import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

import java.io.\*;

public class Editor extends JFrame implements ActionListener

{

private JTextArea ta;

private int count;

private JMenuBar menuBar;

private JMenu fileM,editM,viewM, sizeM, fontM;

private JScrollPane scpane;

private JMenuItem exitI,cutI,copyI,pasteI,selectI,saveI,loadI,statusI, size1I, size2I, font1I, font2I;

private String pad;

private JToolBar toolBar;

public Editor()

{

super("Editor");

setSize(600, 600);

//setLocationRelativeTo(null);

// setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Container pane = getContentPane();

pane.setLayout(new BorderLayout());

count = 0;

pad = " ";

ta = new JTextArea(); //textarea

menuBar = new JMenuBar(); //menubar

fileM = new JMenu("File"); //file menu

editM = new JMenu("Edit"); //edit menu

viewM = new JMenu("View"); //edit menu

fontM = new JMenu("Font"); //edit menu

sizeM = new JMenu("Size"); //edit menu

scpane = new JScrollPane(ta); //scrollpane and add textarea to scrollpane

exitI = new JMenuItem("Exit");

cutI = new JMenuItem("Cut");

copyI = new JMenuItem("Copy");

pasteI = new JMenuItem("Paste");

selectI = new JMenuItem("Select All"); //menuitems

saveI = new JMenuItem("Save"); //menuitems

loadI = new JMenuItem("Load"); //menuitems

statusI = new JMenuItem("Status"); //menuitems

font1I = new JMenuItem("Arial");

font2I = new JMenuItem("Calibri");

size1I = new JMenuItem("increase 10");

size2I = new JMenuItem("decrease 10");

toolBar = new JToolBar();

ta.setLineWrap(true);

//ta.setWrapStyleWord(true);

setJMenuBar(menuBar);

menuBar.add(fileM);

menuBar.add(editM);

menuBar.add(viewM);

menuBar.add(fontM);

menuBar.add(sizeM);

fileM.add(saveI);

fileM.add(loadI);

fileM.add(exitI);

editM.add(cutI);

editM.add(copyI);

editM.add(pasteI);

editM.add(selectI);

viewM.add(statusI);

fontM.add(font1I);

fontM.add(font2I);

sizeM.add(size1I);

sizeM.add(size2I);

/\*saveI.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_S, ActionEvent.CTRL\_MASK));

loadI.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_L, ActionEvent.CTRL\_MASK));

cutI.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_X, ActionEvent.CTRL\_MASK));

copyI.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_C, ActionEvent.CTRL\_MASK));

pasteI.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_V, ActionEvent.CTRL\_MASK));

selectI.setAccelerator(KeyStroke.getKeyStroke(KeyEvent.VK\_A, ActionEvent.CTRL\_MASK));\*/

pane.add(scpane,BorderLayout.CENTER);

pane.add(toolBar,BorderLayout.SOUTH);

saveI.addActionListener(this);

loadI.addActionListener(this);

exitI.addActionListener(this);

cutI.addActionListener(this);

copyI.addActionListener(this);

pasteI.addActionListener(this);

selectI.addActionListener(this);

statusI.addActionListener(this);

font1I.addActionListener(this);

font2I.addActionListener(this);

size1I.addActionListener(this);

size2I.addActionListener(this);

setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

JMenuItem choice = (JMenuItem) e.getSource();

if (choice == saveI)

{

//not yet implmented

}

else if (choice == exitI)

System.exit(0);

else if (choice == cutI)

{

pad = ta.getSelectedText();

ta.replaceRange("", ta.getSelectionStart(), ta.getSelectionEnd());

}

else if (choice == copyI)

pad = ta.getSelectedText();

else if (choice == pasteI)

ta.insert(pad, ta.getCaretPosition());

else if (choice == selectI)

ta.selectAll();

else if (e.getSource() == statusI)

{

//not yet implmented

pad = ta.getSelectedText();

ta.insert(pad, ta.getCaretPosition());

}

else if (e.getSource() == font1I)

{

System.out.println("clicked font menu");

Font font = ta.getFont();

float size = font.getSize();

Font myFont = new Font("Arial", Font.ITALIC | Font.BOLD, 14);

ta.setFont( myFont.deriveFont(size) );

