

Confluences

Convention Manufacture

Tables rondes : contenus

Dossier principal

Citizen Cape

Code principal

```

/*
  _____
 /_ _ \  /_ _ \  /_ _ \  /_ _ \  /_ _ \
| ( ) | | ( ) | | ( ) | | | \_ _ \
 \_ _ /  \_ _ /  \_ _ /  | | |  \_ _ / v 0.7

CC-BY-SA Belgique 2.0 - VANLINDT MARC
*/

color      Confluences      = color(6,103,49);
color      Confluences2     = color(22,125,63);

color      noir              = color(0,0,0);
color      blanc             = color(255,255,255);

color      Haut              = Confluences;
color      Gauche            = color(0,0,255); //color(66,131,255);
color      Droite            = color(255,0,0);

PFont      PoliceConfluences;
PFont      PoliceConfluencesGras;
PFont      PoliceConfluencesItalique;

```

```
float      NombrePortes      = 10.0;

int Profil = 0;

char touches[][] =
{{'a', 'q'}, {'z', 's'}, {'e', 'd'}, {'r', 'f'}, {'t', 'g'}, {'y', 'h'}, {'u', 'j'}, {'i',
'k'}, {'o', 'l'}, {'p', 'm'}, {'=', 'ù'}, {'$', '*'}};

int      largeur      = 980;
int      hauteur      = 900;

int      decalx = 50; // OBSOLETE - A virer du code!
int      decaly = 50; //

int[][]  JimMorrison =
{{1}, {1,2}, {1,2,3}, {1,2,3,4}, {1,2,3,4,5}, {1,2,3,4,5,6}, {1,2,3,4,5,6,7}, {1,2,
3,4,5,6,7,8}, {1,2,3,4,5,6,7,8,9}, {1,2,3,4,5,6,7,8,9,10}, {1,2,3,4,5,6,7,8,9,1
0,11}, {1,2,3,4,5,6,7,8,9,10,11,12}, {1,2,3,4,5,6,7,8,9,10,11,12,13}, {1,2,3,4,
5,6,7,8,9,10,11,12,13,14}, {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15}, {1,2,3,4,5,6
,7,8,9,10,11,12,13,14,15,16}, {1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}};
int[]    RayManzarek = {99,99,99,99,99,99,99,99,99,99,99,99};

String[][] reponses = {
    {"Non", "Oui"},
    {"Revenir dans le passé", "Décider de l'avenir"},
    {"Faire faire", "Faire"},
    {"Pragmatique assis", "Rêveur debout"},
    {"Seul on va plus vite", "Ensemble, on va plus loin"},
    {"Vie hyperactive", "Vie associative"},
    {"Bulletin de vote", "Assemblée citoyenne"},
    {"Propriété privée", "Bien commun"},
    {"Economie financière", "Economie sociale"},
    {"Heindiiiiividu", "Kolegteeffe"}
};

PImage fond;

Table exportcsv;

boolean SauveOK = false;

void setup(){

    remiseazero();
    size(1080,1920,P3D); // Taille de
l'écran. P3D important // En plein
    fullScreen(); // En plein
écran. Mettre ça en commentaire pour test en fenêtré // Fond général
    background(Confluences); // Fond général
de couleur noire // important :
    noStroke();
```

```
empêche les bords aux objets
  frameRate(60); // Pas important
  smooth(4); // adoucit les
polices
  loop(); // Très
important!
  fond = loadImage("./image.png"); // Fond général
de l'écran
  PoliceConfluences = createFont("swiss2.ttf",128); // police de
caractère confluence (swiss1 = gras, swiss3 = italique)
  PoliceConfluencesGras = createFont("swiss1.ttf",128); // police de
caractère confluence (swiss1 = gras, swiss3 = italique)
  PoliceConfluencesItalique = createFont("swiss3.ttf",128); // police de
caractère confluence (swiss1 = gras, swiss3 = italique)
  exportcsv = loadTable("data/TheDoors.csv","header"); // Fichier
d'exportation des tables. Ce fichier doit exister. Il n'y a pas de
vérification de sa présence.
}

