8 --------------------------------//
691.    u09D8.cha.r.text = "\u09D8";
692.    u09D8.thisChar = "\u09D8";
693.
694.    //-- 09D9 --------------------------------//
695.    u09D9.cha.r.text = "\u09D9";
696.    u09D9.thisChar = "\u09D9";
697.
698.    //-- 09DA --------------------------------//
699.    u09DA.cha.r.text = "\u09DA";
700.    u09DA.thisChar = "\u09DA";
701.
702.    //-- 09DB --------------------------------//
703.    u09DB.cha.r.text = "\u09DB";
704.    u09DB.thisChar = "\u09DB";
705.
706.    //-- 09DC --------------------------------//
707.    u09DC.cha.r.text = "\u09DC";
708.    u09DC.thisChar = "\u09DC";
709.
710.    //-- 09DD --------------------------------//
711.    u09DD.cha.r.text = "\u09DD";
712.    u09DD.thisChar = "\u09DD";
713.
714.    //-- 09DE --------------------------------//
715.    u09DE.cha.r.text = "\u09DE";
716.    u09DE.thisChar = "\u09DE";
717.
718.    //-- 09DF --------------------------------//
```

```
719.    u09DF.cha.r.text = "\u09DF";
720.    u09DF.thisChar = "\u09DF";
721.
722.    //-- 09E0 ------------------------------//
723.    u09E0.cha.r.text = "\u09E0";
724.    u09E0.thisChar = "\u09E0";
725.
726.    //-- 09E1 ------------------------------//
727.    u09E1.cha.r.text = "\u09E1";
728.    u09E1.thisChar = "\u09E1";
729.
730.    //-- 09E2 ------------------------------//
731.    u09E2.cha.r.text = "\u09E2";
732.    u09E2.thisChar = "\u09E2";
733.
734.    //-- 09E3 ------------------------------//
735.    u09E3.cha.r.text = "\u09E3";
736.    u09E3.thisChar = "\u09E3";
737.
738.    //-- 09E4 ------------------------------//
739.    u09E4.cha.r.text = "\u09E4";
740.    u09E4.thisChar = "\u09E4";
741.
742.    //-- 09E5 ------------------------------//
743.    u09E5.cha.r.text = "\u09E5";
744.    u09E5.thisChar = "\u09E5";
745.
746.    //-- 09E6 ------------------------------//
747.    u09E6.cha.r.text = "\u09E6";
748.    u09E6.thisChar = "\u09E6";
749.
750.    //-- 09E7 ------------------------------//
751.    u09E7.cha.r.text = "\u09E7";
752.    u09E7.thisChar = "\u09E7";
753.
754.    //-- 09E8 ------------------------------//
755.    u09E8.cha.r.text = "\u09E8";
756.    u09E8.thisChar = "\u09E8";
757.
758.    //-- 09E9 ------------------------------//
759.    u09E9.cha.r.text = "\u09E9";
760.    u09E9.thisChar = "\u09E9";
761.
762.    //-- 09EA ------------------------------//
763.    u09EA.cha.r.text = "\u09EA";
764.    u09EA.thisChar = "\u09EA";
765.
766.    //-- 09EB ------------------------------//
767.    u09EB.cha.r.text = "\u09EB";
768.    u09EB.thisChar = "\u09EB";
769.
770.    //-- 09EC ------------------------------//
771.    u09EC.cha.r.text = "\u09EC";
772.    u09EC.thisChar = "\u09EC";
773.
774.    //-- 09ED ------------------------------//
775.    u09ED.cha.r.text = "\u09ED";
776.    u09ED.thisChar = "\u09ED";
777.
778.    //-- 09EE ------------------------------//
779.    u09EE.cha.r.text = "\u09EE";
780.    u09EE.thisChar = "\u09EE";
781.
782.    //-- 09EF ------------------------------//
783.    u09EF.cha.r.text = "\u09EF";
784.    u09EF.thisChar = "\u09EF";
785.
786.    //-- 09F0 ------------------------------//
787.    u09F0.cha.r.text = "\u09F0";
788.    u09F0.thisChar = "\u09F0";
789.
790.    //-- 09F1 ------------------------------//
791.    u09F1.cha.r.text = "\u09F1";
792.    u09F1.thisChar = "\u09F1";
793.
```

```
794.    //-- 09F2 --------------------------------//
795.    u09F2.cha.r.text = "\u09F2";
796.    u09F2.thisChar = "\u09F2";
797.
798.    //-- 09F3 --------------------------------//
799.    u09F3.cha.r.text = "\u09F3";
800.    u09F3.thisChar = "\u09F3";
801.
802.    //-- 09F4 --------------------------------//
803.    u09F4.cha.r.text = "\u09F4";
804.    u09F4.thisChar = "\u09F4";
805.
806.    //-- 09F5 --------------------------------//
807.    u09F5.cha.r.text = "\u09F5";
808.    u09F5.thisChar = "\u09F5";
809.
810.    //-- 09F6 --------------------------------//
811.    u09F6.cha.r.text = "\u09F6";
812.    u09F6.thisChar = "\u09F6";
813.
814.    //-- 09F7 --------------------------------//
815.    u09F7.cha.r.text = "\u09F7";
816.    u09F7.thisChar = "\u09F7";
817.
818.    //-- 09F8 --------------------------------//
819.    u09F8.cha.r.text = "\u09F8";
820.    u09F8.thisChar = "\u09F8";
821.
822.    //-- 09F9 --------------------------------//
823.    u09F9.cha.r.text = "\u09F9";
824.    u09F9.thisChar = "\u09F9";
825.
826.    //-- 09FA --------------------------------//
827.    u09FA.cha.r.text = "\u09FA";
828.    u09FA.thisChar = "\u09FA";
829.
830.    //----------------------------------------//
831.
832.    //mainText.text == "আমি বাংলায় কথা বলতে পারি";
833.
834.    on (press, keyPress "l") {
835.            tellTarget("listenButton") {
836.                    gotoAndStop("down");
837.            }
838.            if (mainText.text == "আমি বাংলায় কথা বলতে পারি") {
839.                    myMusic = new Sound(myMusicMc);
840.                    myMusic.loadSound("bn.mp3", true);
841.            }
842.            else {
843.                    audioLoadText.text = "Sending text to the Festival engine...";
844.                    varsHolderIn = new LoadVars();
845.                    varsHolderOut = new LoadVars();
846.                    varsHolderIn.onLoad = function(success) {
847.                            audioLoadText.text = "Copying audio file to LABS.com.bd
    from:\n" + varsHolderIn.remoteFileLocation;
848.                            varsHolderIn1 = new LoadVars();
849.                            varsHolderOut1 = new LoadVars();
850.                            varsHolderIn1.onLoad = function(success) {
851.                                    audioLoadText.text = "Sound Loading
    From:\nlabs.com.bd/voice/files/" + varsHolderIn1.playAudioFile;
852.
            getURL("loadRemoteAudio.php?file_base="+varsHolderIn1.playAudioFile,
    "loadSoundPage");
853.                                    //myMusic = new Sound(myMusicMc);
854.
            //myMusic.loadSound("files/"+varsHolderIn1.playAudioFile);
855.                            }
856.                            varsHolderOut1.audio_copy_source =
    varsHolderIn.remoteFileLocation;
857.                            varsHolderOut1.sendAndLoad("getRemoteAudio.php",
    varsHolderIn1, "POST");
858.                    }
859.                    varsHolderOut.callFromRemoteLocationForEnglishLanguage = "";
860.                    varsHolderOut.speech = mainText.text;
```