}

else if (e.getSource() == font2I)

{

Font font = ta.getFont();

float size = font.getSize() + 10.0f;

Font myFont = new Font("Calibri", Font.BOLD, 14);

ta.setFont( myFont.deriveFont(size) );

}

else if (e.getSource() == size1I)

{

Font font = ta.getFont();

float size = font.getSize() + 10.0f;

ta.setFont( font.deriveFont(size) );

}

else if (e.getSource() == size2I)

{

Font font = ta.getFont();

float size = font.getSize() - 10.0f;

ta.setFont( font.deriveFont(size) );

}

}

public static void main(String[] args)

{

new Editor();

}

}

import javax.swing.\*;

import java.awt.event.\*;

import java.awt.\*;

import java.io.\*;

import java.net.\*;

public class DragAndDrop extends JFrame

{

JButton button;

JTextField textfield;

JFileChooser filechooser;

JTextArea textarea;

Socket socket;

FileTransferClient filetransferclient;

public DragAndDrop()

{

try{

socket = new Socket("127.0.0.1",3000);

filetransferclient = new FileTransferClient(socket);

}

catch(Exception e){e.printStackTrace();}

setTitle("Drag and Drop Client");

button = new JButton("Send");

filechooser = new JFileChooser();

filechooser.setDragEnabled(true);

textarea=new JTextArea(100,100);

textarea.setBackground(new Color(89,34,12));

textarea.setForeground(Color.WHITE);

textarea.setTransferHandler(new TransferHandler("text"));

button.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e)

{

String filepath = textarea.getText();

filetransferclient.send\_file(filepath);

}});

add(textarea,"Center");

add(filechooser,"North");

add(button,"South");

setSize(500,500);

setBackground(Color.GREEN);

setVisible(true);

}

public static void main(String args[])

{

DragAndDrop dnd = new DragAndDrop();

}

}

class FileTransferClient

{

Socket socket;

DataOutputStream dout;

FileTransferClient(Socket s)

{

try

{

socket = s;

dout = new DataOutputStream(socket.getOutputStream());

}

catch(Exception e)

{

e.printStackTrace();

}

}

public void send\_file(String filepath)

{

try

{

dout.writeUTF("SEND");

String[] parts = filepath.split("\\\\");

int n = parts.length;

dout.writeUTF(parts[n-1]);

File file= new File(filepath);

if(file.exists())

{

FileInputStream fis = new FileInputStream(file);

int ch;

do

{

ch = fis.read();

dout.writeUTF(String.valueOf(ch));

}

while(ch!=-1);

fis.close();

}

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

import javax.swing.\*;

import java.awt.\*;

import java.net.\*;

import java.io.\*;

public class DragAndDropServer extends JFrame

{

JTextArea textarea;

DragAndDropServer()

{

setTitle("Drag and Drop Server");

textarea = new JTextArea();

textarea.setBackground(new Color(150,200,243));

add(textarea,"Center");

setSize(500,500);

setVisible(true);

}

public static void main(String args[]) throws Exception

{

ServerSocket ss = new ServerSocket(3000);

DragAndDropServer dndserver = new DragAndDropServer();

dndserver.textarea.append("Waiting for Client...");

while(true)

{

TransferFile tf = new TransferFile(ss.accept(),dndserver);

}

}

}

class TransferFile extends Thread

{

DataInputStream din;

FileOutputStream fileout;

Socket socket;

DragAndDropServer dndserver;

TransferFile(Socket s, DragAndDropServer dndserver)

{

try{

this.dndserver = dndserver;

dndserver.textarea.append("\nClient connected..");

socket = s;

din = new DataInputStream(socket.getInputStream());

start();

}

catch(Exception e){e.printStackTrace();}

}

public void run()

{

try{

while(true)

{

if((din.readUTF()).equals("SEND"))

{

dndserver.textarea.append("\nSEND command received.");

receive();

}

}}

catch(Exception e){e.printStackTrace();}

}

public void receive()

{

try{

String filename = din.readUTF();

dndserver.textarea.append("\nFile name : "+filename);

File file = new File(filename);

fileout = new FileOutputStream(file);