void draw(){
  image(fond,0,0);
  creationcouleurs();
  affichageportes();
  appuitouches();
  afficheevolution();

  fill(255,255,255,64);
  rect(50,1230, 980,1920-1230-50);
  textFont(PoliceConfluencesGras);
  textSize(25);
  fill(0,0,0);
  text("Statistiques :",60,1255);

  int NombreParticipants=exportcsv.getRowCount();
  textFont(PoliceConfluences);
  textSize(20);
  fill(0,0,0);
  text("Nombre de participants : "+NombreParticipants,60,1280);
  int[] statsportes = {0,0,0,0,0,0,0,0,0,0,0,0};

  for(int i = 0; i <= NombrePortes; i++){
    for (TableRow row : exportcsv.findRows(str(i+1), "final"))
    {statsportes[i]+=1;}
    textAlign(CENTER);

    textFont(PoliceConfluencesGras);
    textSize(15);
    text("Profil "+(i+1), 100 + (1080-200)/10*i,1860);

    textAlign(LEFT);
  }
}
```

```
int maxstats = max(statsportes);
int[][] statscoord = {{0,0},{0,0}};
int hauteurstats=475;

for(int i = 0; i <= NombrePortes; i++){
circle(100 + (1080-200)/10*i, 1800 - hauteurstats/maxstats*statsportes[i],
10);
if(i<NombrePortes){
stroke(2);
noFill();

bezier(

float (100 + (1080-200)/10*(i+1)),
float (1800 - hauteurstats/maxstats*statsportes[i+1]),
float (100 + (1080-200)/10*(i)),
float (1800 - hauteurstats/maxstats*statsportes[i+1]),

float (100 + (1080-200)/10*(i+1)),
float (1800 - hauteurstats/maxstats*statsportes[i]),
float (100 + (1080-200)/10*(i)),
float (1800 - hauteurstats/maxstats*statsportes[i]));

/*
line(
100 + (1080-200)/10*i,
1800 - hauteurstats/maxstats*statsportes[i],
100 + (1080-200)/10*(i+1),
1800 - hauteurstats/maxstats*statsportes[(i+1)]);

line(
float (100 + (1080-200)/10*(i+1)),
float (1800 - hauteurstats/maxstats*statsportes[i+1]),
float (100 + (1080-200)/10*(i)),
float (1800 - hauteurstats/maxstats*statsportes[i+1])

);*/

}
textAlign(CENTER);
textFont(PoliceConfluencesItalique);
textSize(15);
text(statsportes[i], 100 + (1080-200)/10*i,1800 -
hauteurstats/maxstats*statsportes[i]-10);
textAlign(LEFT);
fill(0,0,0,255);
}
```

```
noStroke();

int Somme = 0;
for(int i=0;i<=NombrePortes;i++){
  if(RayManzarek[i]!=99){
    Somme += (int(RayManzarek[i])*int(pow(2,NombrePortes-i-1)));
  }
}
Profil = 0;
for (int z = 0; z < NombrePortes; z++){
  Profil +=RayManzarek[z];
}
if(Profil<=NombrePortes){
text(Profil+1,largeur+200, (40+hauteur/NombrePortes));
  if(SauveOK == false){
    TableRow newRow = exportcsv.addRow();
    newRow.setInt("id", exportcsv.getRowCount() - 1);
    newRow.setInt("chemin", Somme+1);
    newRow.setInt("final", Profil+1);
    println(Somme);
    saveTable(exportcsv, "data/TheDoors.csv");
    SauveOK=true;
  }
  int NombreProfil=0;
  for (TableRow row : exportcsv.findRows(str(Profil+1), "final")) {
    NombreProfil+=1;
  }
  int NombreChemin=0;
  for (TableRow row : exportcsv.findRows(str(Somme+1), "chemin")) {
    NombreChemin+=1;
  }
  int base=hauteur+100;
  int hauteur2=180;
  fill(Confluences2);
  rect(50,base,980,hauteur2);
  fill(Confluences);
  rect(51,base+1,978,hauteur2-2);
  fill(Confluences2);
  rect(52,base+2,976,hauteur2-4);
  textFont(PoliceConfluencesGras);
  fill(blanc);
  textSize(40);
  text("Profil n°"+(Profil+1)+" atteint par le chemin n°"+(Somme+1),
65,base+50);
```

```