```
861.                    varsHolderOut.volume_scale = "3";
862.                    varsHolderOut.callFromRemoteLocation = "true";
863.                    varsHolderOut.make_audio = " Listen ";
864.                    varsHolderOut.sendAndLoad("getRemoteContent.php", varsHolderIn,
      "POST");
865.            }
866.    }
867.    on (rollOver, dragOver) {
868.            tellTarget("listenButton") {
869.                    gotoAndStop("up");
870.            }
871.    }
872.    on (release, releaseOutside, rollOut, dragOut) {
873.            tellTarget("listenButton") {
874.                    gotoAndStop("normal");
875.            }
876.    }
877.
878.    on (press, keyPress "l") {
879.            tellTarget("listenButton") {
880.                    gotoAndStop("down");
881.            }
882.            if (mainText.text == "The quick brown fox jumped over the lazy dog!") {
883.                    myMusic = new Sound(myMusicMc);
884.                    myMusic.loadSound("en.mp3", true);
885.            }
886.            else {
887.                    audioLoadText.text = "Sending text to the Festival engine...";
888.                    varsHolderIn = new LoadVars();
889.                    varsHolderOut = new LoadVars();
890.                    varsHolderIn.onLoad = function(success) {
891.                            audioLoadText.text = "Copying audio file to LABS.com.bd
      from:\n" + varsHolderIn.remoteFileLocation;
892.                            varsHolderIn1 = new LoadVars();
893.                            varsHolderOut1 = new LoadVars();
894.                            varsHolderIn1.onLoad = function(success) {
895.                                    audioLoadText.text = "Sound Loading
      From:\nlabs.com.bd/voice/files/" + varsHolderIn1.playAudioFile;
896.
            getURL("loadRemoteAudio.php?file_base="+varsHolderIn1.playAudioFile,
      "loadSoundPage");
897.                                    //myMusic = new Sound(myMusicMc);
898.
            //myMusic.loadSound("files/"+varsHolderIn1.playAudioFile);
899.                            }
900.                            varsHolderOut1.audio_copy_source =
      varsHolderIn.remoteFileLocation;
901.                            varsHolderOut1.sendAndLoad("getRemoteAudio.php",
      varsHolderIn1, "POST");
902.                    }
903.                    varsHolderOut.callFromRemoteLocationForEnglishLanguage = "EN";
904.                    varsHolderOut.speech = mainText.text;
905.                    varsHolderOut.volume_scale = "3";
906.                    varsHolderOut.callFromRemoteLocation = "true";
907.                    varsHolderOut.make_audio = " Listen ";
908.                    varsHolderOut.sendAndLoad("getRemoteContent.php", varsHolderIn,
      "POST");
909.            }
910.    }
911.    on (rollOver, dragOver) {
912.            tellTarget("listenButton") {
913.                    gotoAndStop("up");
914.            }
915.    }
916.    on (release, releaseOutside, rollOut, dragOut) {
917.            tellTarget("listenButton") {
918.                    gotoAndStop("normal");
919.            }
920.    }
921.
922.    stop();
923.    var currentImageURL:String = "";
924.    var currentImageFileName:String = "";
925.
926.    //Allow this domain
```

```
927.    System.security.allowDomain("http://localhost/");
928.    import flash.net.FileReference;
929.    // The listener object listens for FileReference events.
930.    var listener:Object = new Object();
931.
932.    // When the user selects a file, the onSelect() method is called, and
933.    // passed a reference to the FileReference object.
934.    listener.onSelect = function(selectedFile:FileReference):Void {
935.      //clean statusArea and details area
936.      statusArea.text = details.text = ""
937.      // Flash is attempting to upload the image.
938.      statusArea.text += "Attempting to upload " + selectedFile.name + "\n";
939.      // Upload the file to the PHP script on the server.
940.      selectedFile.upload("uploadImg.php");
941.    };
942.
943.    // the file is starting to upload.
944.    listener.onOpen = function(selectedFile:FileReference):Void {
945.      statusArea.text += "Uploading " + selectedFile.name + "\n";
946.    };
947.    //Possible file upload errors
948.    listener.onHTTPError = function(file:FileReference, httpError:Number):Void {
949.            imagePane.contentPath = "error";
950.            imagePane.content.errorMSG.text = "HTTPError number: "+httpError +"\nFile:
        "+ file.name;
951.    }
952.
953.    listener.onIOError = function(file:FileReference):Void {
954.            imagePane.contentPath = "error";
955.            imagePane.content.errorMSG.text = "IOError: "+ file.name;
956.    }
957.
958.    listener.onSecurityError = function(file:FileReference, errorString:String):Void {
959.            imagePane.contentPath = "error";
960.            imagePane.content.errorMSG.text = "SecurityError: "+SecurityError+"\nFile:
        "+ file.name;
961.    }
962.
963.    // the file has uploaded
964.    listener.onComplete = function(selectedFile:FileReference):Void {
965.      // Notify the user that Flash is starting to download the image.
966.      statusArea.text += "Upload finished.\nNow downloading " + selectedFile.name + "
        to player\n";
967.      //Show file details
968.      details.text = ""
969.      for(i in selectedFile) details.text +="<b>"+i+":</b> "+selectedFile[i]+"\n"
970.      // Call the custom downloadImage() function.
971.      downloadImage(selectedFile.name);
972.    };
973.
974.    var imageFile:FileReference = new FileReference();
975.    imageFile.addListener(listener);
976.
977.    uploadBtn.onPress = uploadImage;
978.    imagePane.addEventListener("complete", imageDownloaded);
979.
980.    // Call the uploadImage() function, opens a file browser dialog.
981.    function uploadImage(event:Object):Void {
982.      imageFile.browse([{description: "PNG Images (.png)", extension: "*.png"}]);
983.    }
984.
985.    // If the image does not download, the event object's total property
986.    // will equal -1. In that case, display am error message
987.    function imageDownloaded(event:Object):Void {
988.      if(event.total == -1) {
989.        imagePane.contentPath = "error";
990.      }
991.    }
992.
993.    // show uploaded image in scrollPane
994.    function downloadImage(file:Object):Void {
995.      imagePane.contentPath =  "files/" + file;
996.      currentImageURL = 'http://labs.com.bd/voice/files/' + file;
997.      currentImageFileName = file;
998.    }
```

```
999.
1000.   stop();
1001.
1002.   on (press, keyPress "l") {
1003.           tellTarget("listenButton") {
1004.                   gotoAndStop("down");
1005.           }
1006.           if (mainText.text == 'ছবি থেকে স্ক্যান করা লেখা এখানে দেখানো হবে') {
1007.                   myMusic = new Sound(myMusicMc);
1008.                   myMusic.loadSound("bn_scan.mp3", true);
1009.           }
1010.           else {
1011.                   audioLoadText.text = "Sending text to the Festival engine...";
1012.                   varsHolderIn = new LoadVars();
1013.                   varsHolderOut = new LoadVars();
1014.                   varsHolderIn.onLoad = function(success) {
1015.                           audioLoadText.text = "Copying audio file to LABS.com.bd
        from:\n" + varsHolderIn.remoteFileLocation;
1016.                           varsHolderIn1 = new LoadVars();
1017.                           varsHolderOut1 = new LoadVars();
1018.                           varsHolderIn1.onLoad = function(success) {
1019.                                   audioLoadText.text = "Sound Loading
        From:\nlabs.com.bd/voice/files/" + varsHolderIn1.playAudioFile;
1020.
                   getURL("loadRemoteAudio.php?file_base="+varsHolderIn1.playAudioFile,
        "loadSoundPage");
1021.                                   //myMusic = new Sound(myMusicMc);
1022.
                   //myMusic.loadSound("files/"+varsHolderIn1.playAudioFile);
1023.                           }
1024.                           varsHolderOut1.audio_copy_source =
        varsHolderIn.remoteFileLocation;
1025.                           varsHolderOut1.sendAndLoad("getRemoteAudio.php",
        varsHolderIn1, "POST");
1026.                   }
1027.                   varsHolderOut.callFromRemoteLocationForEnglishLanguage = "";
1028.                   varsHolderOut.speech = mainText.text;
1029.                   varsHolderOut.volume_scale = "3";
1030.                   varsHolderOut.callFromRemoteLocation = "true";
1031.                   varsHolderOut.make_audio = " Listen ";
1032.                   varsHolderOut.sendAndLoad("getRemoteContent.php", varsHolderIn,
        "POST");
1033.           }
1034.   }
1035.   on (rollOver, dragOver) {
1036.           tellTarget("listenButton") {
1037.                   gotoAndStop("up");
1038.           }
1039.   }
1040.   on (release, releaseOutside, rollOut, dragOut) {
1041.           tellTarget("listenButton") {
1042.                   gotoAndStop("normal");
1043.           }
1044.   }
1045.
1046.   on (press, keyPress "t") {
1047.           varsHolderIn = new LoadVars();
1048.           varsHolderOut = new LoadVars();
1049.           varsHolderIn.onLoad = function(success) {
1050.                   mainText.text += varsHolderIn.convertedText;
1051.           }
1052.           varsHolderOut.remoteFileName = currentImageURL;
1053.           varsHolderOut.remoteFileURL = currentImageFileName;
1054.           varsHolderOut.remoteLanguage = "BN";
1055.           varsHolderOut.sendAndLoad("getRemoteText.php", varsHolderIn, "POST");
1056.   }
1057.   on (rollOver, dragOver) {
1058.   }
1059.   on (release, releaseOutside, rollOut, dragOut) {
1060.   }
1061.
1062.   on (press, keyPress "s") {
1063.           imagePane.contentPath =  loadURL.text;
1064.   }
```