String str;int ch;

do

{

str = din.readUTF();

ch = Integer.parseInt(str);

if(ch!=-1)

fileout.write(ch);

}

while(ch!=-1);

dndserver.textarea.append("\nFile received successfully");

}

catch(Exception e)

{

e.printStackTrace();

}

}

}

<?php

$window = new GtkWindow();

$window->set\_title('REGISTRATION');

$window->set\_size\_request(500,400);

$window->modify\_bg(Gtk::STATE\_NORMAL, GdkColor::parse("#7591cf"));

$window->connect\_simple('destroy', array('Gtk','main\_quit'));

$window->add($vbox = new GtkVBox());

// display title

$title = new GtkLabel("SAMPLE REGISTRATION FORM");

$title2 = new GtkLabel("\*SELECT USER NAME FROM THE LIST , ENTER PASSWORD AND THE GENDER");

$title->modify\_font(new PangoFontDescription("Times New Roman Italic 10"));

$title->modify\_fg(Gtk::STATE\_NORMAL, GdkColor::parse("#0000ff"));

$title->set\_size\_request(-1, 40);

$title2->modify\_font(new PangoFontDescription("Times New Roman Italic 10"));

$title2->modify\_fg(Gtk::STATE\_NORMAL, GdkColor::parse("#ee3300"));

$title2->set\_size\_request(-1, 60);

$vbox->pack\_start($title, 0, 0);

$vbox->pack\_start($title2, 0, 0);

$fields = array('REGISTRATION NUMBER', 'USERNAME', 'PASSWORD', 'GENDER');

$field\_size = array(120, 200, 80, 80);

$entry = array(); // to store the text entries

$table = new GtkTable();

display\_table($table, $fields, $field\_size); // display the table

$vbox->pack\_start($table);

function display\_table($table, $fields, $field\_size) {

global $entry;

$row = 0;

foreach ($fields as $field) {

$label = new GtkLabel(" $field: ");

$alignment = new GtkAlignment(1, .5, 0, 0);

$alignment->add($label);

$table->attach($alignment, 0, 1, $row, $row+1, Gtk::FILL, Gtk::SHRINK, 0, 0);

$entry[$row] = new GtkEntry();

$alignment = new GtkAlignment(0, .5, 0, 0);

$alignment->add($entry[$row]);

$entry[$row]->set\_size\_request($field\_size[$row], -1);

if ($field=='REGISTRATION NUMBER') {

$hbox = new GtkHBox();

// add a button to popup list after the label

$button = new GtkButton('Select');

$hbox->pack\_start($alignment, 0, 0);

$hbox->pack\_start(new GtkLabel(' '), 0, 0);

$hbox->pack\_start($button, false);

$widget = &$hbox;

} else {

$widget = $alignment;

}

$table->attach($widget, 1, 2, $row, $row+1);

++$row;

}

$button->connect('clicked', 'popup\_list', $entry);

}

// create a submit button

$button = new GtkButton('REGISTER');

$button->set\_size\_request(60, 28);

$button->connect('clicked', 'on\_click');

$row = count($fields);

$alignment = new GtkAlignment(0, 0.5, 0, 0);

$alignment->add($button);

$table->attach($alignment, 1, 2, $row, $row+1);

$window->show\_all();

Gtk::main();

function on\_click($button) {

global $fields, $entry;

$i=0;

foreach($fields as $field) {

echo "$field: ".$entry[$i]->get\_text()."\n";

++$i;

}

}

function popup\_list($button, $entry) {

$selection = prompt();

if (count($selection)>0) {

for ($i=0; $i<4; ++$i) {

$entry[$i]->set\_text($selection[$i]);

}

}

}

function prompt() {

$prompt = new Prompt();

$selection = array();

if ($prompt->selected\_iter!=null) {

$model = $prompt->model;

for ($i=0; $i<4; ++$i) {

$selection[] = $model->get\_value($prompt->selected\_iter, $i);

}

}

return $selection;

}

class Prompt{

var $selected\_iter = null;

function Prompt() {

$dialog = new GtkDialog('Select', null, Gtk::DIALOG\_MODAL|Gtk::DIALOG\_NO\_SEPARATOR);

$dialog->vbox->pack\_start(new GtkLabel(), 0, 0);