    textFont(PoliceConfluences);
    textSize(30);

    textFont(PoliceConfluencesItalique);
    textSize(20);
    String NPText = "";
    if(NombreProfil==1){NPText="Vous êtes le premier à être arrivé à ce
profil et "};else{NPText="Vous êtes "+NombreProfil+" à être arrivés à ce
profil et "};
    String NCText = "";
    if(NombreChemin==1){NCText="le premier à avoir emprunté ce
chemin."};else{NCText=NombreChemin+" à avoir empruntés ce chemin."};

    text(NPText+ NCText,65,base+150);
    textFont(PoliceConfluences);
    textSize(30);
//    text(Profils[Profil], 1010,195);
    if(Profil==0){
    text("Vous n'avez atteint aucun cap sur les 10.", 65,base+110);
    }else{
        String tmp = "cap";
        if(Profil == 0 || Profil ==1){tmp="cap";}else{tmp="caps";}
        text("Vous avez atteint "+Profil+" "+tmp+" sur les 10.",
65,base+110);}
    }
}

void creationcouleurs(){
    for (int x=0; x<=NombrePortes; x+=1){
        color c=lerpColor(Haut, Gauche, x*1.0/NombrePortes); color
d=lerpColor(Haut, Droite, x*1.0/NombrePortes);
        JimMorrison[x][0]=c; JimMorrison[x][x]=d;
    }
    for (int y=1;y<=NombrePortes;y+=1){
        for (int x=0;x<=y;x+=1){
            color c=lerpColor(JimMorrison[y][0], JimMorrison[y][y], x*1.0/(y));
            JimMorrison[y][x]=c;
        }
    }
    JimMorrison[0][0]=Haut;
}

void affichageportes(){
    for (int y=0;y<=NombrePortes;y+=1){
        for (int x=0;x<=y;x+=1){
            int SommeRM = 0;
            for (int z = 0; z < y; z++) {
                SommeRM +=RayManzarek[z];
            }
            fill(JimMorrison[y][x]);
            rect(decalx +(x*largeur/NombrePortes)*(NombrePortes/(y+1)),
```

```

decaly+hauteur/(NombrePortes+1)*(y),
largeur/NombrePortes*(NombrePortes/(y+1)),hauteur/(NombrePortes+1));
    if(x!=SommeRM){
        fill(0,0,0,128);
        rect(decalx +(x*largeur/NombrePortes)*(NombrePortes/(y+1)),
decaly+hauteur/(NombrePortes+1)*(y),
largeur/NombrePortes*(NombrePortes/(y+1)),hauteur/(NombrePortes+1));
    }
}
}
}

void remiseazero(){
    for(int i=0;i<=NombrePortes;i++){
        RayManzarek[i] = 99;
    }
    SauveOK = false;
}

void appuitouches(){

    if (keyPressed) {
        for (int k=0;k<=NombrePortes;k++){
            if (key == touches[k][0]) {RayManzarek[k]=0;}
            if (key == touches[k][1]) {RayManzarek[k]=1;}
        }
        if(key=='w'){
            remiseazero();
        }
    }
}

void afficheevolution(){
    for (int k=0;k<=9;k++){
        fill(blanc);
        textFont(PoliceConfluences);
        textSize(20);
        fill(255,255,255,96);
        String tmp = "";
        if(k==0){tmp="Salut, ça va ? ";} else {tmp=""};
        if(RayManzarek[k]==99) {fill(255,255,255,96);text(tmp+reponses[k][0]+" /
"+reponses[k][1],decalx +20, 30+decaly +
((hauteur/(NombrePortes+1))*1+(hauteur/(NombrePortes+1))*(k)));}

        if(RayManzarek[k]==0) {fill(255,128,128,96);text(tmp+reponses[k][0]+" /
"+reponses[k][1],decalx +20, 30+decaly +
((hauteur/(NombrePortes+1))*1+(hauteur/(NombrePortes+1))*(k)));
            fill(128,128,255);
            text(tmp+reponses[k][0],decalx +20, 30+decaly +
((hauteur/(NombrePortes+1))*1+(hauteur/(NombrePortes+1))*(k)));
        }
    }
}