```
1065.
1066.   on (press, keyPress "l") {
1067.           tellTarget("listenButton") {
1068.                   gotoAndStop("down");
1069.           }
1070.           if (mainText.text == 'Scanned text will appear here!') {
1071.                   myMusic = new Sound(myMusicMc);
1072.                   myMusic.loadSound("en_scan.mp3", true);
1073.           }
1074.           else {
1075.                   audioLoadText.text = "Sending text to the Festival engine...";
1076.                   varsHolderIn = new LoadVars();
1077.                   varsHolderOut = new LoadVars();
1078.                   varsHolderIn.onLoad = function(success) {
1079.                           audioLoadText.text = "Copying audio file to LABS.com.bd
        from:\n" + varsHolderIn.remoteFileLocation;
1080.                           varsHolderIn1 = new LoadVars();
1081.                           varsHolderOut1 = new LoadVars();
1082.                           varsHolderIn1.onLoad = function(success) {
1083.                                   audioLoadText.text = "Sound Loading
        From:\nlabs.com.bd/voice/files/" + varsHolderIn1.playAudioFile;
1084.
                getURL("loadRemoteAudio.php?file_base="+varsHolderIn1.playAudioFile,
        "loadSoundPage");
1085.                                   //myMusic = new Sound(myMusicMc);
1086.
                //myMusic.loadSound("files/"+varsHolderIn1.playAudioFile);
1087.                           }
1088.                           varsHolderOut1.audio_copy_source =
        varsHolderIn.remoteFileLocation;
1089.                           varsHolderOut1.sendAndLoad("getRemoteAudio.php",
        varsHolderIn1, "POST");
1090.                   }
1091.                   varsHolderOut.callFromRemoteLocationForEnglishLanguage = "EN";
1092.                   varsHolderOut.speech = mainText.text;
1093.                   varsHolderOut.volume_scale = "3";
1094.                   varsHolderOut.callFromRemoteLocation = "true";
1095.                   varsHolderOut.make_audio = " Listen ";
1096.                   varsHolderOut.sendAndLoad("getRemoteContent.php", varsHolderIn,
        "POST");
1097.           }
1098.   }
1099.   on (rollOver, dragOver) {
1100.           tellTarget("listenButton") {
1101.                   gotoAndStop("up");
1102.           }
1103.   }
1104.   on (release, releaseOutside, rollOut, dragOut) {
1105.           tellTarget("listenButton") {
1106.                   gotoAndStop("normal");
1107.           }
1108.   }
1109.
1110.   on (press, keyPress "t") {
1111.           varsHolderIn = new LoadVars();
1112.           varsHolderOut = new LoadVars();
1113.           varsHolderIn.onLoad = function(success) {
1114.                   mainText.text += varsHolderIn.convertedText;
1115.           }
1116.           varsHolderOut.remoteFileName = currentImageURL;
1117.           varsHolderOut.remoteFileURL = currentImageFileName;
1118.           varsHolderOut.remoteLanguage = "EN";
1119.           varsHolderOut.sendAndLoad("getRemoteText.php", varsHolderIn, "POST");
1120.   }
1121.   on (rollOver, dragOver) {
1122.   }
1123.   on (release, releaseOutside, rollOut, dragOut) {
1124.   }
1125.
1126.   on (press, keyPress "s") {
1127.           imagePane.contentPath =  loadURL.text;
1128.   }
1129.
1130.   this.createEmptyMovieClip("sound_mc",this.getNextHighestDepth());
1131.   var active_mic:Microphone = Microphone.get();
```

```
1132.   sound_mc.attachAudio(active_mic);
1133.
1134.   limite = 40;
1135.
1136.   this.onEnterFrame = function() {
1137.           //vaias
1138.           if (active_mic.activityLevel>40) {
1139.                   trace("foi");
1140.                   mcAnima.play();
1141.           }
1142.           masc._width=active_mic.activityLevel;
1143.   };
1144.
1145.   soundEnd = true;
1146.
1147.   function callback1() {
1148.           som.start();
1149.           soundEnd = true;
1150.   }
1151.
1152.   var som:Sound = new Sound();
1153.
1154.   som.start();
1155.
1156.   //System.showSettings(2);
1157.   btAudio.onPress = function() {
1158.           System.showSettings(2);
1159.   };
1160.
1161.   //sound.loadSound("vaias.mp3", true);
1162.   stop();
1163.
1164.   on (press, keyPress "l") {
1165.           tellTarget("listenButton") {
1166.                   gotoAndStop("down");
1167.           }
1168.           if (mainText.text == 'Scanned text will appear here!') {
1169.                   myMusic = new Sound(myMusicMc);
1170.                   myMusic.loadSound("en_scan.mp3", true);
1171.           }
1172.           else {
1173.                   audioLoadText.text = "Sending text to the Festival engine...";
1174.                   varsHolderIn = new LoadVars();
1175.                   varsHolderOut = new LoadVars();
1176.                   varsHolderIn.onLoad = function(success) {
1177.                           audioLoadText.text = "Copying audio file to LABS.com.bd
      from:\n" + varsHolderIn.remoteFileLocation;
1178.                           varsHolderIn1 = new LoadVars();
1179.                           varsHolderOut1 = new LoadVars();
1180.                           varsHolderIn1.onLoad = function(success) {
1181.                                   audioLoadText.text = "Sound Loading
      From:\nlabs.com.bd/voice/files/" + varsHolderIn1.playAudioFile;
1182.
           getURL("loadRemoteAudio.php?file_base="+varsHolderIn1.playAudioFile,
      "loadSoundPage");
1183.                                   //myMusic = new Sound(myMusicMc);
1184.
           //myMusic.loadSound("files/"+varsHolderIn1.playAudioFile);
1185.                           }
1186.                           varsHolderOut1.audio_copy_source =
      varsHolderIn.remoteFileLocation;
1187.                           varsHolderOut1.sendAndLoad("getRemoteAudio.php",
      varsHolderIn1, "POST");
1188.                   }
1189.                   varsHolderOut.callFromRemoteLocationForEnglishLanguage = "EN";
1190.                   varsHolderOut.speech = mainText.text;
1191.                   varsHolderOut.volume_scale = "3";
1192.                   varsHolderOut.callFromRemoteLocation = "true";
1193.                   varsHolderOut.make_audio = " Listen ";
1194.                   varsHolderOut.sendAndLoad("getRemoteContent.php", varsHolderIn,
      "POST");
1195.           }
1196.   }
1197.   on (rollOver, dragOver) {
1198.           tellTarget("listenButton") {
```

```
                gotoAndStop("up");
        }
}
on (release, releaseOutside, rollOut, dragOut) {
        tellTarget("listenButton") {
                gotoAndStop("normal");
        }
}

on (press, keyPress "s") {
        audioLoadText.text = "ডাটাবেস থেকে লোড হচ্ছে...";
        varsHolderIn = new LoadVars();
        varsHolderOut = new LoadVars();
        varsHolderIn.onLoad = function(success) {
                audioLoadText.text = "তথ্য লোড হয়েছে";
                if (varsHolderIn.outputWordMeaning == "0") {
                        outputText.text = "দুঃখিত, আমাদের কাছে শব্দটি সম্পর্কে কোনো তথ্য নেই...";
                }
                else {
                        outputText.htmlText = "<b><i> " + varsHolderOut.searchWord + "</i></b> সম্পর্কে নিম্নলিখিত ফলাফল পাওয়া গেছে:<br>(শব্দ, পদ,অর্থ, উদাহরণ, সম্পর্কিত শব্দ)<br>" + varsHolderIn.outputWordMeaning;
                }
        }
        varsHolderOut.searchWord = mainText.text;
        varsHolderOut.sendAndLoad("getWord.php", varsHolderIn, "POST");
}

on (press, keyPress "s") {
        audioLoadText.text = "Loading word meaning from database...";
        varsHolderIn = new LoadVars();
        varsHolderOut = new LoadVars();
        varsHolderIn.onLoad = function(success) {
                audioLoadText.text = "Output text loaded";
                if (varsHolderIn.outputWordMeaning == "0") {
                        outputText.text = "Sorry! The word is not available in our WordNet database yet...";
                }
                else {
                        outputText.htmlText = "We found the following result(s) with the search term <b><i> " + varsHolderOut.searchWord + "</i></b>:<br>Format: (Word, Parts of Speech, Meaning, Example, Related Words)<br>" + varsHolderIn.outputWordMeaning;
                }
        }
        varsHolderOut.searchWord = mainText.text;
        varsHolderOut.sendAndLoad("getWord.php", varsHolderIn, "POST");
}

on(press){
  //_root.doFullScreen();
  _root.lan = "EN";
  _root.play();
  _root.clicked = true;
}
on (rollOver, dragOver) {
        if (!_root.clicked) {
                gotoAndPlay("over");
        }
}
on (rollOut, dragOut) {
        if (!_root.clicked) {
                gotoAndPlay("out");
        }
}

var i = 1;
while (i < 80)
{
    duplicateMovieClip("circle", "circle" + i, i);
    setProperty("circle" + i, _rotation, i * 4.500000E+000);
    setProperty("circle" + i, _alpha, 80 - i);
```