$dialog->vbox->pack\_start(new GtkLabel('Select item with arrow keys and press Return.'), 0, 0);

$dialog->vbox->pack\_start(new GtkLabel('Or select by double-click the desired row.'), 0, 0);

$dialog->vbox->pack\_start(new GtkLabel(), 0, 0);

// Set up a scroll window

$scrolled\_win = new GtkScrolledWindow();

$scrolled\_win->set\_policy( Gtk::POLICY\_AUTOMATIC, Gtk::POLICY\_AUTOMATIC);

$dialog->vbox->pack\_start($scrolled\_win);

// the 2D table

/\*$data = array(

array('2013506050','SUKANYA','Enter Password......','Enter Gender.......'),

array('2013506020','JAYA','Enter PAssword.....','Enter Gender.....'),

array('2013505051','SURIYA','Enter Password','Enter Gender'),

array('2013506000','JOHN','Enter Password','Enter Gender'));

\*/

$data = array(

array('2013506050','SUKANYA',1,1),

array('2013506020','JAYA',1,1),

array('2013505051','SURIYA',1,1),

array('2013506000','JOHN',1,1));

$this->display\_table($scrolled\_win, $data);

$this->dialog = $dialog;

$dialog->connect('key-press-event', array($this, 'on\_key'));

$dialog->show\_all();

$dialog->run();

$dialog->destroy();

}

function display\_table($scrolled\_win, $data) {

// Creates the list store

if (defined("GObject::TYPE\_STRING")) {

$model = new GtkListStore(GObject::TYPE\_STRING, GObject::TYPE\_STRING,

GObject::TYPE\_LONG, GObject::TYPE\_DOUBLE);

} else {

$model = new GtkListStore(Gtk::TYPE\_STRING, Gtk::TYPE\_STRING,

Gtk::TYPE\_LONG, Gtk::TYPE\_DOUBLE);

}

$this->model = $model;

$field\_header = array('REG.NO','USERNAME');

$field\_justification = array(0.5, 0.0);

// Creates the view to display the list store

$view = new GtkTreeView($model);

$this->view = $view;

$scrolled\_win->add($view);

$scrolled\_win->set\_size\_request(300, 186);

// Creates the columns

for ($col=0; $col<count($field\_header); ++$col) {

$cell\_renderer = new GtkCellRendererText();

$cell\_renderer->set\_property("xalign", $field\_justification[$col]);

$column = new GtkTreeViewColumn($field\_header[$col], $cell\_renderer, 'text', $col);

$column->set\_alignment($field\_justification[$col]);

$column->set\_sort\_column\_id($col);

// set the header font and color

$label = new GtkLabel($field\_header[$col]);

$label->modify\_font(new PangoFontDescription("Arial Bold"));

$label->modify\_fg(Gtk::STATE\_NORMAL, GdkColor::parse("#0000FF"));

$column->set\_widget($label);

$label->show();

// setup self-defined function to display alternate row color

$column->set\_cell\_data\_func($cell\_renderer, array($this, 'format\_col'), $col);

$view->append\_column($column);

}

for ($row=0; $row<count($data); ++$row) {

$values = array();

for ($col=0; $col<count($data[$row]); ++$col) {

$values[] = $data[$row][$col];

}

$model->append($values);

}

$view->connect('button-press-event', array($this, 'on\_button')); // note 1

}

function format\_col($column, $cell, $model, $iter, $col\_num) {

$path = $model->get\_path($iter);

$row\_num = $path[0];

if ($col\_num==3) {

$amt = $model->get\_value($iter, 3);

$cell->set\_property('text', '$'.number\_format($amt,2));

}

$row\_color = ($row\_num%2==1) ? '#dddddd' : '#ffffff';

$cell->set\_property('cell-background', $row\_color);

}

function on\_selection($selection) {

list($model, $iter) = $selection->get\_selected();

$desc = $model->get\_value($iter, 1);

$qty = $model->get\_value($iter, 2);

$price = $model->get\_value($iter, 3);

print "You have selected $desc: $qty ($price)\n";

}

function on\_key($widget, $event) {

print "key = $event->keyval\n";

if ($event->keyval==65307) {

// Gdk::KEY\_Escape

$this->dialog->destroy();

return true;