```

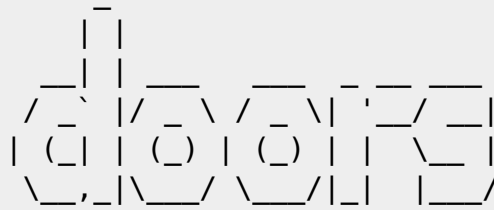
```

    if(RayManzarek[k]==1) {fill(128,128,255,96); text(tmp+reponses[k][1]+" /
"+reponses[k][0],decalx +20, 30+decaly +
((hauteur/(NombrePortes+1))*1+(hauteur/(NombrePortes+1))*(k)));
        fill(255,128,128);
        text(tmp+reponses[k][1],decalx +20, 30+decaly +
((hauteur/(NombrePortes+1))*1+(hauteur/(NombrePortes+1))*(k)));
    }
}
}
}

```

Code secondaire

```

/*

(ENTRY) \v 0.1

```

CC-BY-SA Belgique 2.0 - VANLINDT MARC

```

*/
color          Confluences      = color(6,103,49);
color          Confluences2     = color(22,125,63);
color          couleurnom       = color(Confluences);
color          couleurmail      = color(Confluences);
color          couleurcodepostal = color(Confluences);

PFont          PoliceConfluences;
PFont          PoliceConfluencesGras;
PFont          PoliceConfluencesItalique;
PImage         fond;
PImage         fond2;
Table          exportcsv;
boolean        SauveOK = false;
boolean        SauveOK2 = false;
int            delai=300;
char[][][][]   touches1 =
{
  {
    {'1','1'},{'2','2'},{'3','3'},{'4','4'},{'5','5'},{'6','6'},{'7','7'},{'8','8'},{'9','9'},{'0','0'}
  },
  {
    {'a','A'},{'z','Z'},{'e','E'},{'r','R'},{'t','T'},{'y','Y'},{'u','U'},{'i','I'},{'o','O'},{'p','P'}
  },
  {
    {'q','Q'},{'s','S'},{'d','D'},{'f','F'},{'g','G'},{'h','H'},{'j','J'},{'k','K'},

```



```

K'},{'l','L'},{'m','M'}
    },
    {
{'w','W'},{'x','X'},{'c','C'},{'v','V'},{'b','B'},{'n','N'},{'.',',''},{'-
','_'},{' ':' '='},{'@','@'}
    }
};

String Nom1 = "";
String Mail = "";
String CodePostal = "";
String nom="nom";

int Shift = 0;
int Shift2 = 1;
int compteur=0;

void setup(){
    size(1920,1080,P3D); // Taille
    &de l'écran. P3D important // En plein
    fullScreen(); // Fond
    écran. Mettre ça en commentaire pour test en fenêtré // important
    background(Confluences); // Pas
    général de couleur noire // adoucité
    noStroke(); // Très
    : empêche les bords aux objets // Fond
    frameRate(60); // Fond
    important // Fond
    smooth(4); // Fond
    les polices // Fond
    loop(); // Fond
    important! // Fond
    fond = loadImage("./imageentry.png"); // Fond
    général de l'écran // Fond
    fond2 = loadImage("./imageentry2.png"); // Fond
    général de l'écran // Fond
    PoliceConfluences = createFont("swiss2.ttf",128); // police de
    caractère confluence (swiss1 = gras, swiss3 = italique) // police de
    PoliceConfluencesGras = createFont("swiss1.ttf",128); // police de
    caractère confluence (swiss1 = gras, swiss3 = italique) // police de
    PoliceConfluencesItalique = createFont("swiss3.ttf",128); // police de
    caractère confluence (swiss1 = gras, swiss3 = italique) // Fichier
    exportcsv = loadTable("data/TheDoorsEntree.csv","header"); // Fichier
    d'exportation des tables. Ce fichier doit exister. Il n'y a pas de
    vérification de sa présence.
}

void draw(){
    image(fond,0,0);
    if(SauveOK==false){