```
1265.       ++i;
1266.   } // end while
1267.   this.onEnterFrame = function ()
1268.   {
1269.       this._rotation = this._rotation - 17;
1270.   };
1271.
1272.   stop();
1273.   var thisChar:String;
1274.
1275.   on (press) {
1276.           gotoAndStop("down");
1277.           _parent.mainText.text = _parent.mainText.text + "\n";
1278.   }
1279.   on (rollOver, dragOver) {
1280.           gotoAndStop("up");
1281.   }
1282.   on (release, releaseOutside, rollOut, dragOut) {
1283.           gotoAndStop("normal");
1284.   }
1285.
1286.   on (press) {
1287.           gotoAndStop("down");
1288.           _parent.mainText.text =
       substring(_parent.mainText.text,0,_parent.mainText.text.length-1);
1289.   }
1290.   on (rollOver, dragOver) {
1291.           gotoAndStop("up");
1292.   }
1293.   on (release, releaseOutside, rollOut, dragOut) {
1294.           gotoAndStop("normal");
1295.   }
1296.
1297.   on (press) {
1298.           gotoAndStop("down");
1299.           _parent.mainText.text = _parent.mainText.text + "\u09CD";
1300.   }
1301.   on (rollOver, dragOver) {
1302.           gotoAndStop("up");
1303.   }
1304.   on (release, releaseOutside, rollOut, dragOut) {
1305.           gotoAndStop("normal");
1306.   }
1307.
1308.   function switchcaps()
1309.   {
1310.       if (!_root.currentPage.mainContent.keyboard_mc.caps)
1311.       {
1312.           for (c = 65; c <= 90; c++)
1313.           {
1314.               _root.currentPage.mainContent.keyboard_mc["k_" + String.fromCharCode(c
       + 32)].label = String.fromCharCode(c);
1315.           } // end of for
1316.           _root.currentPage.mainContent.keyboard_mc.caps = true;
1317.       }
1318.       else
1319.       {
1320.           for (c = 65; c <= 90; c++)
1321.           {
1322.               _root.currentPage.mainContent.keyboard_mc["k_" + String.fromCharCode(c
       + 32)].label = String.fromCharCode(c + 32);
1323.           } // end of for
1324.           _root.currentPage.mainContent.keyboard_mc.caps = false;
1325.           _root.currentPage.mainContent.keyboard_mc.shift = false;
1326.       } // end else if
1327.   } // End of the function
1328.   function switchshift()
1329.   {
1330.       if (!_root.currentPage.mainContent.keyboard_mc.shift)
1331.       {
1332.           for (c = 65; c <= 90; c++)
1333.           {
1334.               _root.currentPage.mainContent.keyboard_mc["k_" + String.fromCharCode(c
       + 32)].label = String.fromCharCode(c + 32);
1335.           } // end of for
```

```
1336.           _root.currentPage.mainContent.keyboard_mc.k_semi.label = ":";
1337.           _root.currentPage.mainContent.keyboard_mc.k_apost.label = "\"";
1338.           _root.currentPage.mainContent.keyboard_mc.k_comma.label = "<";
1339.           _root.currentPage.mainContent.keyboard_mc.k_period.label = ">";
1340.           _root.currentPage.mainContent.keyboard_mc.k_slash.label = "?";
1341.           _root.currentPage.mainContent.keyboard_mc.k_dash.label = "_";
1342.           _root.currentPage.mainContent.keyboard_mc.k_equal.label = "+";
1343.           _root.currentPage.mainContent.keyboard_mc.k_L_brack.label = "{";
1344.           _root.currentPage.mainContent.keyboard_mc.k_R_brack.label = "}";
1345.           _root.currentPage.mainContent.keyboard_mc.k_backslash.label = "|";
1346.           _root.currentPage.mainContent.keyboard_mc.k_tilde.label = "~";
1347.           _root.currentPage.mainContent.keyboard_mc.k_1.label = "!";
1348.           _root.currentPage.mainContent.keyboard_mc.k_2.label = "@";
1349.           _root.currentPage.mainContent.keyboard_mc.k_3.label = "#";
1350.           _root.currentPage.mainContent.keyboard_mc.k_4.label = "$";
1351.           _root.currentPage.mainContent.keyboard_mc.k_5.label = "%";
1352.           _root.currentPage.mainContent.keyboard_mc.k_6.label = "^";
1353.           _root.currentPage.mainContent.keyboard_mc.k_7.label = "&";
1354.           _root.currentPage.mainContent.keyboard_mc.k_8.label = "*";
1355.           _root.currentPage.mainContent.keyboard_mc.k_9.label = "(";
1356.           _root.currentPage.mainContent.keyboard_mc.k_0.label = ")";
1357.           _root.currentPage.mainContent.keyboard_mc.shift = true;
1358.       }
1359.       else
1360.       {
1361.           for (c = 65; c <= 90; c++)
1362.           {
1363.               _root.currentPage.mainContent.keyboard_mc["k_" + String.fromCharCode(c
      + 32)].label = String.fromCharCode(c);
1364.           } // end of for
1365.           _root.currentPage.mainContent.keyboard_mc.k_semi.label = ";";
1366.           _root.currentPage.mainContent.keyboard_mc.k_apost.label = "\'";
1367.           _root.currentPage.mainContent.keyboard_mc.k_comma.label = ",";
1368.           _root.currentPage.mainContent.keyboard_mc.k_period.label = ".";
1369.           _root.currentPage.mainContent.keyboard_mc.k_slash.label = "/";
1370.           _root.currentPage.mainContent.keyboard_mc.k_dash.label = "-";
1371.           _root.currentPage.mainContent.keyboard_mc.k_equal.label = "=";
1372.           _root.currentPage.mainContent.keyboard_mc.k_L_brack.label = "[";
1373.           _root.currentPage.mainContent.keyboard_mc.k_R_brack.label = "]";
1374.           _root.currentPage.mainContent.keyboard_mc.k_backslash.label = "\\";
1375.           _root.currentPage.mainContent.keyboard_mc.k_tilde.label = "`";
1376.           _root.currentPage.mainContent.keyboard_mc.k_1.label = "1";
1377.           _root.currentPage.mainContent.keyboard_mc.k_2.label = "2";
1378.           _root.currentPage.mainContent.keyboard_mc.k_3.label = "3";
1379.           _root.currentPage.mainContent.keyboard_mc.k_4.label = "4";
1380.           _root.currentPage.mainContent.keyboard_mc.k_5.label = "5";
1381.           _root.currentPage.mainContent.keyboard_mc.k_6.label = "6";
1382.           _root.currentPage.mainContent.keyboard_mc.k_7.label = "7";
1383.           _root.currentPage.mainContent.keyboard_mc.k_8.label = "8";
1384.           _root.currentPage.mainContent.keyboard_mc.k_9.label = "9";
1385.           _root.currentPage.mainContent.keyboard_mc.k_0.label = "0";
1386.           _root.currentPage.mainContent.keyboard_mc.shift = false;
1387.           _root.currentPage.mainContent.keyboard_mc.caps = false;
1388.       } // end else if
1389.   } // End of the function
1390.   _global.style.setStyle("fontSize", 16);
1391.   _global.style.setStyle("fontWeight", "bold");
1392.   _root.currentPage.mainContent.keyboard_mc.shift = false;
1393.   _root.currentPage.mainContent.keyboard_mc.caps = false;
1394.   //_root.currentPage.mainContent.mainText.text = "";
1395.   form = new Object();
1396.   form.click = function (eventObj)
1397.   {
1398.       switch (eventObj.target._name)
1399.       {
1400.           case "k_L_shift":
1401.           {
1402.               _root.currentPage.mainContent.keyboard_mc.switchshift();
1403.               break;
1404.           }
1405.           case "k_R_shift":
1406.           {
1407.               _root.currentPage.mainContent.keyboard_mc.switchshift();
1408.               break;
1409.           }
```