}

if ($event->keyval!=65293) return false;

// 65293 = Gdk::KEY\_Return

$selection = $this->view->get\_selection();

list($model, $iter) = $selection->get\_selected();

if ($iter==NULL) {

echo("Please select an item first.");

return true;

}

$this->selected\_iter = $iter;

$this->dialog->destroy();

}

function on\_button($widget, $event) {

if ($event->type==Gdk::\_2BUTTON\_PRESS) { // note 2

$selection = $this->view->get\_selection();

list($model, $iter) = $selection->get\_selected();

if ($iter==NULL) { // note 3

echo("Please select an item first.");

return true;

}

$this->selected\_iter = $iter; // note 4

$this->dialog->destroy();

}

}

}

?>

<?php

// function to simplify the display of grouped radio buttons

function setup\_radio($radio\_button\_grp, $button\_label, $button\_value) { // note 3

$radio = new GtkRadioButton($radio\_button\_grp, $button\_label);

$radio->connect('toggled', "on\_toggle", $button\_value); // note 4

return $radio;

}

// call-back function when user pressed a radio button

function on\_toggle($radio, $value) { // note 5

global $status\_area;

$label = $radio->child->get\_label(); // note 6

$active = $radio->get\_active(); // note 7

if ($active) $status\_area->set\_text("radio button pressed: $label (value = $value)\n"); // note 8

}

function login(GtkWindow $wnd, GtkEntry $txtUsername, GtkEntry $txtPassword,GtkEntry

$txtRegNo,GtkEntry $txtDateofBirth,GtkEntry $txtMailID)

{

$strUsername = $txtUsername->get\_text();

$strPassword = $txtPassword->get\_text();

$strRegNo=$txtRegNo->get\_text();

$strDateofBirth=$txtDateofBirth->get\_text();

$strMailID=$txtMailID->get\_text();

$errors = null;

if($strUsername=="SUKANYA"&& $strPassword=="sukanya"&&

$strRegNo=="2013506050"&& $strDateofBirth=="99.99.9999"&&

$strMailID=="sukanya@gmail.com")

{

$dialog = new GtkMessageDialog($wnd,

Gtk::DIALOG\_MODAL,Gtk::MESSAGE\_INFO, Gtk::BUTTONS\_NONE, None);

$dialog->set\_markup("RegNo".$strRegNo."\nName:".$strUsername."\nSUCCESSFULLY REGISTERED\n");

$dialog->run();

$dialog->destroy();

}

else

{

$dialog = new GtkMessageDialog($wnd,

Gtk::DIALOG\_MODAL,Gtk::MESSAGE\_ERROR, Gtk::BUTTONS\_OK, $errors);

$dialog->set\_markup("AUTHORIZATION FAILED,CHECK THE CREDENTIALS ENTERED\n");

$dialog->run();

$dialog->destroy();

}

}

$wnd = new GtkWindow();

$wnd->set\_title('REGISTRATION');

$wnd->set\_size\_request(500,400);

$wnd->modify\_bg(Gtk::STATE\_NORMAL, GdkColor::parse("#7591cf"));

$wnd->connect\_simple('destroy', array('gtk', 'main\_quit'));

$lblCredit = new GtkLabel('SIMPLE REGISTRATION FORM');

$lblCredit->modify\_font(new PangoFontDescription("Times New Roman Italic 10"));

$lblCredit->modify\_fg(Gtk::STATE\_NORMAL, GdkColor::parse("#0000ff"));

$lblCredit->set\_size\_request(-1, 40);

/\*

$pixbuf=GdkPixbuf::new\_from\_file("bg.png");

list($pixmap,$mask)=$pixbuf-> render\_pixmap\_and\_mask(255);

$style = $wnd->get\_style();

$style=$style->copy();

$style->bg\_pixmap[Gtk::STATE\_NORMAL]=$pixmap;

$wnd->set\_style($style);

$vbox->pack\_start($lblCredit, 0);

$vbox->pack\_start(new GtkLabel(), 0);

\*/

$lblRegNo=new GtkLabel('RegNo',true);

$lblUsername = new GtkLabel('UserName', true);

$lblPassword = new GtkLabel('Password', true);

