*// La fonction utt permet de calculer la distance entre deux villes parmi les 35249 villes de la base de données, en assimilant la terre à une sphère.* *On possède 2 vecteurs : SST contient le nom de villes dans l'ordre alphabétique. Value contient les coordonnées GPS des villes, dans le même ordre que SST.* *La 1ere colonne de Value contient les latitudes, la 2ème colonne les longitudes. La 3ème colonne indique si la ville est un doublon, c'est à dire s’il existe* *plusieurs villes qui ont le même nom. Cette colonne ne contient que des zéros sauf pour les villes dont la suivante sur la liste possède le même nom, dans ce cas la valeur de la 3ème colonne est 1.*

*//Chargement de la base de données.*

*//Ouverture de la liste des villes contenue dans Mat2.dat*

fd=mopen('C:\Users\thibault\Desktop\UTT\Mat2.Dat','rb');

load(fd,"SST");

mclose(fd);

*//Ouverture de la liste des latitudes contenue dans lat*

X=fscanfMat('C:\Users\thibault\Desktop\UTT\lat');

*//Ouverture de la liste des longitudes contenue dans long.dat*

fd=mopen('C:\Users\thibault\Desktop\UTT\long.Dat','rb');

load(fd,"Y");

mclose(fd);

*//Ouverture de la liste des doublons contenue dans doublon.*

Z=fscanfMat('C:\Users\thibault\Desktop\UTT\doublon');

*//Création de la matrice Value : [Latitude, Longitude, Doublon].*

Value(:,1)=X;

Value(:,2)=Y;

Value(:,3)=Z;

function [**nville1**]=cleaner(**ville1**) *//Remplace tous les caractères non supportés par scilab*

**nville1**=strsubst(**ville1**,"é","e"); **nville1**=strsubst(**nville1**,"è","e"); **nville1**=strsubst(**nville1**,"à","a");

**nville1**=strsubst(**nville1**,"ç","c"); **nville1**=strsubst(**nville1**,"-","");

**nville1**=strsubst(**nville1**,"â","a"); **nville1**=strsubst(**nville1**,"ê","e");

**nville1**=strsubst(**nville1**,"î","i"); **nville1**=strsubst(**nville1**,"ô","o");

**nville1**=strsubst(**nville1**,"û","u"); **nville1**=strsubst(**nville1**,"ù","u");

**nville1**=strsubst(**nville1**,"ö","o"); **nville1**=strsubst(**nville1**,"ü","u");

**nville1**=strsubst(**nville1**,"ï","i"); **nville1**=strsubst(**nville1**," ","");

**nville1**=strsubst(**nville1**,"''",""); **nville1**=convstr(**nville1**,'l');

endfunction

function [**rang**]=finder(**V**)*//Trouve le rang de la ville dans la liste (SST) à partir de son nom, sort 0 si la ville n’existe pas.*

test=%F;n=max(size(SST));

**rang**=1;*//initialisation*

while test==%F&**rang**<=n do test=SST(**rang**)==**V** *// on sort de la boucle si la ville est trouvé*

**rang**=**rang**+1 *// on incrémente le rang de +1*

end

if test==%F&**rang**==n+1 then **rang**=0 *// si le test est faux on sort 0 dans rang car la ville n’a pas été trouvée.*

else **rang**=**rang**-1 *//Pour obtenir le bon rang.*

end

endfunction

function [**coord**]=transform(**coord**)

*//Convertit les cordonnées latitude/longitude en pixels pour correspondre à la carte de France*

*// la carte est volontairement déformé pour qu'il y'ait proportionnalité entre lat/long et pixels*

n=length(**coord**(:,1))

h=386 *//hauteur*

w=393 *//largeur*

maxlat=51.1

minlat=42.3

maxlong=8.3

minlong=-5.2

for i=1:n do **coord**(i,1)=floor((**coord**(i,1)-minlat)\*h/(maxlat-minlat)) *//il s'agit d’une règle de 3*

**coord**(i,2)=floor((**coord**(i,2)-minlong)\*w/(maxlong-minlong))

end

endfunction

function [**y**]=doublon(**rang**) *// Si la ville est un doublon on affiche sur la carte de france tous les doublons et le choix se fait par un clic de souris.*

j=1

coord(j,1)=Value(rang,1); coord(j,2)=Value(rang,2)

while Value(rang,3)==1 do j=j+1;

rang=rang+1;

coord(j,1)=Value(rang,1); coord(j,2)=Value(rang,2)

*// On fabrique une matrice contenant toutes les coordonnées des villes du même nom.*

end

RGB = ReadImage('C:\Users\thibault\Desktop\UTT\france1.jpg');

[image, ColorMap] = RGB2Ind(RGB);