```
1410.           case "k_caps":
1411.           {
1412.               _root.currentPage.mainContent.keyboard_mc.switchcaps();
1413.               break;
1414.           }
1415.           case "k_space":
1416.           {
1417.               _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + " ";
1418.               break;
1419.           }
1420.           case "k_backspace":
1421.           {
1422.               _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text.substr(0, -1);
1423.               break;
1424.           }
1425.           case "k_tab":
1426.           {
1427.               _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "\t";
1428.               break;
1429.           }
1430.           case "k_tilde":
1431.           {
1432.               if (_root.currentPage.mainContent.keyboard_mc.shift)
1433.               {
1434.                   _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "~";
1435.               }
1436.               else
1437.               {
1438.                   _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "`";
1439.               } // end else if
1440.               break;
1441.           }
1442.           case "k_dash":
1443.           {
1444.               if (_root.currentPage.mainContent.keyboard_mc.shift)
1445.               {
1446.                   _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "_";
1447.               }
1448.               else
1449.               {
1450.                   _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "-";
1451.               } // end else if
1452.               break;
1453.           }
1454.           case "k_equal":
1455.           {
1456.               if (_root.currentPage.mainContent.keyboard_mc.shift)
1457.               {
1458.                   _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "+";
1459.               }
1460.               else
1461.               {
1462.                   _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "=";
1463.               } // end else if
1464.               break;
1465.           }
1466.           case "k_L_brack":
1467.           {
1468.               if (_root.currentPage.mainContent.keyboard_mc.shift)
1469.               {
1470.                   _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "{";
1471.               }
1472.               else
1473.               {
```

```
1474.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "[";
1475.             } // end else if
1476.             break;
1477.         }
1478.         case "k_R_brack":
1479.         {
1480.             if (_root.currentPage.mainContent.keyboard_mc.shift)
1481.             {
1482.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "}";
1483.             }
1484.             else
1485.             {
1486.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "]";
1487.             } // end else if
1488.             break;
1489.         }
1490.         case "k_backslash":
1491.         {
1492.             if (_root.currentPage.mainContent.keyboard_mc.shift)
1493.             {
1494.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "|";
1495.             }
1496.             else
1497.             {
1498.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "\"";
1499.             } // end else if
1500.             break;
1501.         }
1502.         case "k_semi":
1503.         {
1504.             if (_root.currentPage.mainContent.keyboard_mc.shift)
1505.             {
1506.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + ":";
1507.             }
1508.             else
1509.             {
1510.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + ";";
1511.             } // end else if
1512.             break;
1513.         }
1514.         case "k_apost":
1515.         {
1516.             if (_root.currentPage.mainContent.keyboard_mc.shift)
1517.             {
1518.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "\"";
1519.             }
1520.             else
1521.             {
1522.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "\'";
1523.             } // end else if
1524.             break;
1525.         }
1526.         case "k_comma":
1527.         {
1528.             if (_root.currentPage.mainContent.keyboard_mc.shift)
1529.             {
1530.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + "<";
1531.             }
1532.             else
1533.             {
1534.                 _root.currentPage.mainContent.mainText.text =
       _root.currentPage.mainContent.mainText.text + ",";
1535.             } // end else if
1536.             break;
1537.         }
```

```
1538.           case "k_period":
1539.           {
1540.               if (_root.currentPage.mainContent.keyboard_mc.shift)
1541.               {
1542.                   _root.currentPage.mainContent.mainText.text =
      _root.currentPage.mainContent.mainText.text + ">";
1543.               }
1544.               else
1545.               {
1546.                   _root.currentPage.mainContent.mainText.text =
      _root.currentPage.mainContent.mainText.text + ".";
1547.               } // end else if
1548.               break;
1549.           }
1550.           case "k_slash":
1551.           {
1552.               if (_root.currentPage.mainContent.keyboard_mc.shift)
1553.               {
1554.                   _root.currentPage.mainContent.mainText.text =
      _root.currentPage.mainContent.mainText.text + "?";
1555.               }
1556.               else
1557.               {
1558.                   _root.currentPage.mainContent.mainText.text =
      _root.currentPage.mainContent.mainText.text + "/";
1559.               } // end else if
1560.               break;
1561.           }
1562.           case "k_L_clear":
1563.           {
1564.               _root.currentPage.mainContent.mainText.text = "";
1565.               break;
1566.           }
1567.           case "k_R_clear":
1568.           {
1569.               _root.currentPage.mainContent.mainText.text = "";
1570.               break;
1571.           }
1572.           case "k_enter":
1573.           {
1574.               _root.currentPage.mainContent.mainText.text =
      _root.currentPage.mainContent.mainText.text + "\n";
1575.               break;
1576.           }
1577.           default:
1578.           {
1579.               _root.currentPage.mainContent.mainText.text =
      _root.currentPage.mainContent.mainText.text + eventObj.target.label;
1580.               break;
1581.           }
1582.       } // End of switch
1583.   };
1584.   myRoot = _root.currentPage.mainContent.keyboard_mc;
1585.   for (i in myRoot)
1586.   {
1587.       if (myRoot[i] instanceof MovieClip)
1588.       {
1589.           myRoot[i].addEventListener("click", form);
1590.       } // end if
1591.   } // end of for...in
1592.   stop ();
1593.
1594.   onClipEvent (construct)
1595.   {
1596.       icon = "";
1597.       label = "SPACE";
1598.       labelPlacement = "right";
1599.       selected = false;
1600.       toggle = false;
1601.       enabled = true;
1602.       visible = true;
1603.       minHeight = 0;
1604.       minWidth = 0;
1605.   }
1606.
```