$lblDateofBirth=new GtkLabel('Date of Birth',true);

$title1 = new GtkLabel();

$title1->set\_markup('<span color="blue" font\_desc="Times New Roman Italic 12">

GENDER</span>');

// setup grouped radio buttons

$radio1 = setup\_radio(null, 'MALE', '101'); // note 1

$radio2 = setup\_radio($radio1, 'FEMALE', '102'); // note 2

//$radio3 = setup\_radio($radio1, 'radio button 3', '103');

$lblMailID=new GtkLabel('EmailID',true);

$lblCredit->modify\_font(new PangoFontDescription('Eras Demi ITC 20'));

$lblRegNo->modify\_font(new PangoFontDescription('Eras Demi ITC 20'));

$lblUsername->modify\_font(new PangoFontDescription('Eras Demi ITC 20'));

$lblPassword->modify\_font(new PangoFontDescription('Eras Demi ITC 20'));

$lblDateofBirth->modify\_font(new PangoFontDescription('Eras Demi ITC 20'));

$lblMailID->modify\_font(new PangoFontDescription('Eras Demi ITC 20'));

$txtRegNo=new GtkEntry();

$txtUsername = new GtkEntry();

$txtPassword = new GtkEntry();

$txtDateofBirth=new GtkEntry();

$txtMailID=new GtkEntry();

$btnLogin = new GtkButton('REGISTER');

$btnCancel = new GtkButton('CANCEL');

$lblRegNo->set\_mnemonic\_widget($txtRegNo);

$lblUsername->set\_mnemonic\_widget($txtUsername);

$lblPassword->set\_mnemonic\_widget($txtPassword);

$lblDateofBirth->set\_mnemonic\_widget($txtDateofBirth);

$lblMailID->set\_mnemonic\_widget($txtMailID);

$btnCancel->connect\_simple('clicked', array($wnd, 'destroy'));

$btnLogin->connect\_simple('clicked', 'login', $wnd ,$txtUsername,

$txtPassword,$txtRegNo,$txtDateofBirth,$txtMailID);

$tbl = new GtkTable(11, 2);

$tbl->attach($lblCredit, 0, 2, 0, 1);

$tbl->attach($lblRegNo, 0, 1, 1, 2);

$tbl->attach($txtRegNo, 1, 2, 1, 2);

$tbl->attach($lblUsername, 0, 1, 2, 3);

$tbl->attach($txtUsername, 1, 2, 2, 3);

$tbl->attach($lblPassword, 0, 1, 3, 4);

$tbl->attach($txtPassword, 1, 2, 3, 4);

$tbl->attach($lblDateofBirth,0,1,4,5);

$tbl->attach($txtDateofBirth,1,2,4,5);

$tbl->attach($lblMailID,0,1,5,6);

$tbl->attach($txtMailID,1,2,5,6);

$bbox = new GtkHButtonBox();

$bbox->set\_layout(Gtk::BUTTONBOX\_EDGE);

$bbox->add($btnCancel);

$bbox->add($btnLogin);

$vbox = new GtkVBox();

$vbox->pack\_start($tbl);

$vbox->pack\_start($bbox);

$wnd->add($vbox);

$wnd->show\_all();

Gtk::main();

?>

<html>

<head>

<link rel = "stylesheet" href = "https://www.rawgit.com/jsapilib/jsapi/master/masking.css" type="text/css">

</head>

<body>

<section>

<button onclick="trigger('1')">Mask 1</button>

<button onclick="trigger('2')">Mask 2</button>

<button onclick="trigger('3')">Mask 3</button>

</section>

<br>

<br>

<br>

<section class="masks">

<div class="image-mask">

<div id="mask1" class="mask"></div>

<img src="image1.jpg">

</div>

<div class="image-mask">

<div id="mask2" class="mask"></div>

<img src="image2.jpg">

</div>

<div class="image-mask">

<div id="mask3" class="mask"></div>

<img src="image3.jpg">

</div>

</section>

<!-- Always put javascript at bottom -->

<script>

function trigger(str) {

var el = document.getElementById('mask' + str);

var clname = el.className;

el.className +=' anim-class' + str;

/\* remove class after 6 seconds so that we can retrigger using button click \*/

setTimeout(function() {el.className = clname;}, 6000);