FigureHandle = ShowImage(image, 'Example', ColorMap); *//affiche l'image de la France en couleur !!! nécessite Image Processing Design !!!*

coordIm=transform(coord) *//coordIm est le transformé des coordonnées latitude et longitude en des coordonnées adapté à l'image*

n=length(coord(:,1)); *//nombre de ville du même nom (au moins 2)*

for **i**=1:n do xstring(coordIm(**i**,2),coordIm(**i**,1),**ville**)

 xstring(coordIm(**i**,2),coordIm(**i**,1),"+") *// on affiche l’emplacement des différentes villes sur la carte de France*

end

Clic=xclick() *// Clic contient les coordonnées en pixel du clic de souris*

D=999999;d=0;**y**=0;compt=1

for **i**=1:n do d=(coordIm(**i**,2)-Clic(2))^2+(coordIm(**i**,1)-Clic(3))^2;

if d<D then D=d;

**y**=coord(**i**,1);

**y**(1,2)=coord(**i**,2);

compt=**i**;

end

end

*// y contient les coordonnées en lat et long de la ville la plus proche du clic*

xstring(coordIm(compt,2),coordIm(compt,1),**ville**,0,1)

endfunction

function [**y**, **ville**]=bonneville(**i**, **texte**)

mess=["Entrez le nom de la '+string(**i**)+string(**texte**)+' ville"];**ville**=x\_mdialog('Nom de la '+string(**i**)+string(**texte**)+' ville ',mess,['']); nville=cleaner(**ville**);rang=finder(nville); *// on cherche l'indice de la ville dans SST*

while rang==0 do messagebox("La ville n''est pas référencée dans notre base de données. Vérifiez l''orthographe, on ne sait jamais!", "Il y a un problème!", "warning", "Corriger le nom","modal");

mess=['Nouveau nom']; **ville**=x\_mdialog('Correction ',mess,['']); nville=cleaner(**ville**);rang=finder(nville);

end *// Tant que la ville n'est pas référencée on recommence*

if Value(rang,3)==1 then **y**=doublon(rang) *// On utilise la fonction doublon si il s’agit d’un doublon*

else **y**=Value(rang,1);**y**(1,2)=Value(rang,2)

end

endfunction

function [**d**]=calculdistance(**C**, **i**, **j**) *//Calcule la distance entre les deux points en utilisant les coordonnées sphériques*

*//theta est la longitude*

*//phi est la latitude*

phi=**C**(**i**,1)\*%pi/180

theta=**C**(**i**,2)\*%pi/180

phi1=**C**(**j**,1)\*%pi/180

theta1=**C**(**j**,2)\*%pi/180

x=6378\*cos(phi)\*cos(theta)

y=6378\*cos(phi)\*sin(theta)

z=6378\*sin(phi)

X=6378\*cos(phi1)\*cos(theta1)

Y=6378\*cos(phi1)\*sin(theta1)

Z=6378\*sin(phi1)

**d**=sqrt((z-Z)^2+(x-X)^2+(y-Y)^2)

alpha=2\*asin(**d**/(2\*6378))

D=6378\*alpha

endfunction

function []=affichage(**y1**, **y2**, **ville1**, **ville2**, **d**)*//Affiche la carte de france avec le nom des villes et la distance qui les sépare. Fonction à utilité graphique mais sans intérêt algorithmique.*

RGB = ReadImage('C:\Users\thibault\Desktop\UTT\france1.jpg');

[image, ColorMap] = RGB2Ind(RGB);

FigureHandle = ShowImage(image, 'Example', ColorMap);

coordIm=transform([**y1**;**y2**])

if **d**>=100 then

xstring(coordIm(1,2),coordIm(1,1),**ville1**)

xstring(coordIm(2,2),coordIm(2,1),**ville2**)

xpoly([coordIm(1,2),coordIm(2,2)],[coordIm(1,1),coordIm(2,1)],"lines",1);

**d**=round(**d**);

**d**=string(**d**)+" Km";

xstring((coordIm(1,2)+coordIm(2,2))/2,(coordIm(1,1)+coordIm(2,1))/2,**d**);

else

yhaut=max(coordIm(1,1),coordIm(2,1));

if yhaut==coordIm(1,1) then xhaut=coordIm(1,2);yhaut=yhaut+5;ybas=coordIm(2,1)-5;xbas=coordIm(2,2); villehaut=**ville1**;villebas=**ville2**;

else yhaut=yhaut+5; xhaut=coordIm(2,2); ybas=coordIm(1,1)-5;xbas=coordIm(1,2); villehaut=**ville2**;villebas=**ville1**;

end

xstring(xhaut,yhaut,villehaut);

xstring(xbas,ybas,villebas);

xpoly([coordIm(1,2),coordIm(2,2)],[coordIm(1,1),coordIm(2,1)],"lines",1);

**d**=round(**d**);

**d**=string(**d**)+" Km";

xstring((xhaut+xbas)/2,(yhaut+ybas)/2,**d**);

uicontrol("String", "Close the window", "Position", [10 10 100, 25], "Callback", "delete(gcf())");

end

endfunction

function []=utt()

[y1,ville1]=bonneville(1,'ère')

[y2,ville2]=bonneville(2,'ème')

*//Calcul de la distance*

d=calculdistance([y1;y2],1,2)

*//Affichage du résultat final.*

affichage(y1,y2,ville1,ville2,d);

endfunction