```
1607.   onClipEvent (initialize)
1608.   {
1609.       selected = false;
1610.       toggle = false;
1611.       enabled = true;
1612.       visible = true;
1613.       minHeight = 0;
1614.       minWidth = 0;
1615.   }
1616.    #initclip 54
1617.   mx.skins.SkinElement.registerElement(mx.skins.RectBorder.symbolName,
        Object(mx.skins.RectBorder));
1618.   Object.registerClass("RectBorder", mx.skins.halo.RectBorder);
1619.   #endinitclip
1620.   stop ();
1621.
1622.   on (keyPress "<Tab>")
1623.   {
1624.       this.tabHandler();
1625.   }
1626.
1627.   mx.skins.ColoredSkinElement.setColorStyle(this, "buttonColor");
1628.
1629.   mx.skins.ColoredSkinElement.setColorStyle(this, "shadowColor");
1630.
1631.   mx.skins.ColoredSkinElement.setColorStyle(this, "borderColor");
1632.
1633.   #initclip 47
1634.   Object.registerClass("FocusManager", mx.managers.FocusManager);
1635.   if (_root.focusManager == undefined)
1636.   {
1637.       _root.createClassObject(mx.managers.FocusManager, "focusManager",
        mx.managers.DepthManager.highestDepth--);
1638.   } // end if
1639.   #endinitclip
```

A.2 Open-Source Documentations

Festival Speech Synthesis System

5 Overview

Festival is designed as a speech synthesis system for at least three levels of user. First, those who simply want high quality speech from arbitrary text with the minimum of effort. Second, those who are developing language systems and wish to include synthesis output. In this case, a certain amount of customization is desired, such as different voices, specific phrasing, dialog types etc. The third level is in developing and testing new synthesis methods.

This manual is not designed as a tutorial on converting text to speech but for documenting the processes and use of our system. We do not discuss the detailed algorithms involved in converting text to speech or the relative merits of multiple methods, though we will often give references to relevant papers when describing the use of each module.

For more general information about text to speech we recommend Dutoit's `An introduction to Text-to-Speech Synthesis' dutoit97. For more detailed research issues in TTS see sproat98 or vansanten96.

5.1 Philosophy

One of the biggest problems in the development of speech synthesis, and other areas of speech and language processing systems, is that there are a lot of simple well-known techniques lying around which can help you realise your goal. But in order to improve some part of the whole system it is necessary to have a whole system in which you can test and improve your part. Festival is intended as that whole system in which you may simply work on your small part to improve the whole. Without a system like Festival, before you could even start to test your

new module you would need to spend significant effort to build a whole system, or adapt an existing one before you could start working on your improvements.

Festival is specifically designed to allow the addition of new modules, easily and efficiently, so that development need not get bogged down in re-implementing the wheel.

But there is another aspect of Festival which makes it more useful than simply an environment for researching into new synthesis techniques. It is a fully usable text-to-speech system suitable for embedding in other projects that require speech output. The provision of a fully working easy-to-use speech synthesizer in addition to just a testing environment is good for two specific reasons. First, it offers a conduit for our research, in that our experiments can quickly and directly benefit users of our synthesis system. And secondly, in ensuring we have a fully working usable system we can immediately see what problems exist and where our research should be directed rather where our whims take us.

These concepts are not unique to Festival. ATR's CHATR system (black94) follows very much the same philosophy and Festival benefits from the experiences gained in the development of that system. Festival benefits from various pieces of previous work. As well as CHATR, CSTR's previous synthesizers, Osprey and the Polyglot projects influenced many design decisions. Also we are influenced by more general programs in considering software engineering issues, especially GNU Octave and Emacs on which the basic script model was based.

Unlike in some other speech and language systems, software engineering is considered very important to the development of Festival. Too often research systems consist of random collections of hacky little scripts and code. No one

person can confidently describe the algorithms it performs, as parameters are scattered throughout the system, with tricks and hacks making it impossible to really evaluate why the system is good (or bad). Such systems do not help the advancement of speech technology, except perhaps in pointing at ideas that should be further investigated. If the algorithms and techniques cannot be described externally from the program such that they can reimplemented by others, what is the point of doing the work?

Festival offers a common framework where multiple techniques may be implemented (by the same or different researchers) so that they may be tested more fairly in the same environment.

As a final word, we'd like to make two short statements which both achieve the same end but unfortunately perhaps not for the same reasons:

Good software engineering makes good research easier

But the following seems to be true also

If you spend enough effort on something it can be shown to be better than its competitors.

5.2 Future

Festival is still very much in development. Hopefully this state will continue for a long time. It is never possible to complete software, there are always new things that can make it better. However as time goes on Festival's core architecture will stabilise and little or no changes will be made. Other aspects of the system will gain greater attention such as waveform synthesis modules, intonation techniques, text type dependent analysers etc.

Festival will improve, so don't expected it to be the same six months from now.

A number of new modules and enhancements are already under consideration at various stages of implementation. The following is a non-exhaustive list of what we may (or may not) add to Festival over the next six months or so.

Selection-based synthesis: Moving away from diphone technology to more generalized selection of units for speech database.

New structure for linguistic content of utterances: Using techniques for Metrical Phonology we are building more structure representations of utterances reflecting there linguistic significance better. This will allow improvements in prosody and unit selection.

Non-prosodic prosodic control: For language generation systems and custom tasks where the speech to be synthesized is being generated by some program, more information about text structure will probably exist, such as phrasing, contrast, key items etc. We are investigating the relationship of high-level tags to prosodic information through the Sole project http://www.cstr.ed.ac.uk/projects/sole.html

Dialect independent lexicons: Currently for each new dialect we need a new lexicon, we are currently investigating a form of lexical specification that is dialect independent that allows the core form to be mapped to different dialects. This will make the generation of voices in different dialects much easier.

GOCR

Abstract:

In this documentation I describe some ideas for my OCR-program. It contains algorithms and examples and gives you an impression of what the program can (or could) do.

First I have to say that I am not a expert in pattern recognition or similar things. My knowledge is based mostly on experiments with my program. Therefore do not worry about stupid algorithms I put in this document. In this documentation I describe some ideas for my OCR-program. The examples give you an impression of how the program handles your images. If you have comments regarding contents or spelling please write to the author.

Segmentation of textual regions / Layout analysis

This is implemented as a recursive division in two parts.

look for the thickest horizontal or vertical gap through the box if the gap is less than five times longer than thick do not divide do the same with the two new parts I know that this algorithm is not as good as you wish, but I do not know a better one.

It would be very helpful to know about a function which is able to decide whether the box represents a single text line or a more complex object.

Line detection

Line detection is very importand for good recognition. For example it is difficult to distinguish between lowercase letter p and uppercase letter P without having a baseline (same total height). The lowercase version of p has a depht (the lower end is below the baseline) and therefore its easy to distinguish from the uppercase version if the baseline is known. The line detection is responsible for finding the baseline of every text line.

Lines of characters are detected by looking for interline spaces. These are characterized by a large number of non-black pixels in a row. Image rotation

(skewing) presents a problem, therefore the program first looks only at the left half of the image. When a line is found, the left half of the right side is scanned, because lines are often short. The variation in height gives an indication of the rotation angle. Using this angle, a second run detects lines more accurately. Line detection may fail if there is dust on the image.

In version v0.2.3 this behaviour is slightly changed. To detect the rotation angle, the line through the most characters is detected.

Cluster detection

A cluster is a group of pixels which are connected with each other. The simplest way to detect a cluster is to look for a pixel. If you find one, look to the neighbouring pixels. This can be done recursively.

This method needs a lot of stack space if a cluster is very large, and can cause problems with the memory.

Do you remember the algorithm for leaving a maze? Go along the right (or left) wall. This seems to be a good approach for detecting clusters without recursion. The following picture shows a trace of the maze algorithm.

first 35 steps next 36 steps

```
..@@@@@..@@@@<..   ..v<<<<..v<<<@..   * = starting point
..@@@@@@@@@.@^<.   ..>>v@^<<<@.@@@.   >^<v = go right,up,left,down
....@@@@@...@@^.   ....v@@@@...@@@.   @ = black pixel
....@@@@....@@^.   ....v@@@....@@@.
....@@@.....@@^.   ....v@@.....@@@.
....@@@.....@@^.   ....v@@.....@@@.
...@@@@.....@@^.   ...v<@@.....@@@.
...@@@......@@^.   ...v@@......@@@.
...@@@......@@^.   ...v@@......@@@.
...@@@.....@@@^.   ...v@@.....@@@@.
...@@@.....@@>^.   ...v@@.....@@@@.
...@@@.....@@^..   ...v@@.....@@@..
..@@@@.....@@^..   ..v<@@.....@@@..
..@@@@....@@@^..   ..v@@@....@@@@..
*>>>>>>>>>>>^<<   @@@@@@@@@@@@@@@@
```

The minimum and maximum coordinates can be used to create a box around the cluster. But does this algorithm work with diagonally connected pixels?

Engines

GOCR is able to work with different recognition engines. Since version 0.37 engines have to return a probability value together with the recognized character or a table of values to a table of characters. If the probability value is 100, the engine is 100% sure to have found the right character otherwise the value is less. This gives GOCR the possibility to compare results of different engines or in case of a not recognized character to inform the user or another application (spell checker) which characters probably could be there.

Base-Engine

The base engine (src/ocrX.c) is the original engine used in the first implementation of GOCR by Jörg. The idea was to get a fast and acceptable result without learning theoretical background. Later it should be replaced or completed by a better engine. The base engine is a rule based engine. The engine was written without theoretical background and is improved by try and error method but is is still far from perfect. The algorithm is very tolerant to size and form af characters (omnifont). How does the engine identify a character? For the explanation look at the following pattern.

```
vvvv        vv- white regions

......@@......  <- crossing one line

......@@......

.....@@@@.....

.....@@@@.....

.....@@@@.....

....@..@@@....  <- white hole / crossing two lines

....@..@@@....  <- crossing two lines

....@..@@@....

...@....@@@...

...@....@@@...

...@....@@@...

..@@@@@@@@@@..  <- horizontal line near center

..@......@@@..

..@......@@@..