}

</script>

</body>

</html>

<html>

<head>

<link rel="stylesheet" href="https://rawgit.com/jsapilib/jsapi/master/ammap.css" type="text/css">

<script src="https://rawgit.com/jsapilib/jsapi/master/ammap.js" type="text/javascript"></script>

<script src="https://rawgit.com/jsapilib/jsapi/master/worldLow.js" type="text/javascript"></script>

<script>

AmCharts.makeChart("mapdiv", {

type: "map",

dataProvider: {

map: "worldLow",

getAreasFromMap:true,

},

});

</script>

</head>

<body>

<div id="mapdiv" style="width: 100%; background-color:#eeeeee; height: 500px;"></div>

</body>

</html>

<html>

<title>Hide and Seek</title>

<head>

<script type="text/javascript">

count=0;

function check(){

var x=document.getElementById('ans').value;

var y=document.getElementById('txtarea');

switch(count) {

case 0: if(x=="bmw") {

count++;

var z=document.getElementById('one');

z.style.opacity=0.0;

y.innerHTML="Name the Car";

document.getElementById('ans').value=''; }

else{alert("You are close to answer");}

break;

case 1: if(x=="benz") {

count++;

var z=document.getElementById('two');

z.style.opacity=0.0;

y.innerHTML="Name the Car";

document.getElementById('ans').value=''; }

else{alert("You are close to answer");}

break;

case 2: if(x=="audi") {

count++;

var z=document.getElementById('three');

z.style.opacity=0.0;

y.innerHTML="Name the Car";

document.getElementById('ans').value=''; }

else{alert("You are close to answer");}

break;

case 3: if(x=="maruthi") {

count++;

var z=document.getElementById('four');

z.style.opacity=0.0;

y.innerHTML="Name the Car";

document.getElementById('ans').value=''; }

else{alert("You are close to answer");}

break;

case 4: if(x=="audi") {

count++;

var z=document.getElementById('five');

z.style.opacity=0.0;

y.innerHTML="Name the Car";

document.getElementById('ans').value=''; }

else{alert("You are close to answer");}

break;

case 5: if(x=="jaguar") {

count++;

var z=document.getElementById('six');

z.style.opacity=0.0;

y.innerHTML="Name the Car";

document.getElementById('ans').value=''; }

else{alert("You are close to answer");}

break;

case 6: if(x=="lamborghini") {

count++;

var z=document.getElementById('seven');

z.style.opacity=0.0;

y.innerHTML="Name the Car";

document.getElementById('ans').value=''; }

else{alert("You are close to answer");}

break;

case 8: if(x=="bmw") {

count++;

var z=document.getElementById('nine');

z.style.opacity=0.0;

y.innerHTML="Perfect guess. Yes, its KABALI";

var tab=document.getElementById('table1');

document.getElementById('msg').style.opacity=1.0;

document.getElementById('msg1').style.opacity=1.0;

document.getElementById('ans').value=''; }

else{alert("Try Again");}

break; } }

</script>

</head>

<body bgcolor="#1A8072">

<center>

<div id="title">

<font color="beige" id="fontstyle" face="timesnewroman" size="20">Make it!!!!</center></font>

</div><br>

<center>

<div id="main">

<image id="image" src="image.jpg">

<table id="table1"><center> <tr >

<td id="one"><img src="tile1.jpg" width=183 height=122></td>

<td id="two"><img src="tile2.jpg" width=183 height=122></td>

<td id="three"><img src="tile3.jpg" width=183 height=122></td>

</tr> <tr >

<td id="seven"><img src="tile7.jpg" width=183 height=122></td>

<td id="eight"><img src="tile8.jpg" width=183 height=122></td>

<td id="nine"><img src="tile9.jpg" width=183 height=122></td>

</tr></center>

</table>

<div id="qa"> <p>

<textarea id="txtarea" rows="3" cols="30" readonly> Name the car.</textarea>

</p>

<br>Answer:<input type="text" id="ans" >

<input type="submit" text="Submit" onclick="check()">

</div>

<font id="msg" face="arial">You Make It!!!</font>

<font id="msg1" face="arial"> ...Well done... </font>

</div>

</body> </html>