.@........@@@.  v- increasing width of pattern

.@........@@@.  v

.@........@@@.  v

@@@......@@@@@

    ^^^-- gap
```

In the future the program should detect edges, vertices, gaps, angles and so on. This is called feature extraction (as far as I know). With such data the engine

could make a cluster analysis. But this is a difficult task, if the scanned image is noisy.

Database-Engine

The database engine (src/database.c) was the second engine added to GOCR. It was primary written to give users a simple tool to recognize special language-specific characters. The program generates a list (text file db.lst of image filenames and character codes) and image samples (pnm-files) in a database path (./db/). The database can be created by hand or extern programs or by GOCR itself using option (-m 130). In the last case GOCR prompts the user for not recognized characters. If he enters the character the pattern is saved in the database path as pnm-file and its file name is added to the database list (db.lst) together with the text string the pattern should be replaced by. For recognition GOCR first loads the database into memory (option -m 2). The main algorithm compares not recognized characters with stored images and calculates a distance value. If the distance value is small enough, the character is treated as recognized.

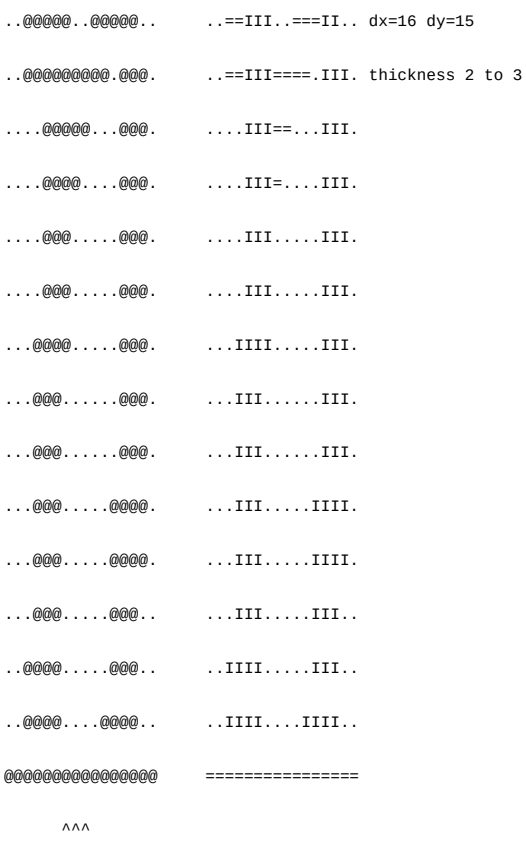
Remove pixels

The following picture shows an n which has additional pixels at the bottom. Therefore it can not be detected as n. What can be done?

classify horizontal (=) and vertical (I) pixels by comparing the distance between the next vertical and next horizontal white pixels (.)

measure mean thickness of vertical and horizontal clusters

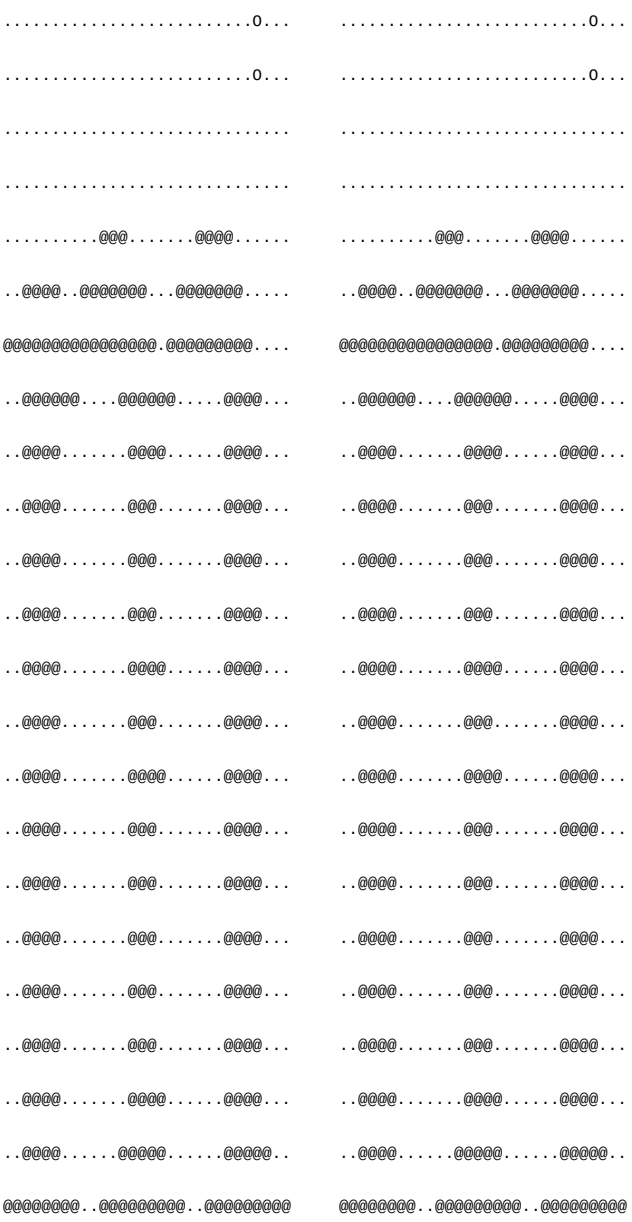
erase unusually thin horizontal pixels at the bottom line



this causes the problem

A better way is to find serifs (horizontal lines glued on the lower end of vertical lines) which touch together (v0.2.5).

The next picture shows blind pixels which are caused by dust on the paper. The upper right dots are not connected with the rest of the character. This can be detected via fill-algorithms. Currently the program assumes that dots near the upper end of a character are ``i''-dots or diaereses (umlaut dots).

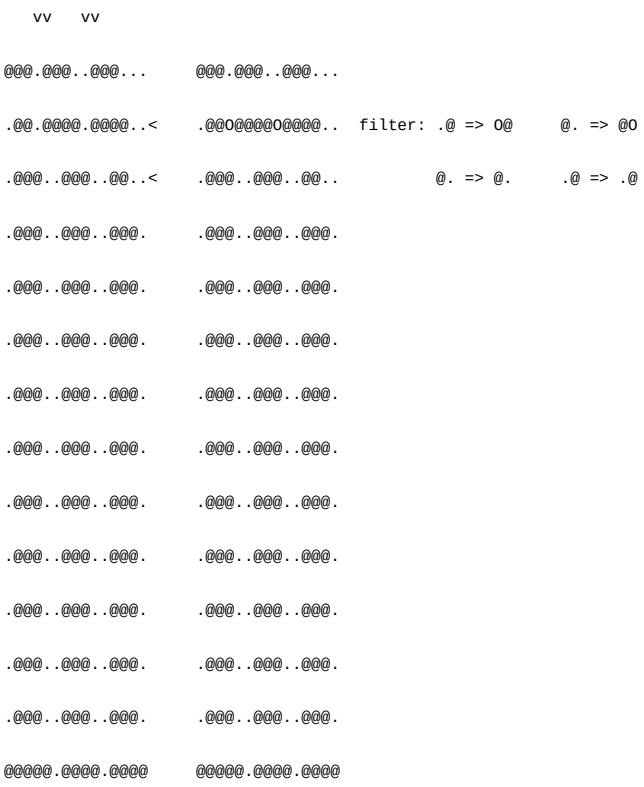


Add pixels

The following picture shows an m. The legs are only barely connected. How do we handle this?

if the engine has failed, a filter is switched on and the engine starts over

the 2x2 filter sets pixels to (O) near barely connected pixels



Similarity analyzer

Some characters are a little bit noisy. These characters can be identified by comparison with other, already recognized characters. This can be done via a good distance function. May be the distance function in the actual version of GOCR is not very good. Feel free to send me your ideas, but be sure it does not waste my time.

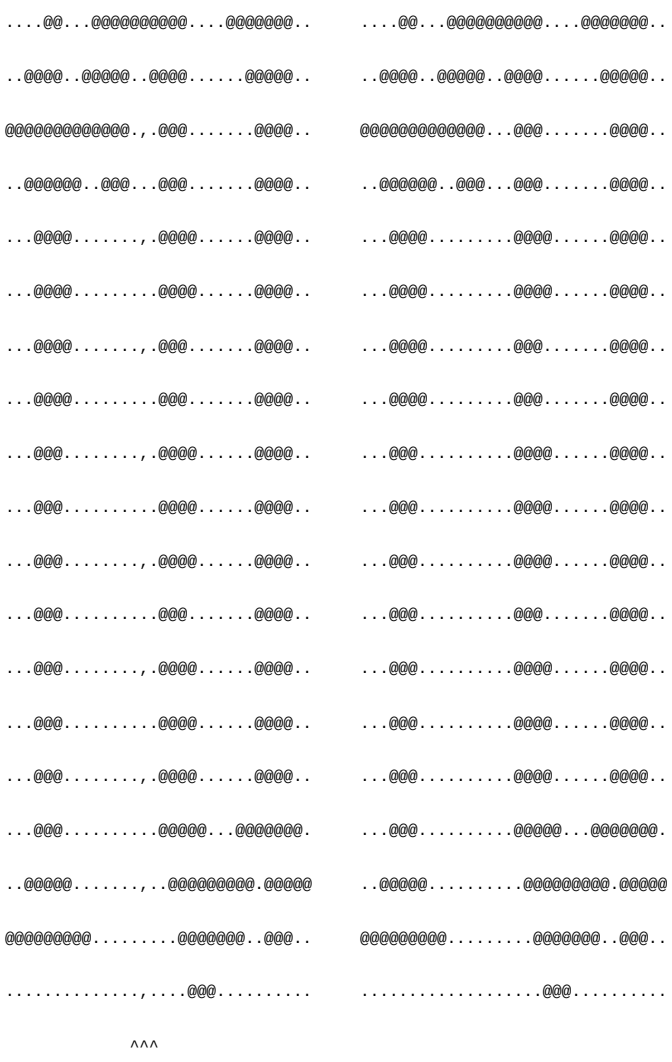
Overlapping characters

The following picture shows an overlapping ru. How do we handle this?

look for 3 weak connections (sum over y is small, start in the middle)

test if the right and left part can be detected by the engine

correction of surrounding box

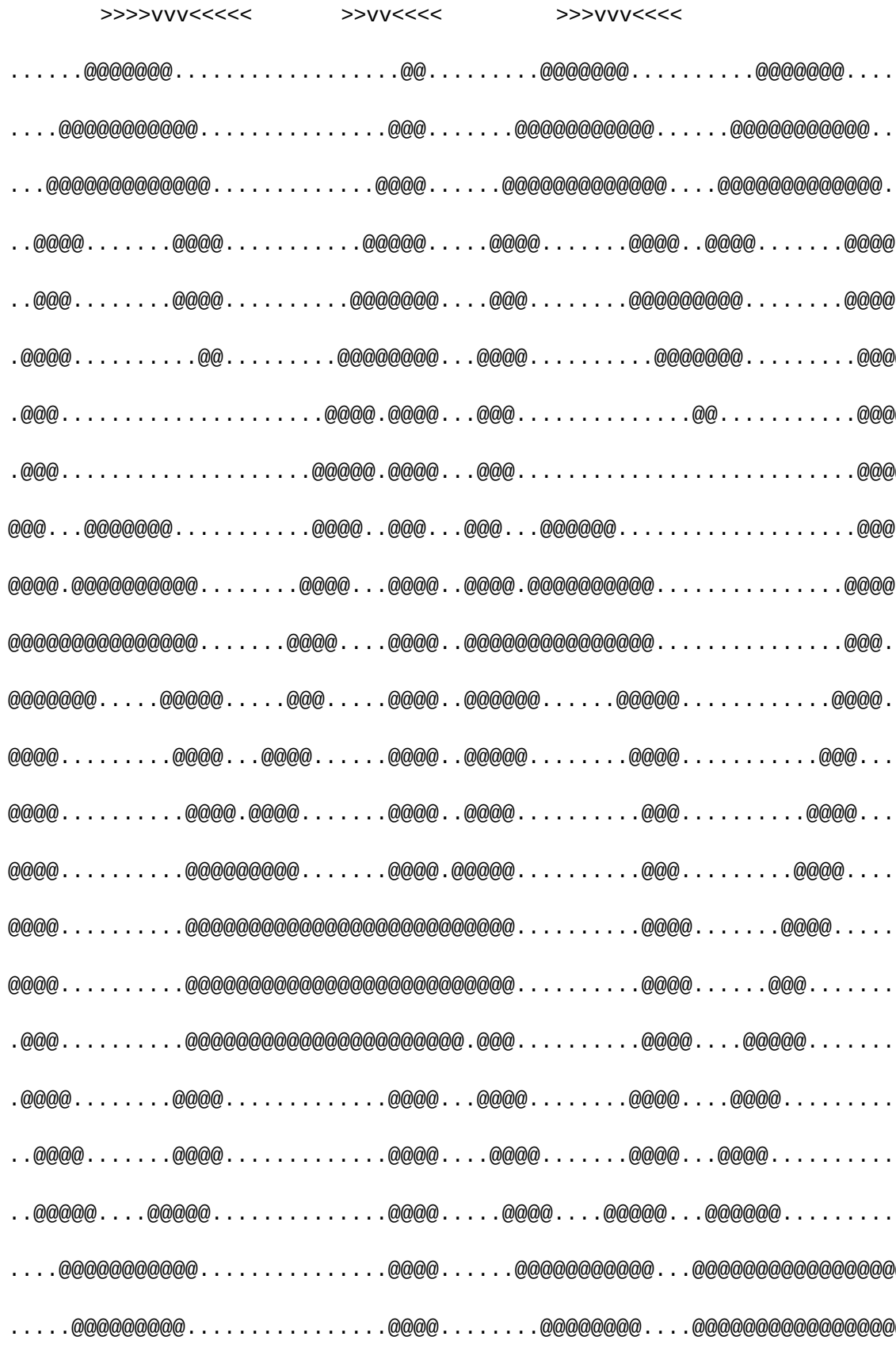


^^^

213 weak vertical lines

Of course the situation is more difficult with slanted characters.

The following example shows, how to deal with larger clusters. To get a fast program a first test should select the possible positions of division. That can be done by following upper and lower bows to a crease or a break. Than try to break on all detected creases, start at most important one (not implemented yet v0.2.4).


>>>>vvv<<<<<
>>vv<<<<
>>>vvv<<<<

```
........@@@@....................@@..........@@@@@.........@@@@@@@..@.@@@.
           >>>>^             ^<<>>^ ^<<<<<         >>>^<<<       ^^ ^
```

>,< show the path of the detection algorithm

The latest version of GOCR may use different algorithms. You have to look at the sources learn more.

Black/White, Gray and Colors

For simplicity colored images are converted to gray internally. That means a red text on green background will not be detected. You should use your own filter for this purpose.

If the original image is gray, a critical value is calculated to extract characters from the background. This can fail, if images are on the scanned page or tha scan is bad (dark edges or borders). It is difficult to overcome this problem because graylevels are mostly restricted to the 8 bit limit (16 bit would help to overcome this problem).

Black/White images are internally converted to gray with two levels (0 and 255).

The lowest 4 bits are not used, because they are used by internal functions (this can be changed in future).

After calculation of the threshold value (otsu.c) the brightness of every pixel is recalculated to a new internal threshold value of 160 (128+32). This is a bit above the middle of the 8 bit range. The idea is to make the live easier for the other

routines. Pixels which does not sure belong to the white or black ones get a value near the threshold value. Some routines can use this bit of more information to ignore outriders. Second point is, that this is necessary for using lowest for bits without destroying image informations.

Pictures on scanned pages

At first all objects on the scanned page are detected. Objects are clusters of black pixels. Pictures are detected if they are larger than 4 times the mean size of all objects. This rule is very simple and can fail some times. But it works fast and mostly the result is ok.

Tools

pbmclean:

This program is written by Angus Duggan and Jef Poskanzer. It cleans up ``snow'' on bitmap images.

pnmtools:

This tools are used to convert different image-formats to easy readable PNM (PBM,PGM,PPM) format. GOCR uses the popen-routine to call this programs if the suffix of the filename matches to a list in pnm.c. This will fail if pnmtools are missing.

related projects (to learn from)

unpaper:

unpaper - post-processing scanned and photocopied book pages, written by Jens Gulden 2005, GPL

glossary

font series:

bold, condensed

font shape:

normal, italic, slanted, sc...

points:

length unit used for font size, 1/72 inch, but I do not know its exact relation to the font size (height? totalheight? width? 10pt and 300dpi results in 40 pixel heigh font?)

sans serif:

font without the (often thin) lines on the ends of the character

descewing:

compensation of (slightly) rotated text

About this document

This Document was originaly written in LaTeX. In May 2002 Joerg has convertet it to HTML. The reason is, that you can read it now directly and you does not need to have LaTeX and Ghostscript installed on your computer to read it. As a side effect you do not need tetex package to build the gocr.rpm-package. A good viewer to read this document is lynx, links or w3m.

REFERENCES

[1] **CRBLP**. (2009). Centre for Research on Bangla Language Processing. [Accessed 16 December 2009]. Available from World Wide Web: <http://crblp.bracu.ac.bd>.

[2] **Wikipedia**. (2009). WordNet. [Accessed 16 December 2009]. Available from World Wide Web: < http://en.wikipedia.org/wiki/Wordnet>.

[3] **Wikipedia**. (2009). Natural language processing. [Accessed 16 December 2009]. Available from World Wide Web: <http://en.wikipedia.org/wiki/Natural_language_processing>.

[4] **Wikipedia**. (2009). Adobe Flash. [Accessed 16 December 2009]. Available from World Wide Web: <http://en.wikipedia.org/wiki/Adobe_Flash>.

[5] **Wikipedia**. (2009). PHP. [Accessed 16 December 2009]. Available from World Wide Web: < http://en.wikipedia.org/wiki/Php>.

[6] **Wikipedia**. (2009). WordNet. [Accessed 16 December 2009]. Available from World Wide Web: <http://en.wikipedia.org/wiki/Festival_Speech_Synthesis_System>.

[7] **Wikipedia**. (2009). Natural language processing. [Accessed 16 December 2009]. Available from World Wide Web: <http://en.wikipedia.org/wiki/Optical_character_recognition>.

[8] **Wikipedia**. (2009). Adobe Flash. [Accessed 16 December 2009]. Available from World Wide Web: <http://en.wikipedia.org/wiki/Speech_Recognition>.

[9] **Wikipedia**. (2009). PHP. [Accessed 16 December 2009]. Available from World Wide Web: <http://en.wikipedia.org/wiki/Speech_synthesis>.