```

```
    PasOk();
} else {
    ok();
}
}
void PasOk()
{
    if(nom=="nom")
    {
        couleurnom=color(Confluences);
        couleurmail=color(255,255,255,32);
        couleurcodepostal=color(255,255,255,32);
    }
    if(nom=="mail")
    {
        couleurmail=color(Confluences);
        couleurnom=color(255,255,255,32);
        couleurcodepostal=color(255,255,255,32);
    }
    if(nom=="codepostal")
    {
        couleurcodepostal=color(Confluences);
        couleurnom=color(255,255,255,32);
        couleurmail=color(255,255,255,32);
    }
    // zone nom
    fill (255,255,255,64);
    rect (50,50,600,60,10);
    fill(couleurnom);
    textAlign(LEFT);
    textSize(35);
    text("Prénom : "+Nom1,50+15,50+35+12);
    //zone remise à zéro
    fill (255,255,255,64);
    rect (660,50,220,60,10);

    text("Remise à zéro",660+15,50+35+12);

//Zone Mail
    fill (255,255,255,64);

    rect (50,120,600,60,10);
    fill (couleurmail);

    textAlign(LEFT);
    textSize(35);
    text("Mail : "+Mail,50+15,50+35+12+70);

//Zone code postal
    fill (255,255,255,64);
```

```
rect (50,190,600,60,10);
fill (couleurcodepostal);

textAlign(LEFT);
textSize(35);
text("Code Postal : "+CodePostal,50+15,50+35+12+140);

//zone "OK"
fill (255,255,255,64);
rect (50,260,600,60,10);

fill (Confluences);
textAlign(CENTER);
textSize(35);
text("OK",325,260+35+12);

if (
  mousePressed == true &&
  mouseX>660 &&
  mouseX<880 &&
  mouseY>50 &&
  mouseY<110)
{
  remiseazero();
  println("toto");
  delay(delai);
}

if (
  mousePressed == true &&
  mouseX>50 &&
  mouseX<650 &&
  mouseY>50 &&
  mouseY<110)
{
  nom="nom";
  delay(delai);
}

if (
  mousePressed == true &&
  mouseX>50 &&
  mouseX<650 &&
  mouseY>120 &&
  mouseY<180)
{
  nom="mail";
  delay(delai);
}

if (
```

```
mousePressed == true &&
mouseX>50 &&
mouseX<650 &&
mouseY>170 &&
mouseY<230)
{
  nom="codepostal";
  delay(delai);
}

if (
  mousePressed == true &&
  mouseX>50 &&
  mouseX<650 &&
  mouseY>240 &&
  mouseY<300)
{
  SauveOK=true;
  delay(delai);
}

// affichages des touches
for(int i=0;i<=3;i++){
  for(int j=0;j<=9;j++){
    stroke(Confluences2);
    fill(255,255,255,64);
    rect(
      50+(j*((width-100)/10)),
      (height/2)+(i*height/11),
      (width-100)/11,
      height/12,
      10);
    textAlign(CENTER);
    fill(Confluences);
    textFont(PoliceConfluencesGras);
    textSize(50);
    text(
      touches1[i][j][Shift],
      50+(j*((width-100)/10))+((width-100)/10)/3,
      ((height/2)+(i*height/11))+height/11-35
    );
    textFont(PoliceConfluencesItalique);
    fill(Confluences2);
    textSize(30);
    text(
      touches1[i][j][Shift]+" "+touches1[i][j][Shift2],
      50+(j*((width-100)/10))+((width-100)/10)/3*2+10,
      ((height/2)+(i*height/11))+height/11-20
    );
    if (
      mousePressed == true &&
```

```
mouseX>(50+(j*((width-100)/10))) &&
mouseX<(50+((j+1)*((width-100)/10))) &&
mouseY>((height/2)+(i*height/11)) &&
mouseY<((height/2)+((i+1)*height/11))
{
  if(nom=="nom"){
    Nom1=Nom1+=touches1[i][j][Shift];
    delay(delai);
  }
  if(nom=="mail"){
    Mail=Mail+=touches1[i][j][Shift];
    delay(delai);
  }
  if(nom=="codepostal"){
    CodePostal=CodePostal+=touches1[i][j][Shift];
    delay(delai);
  }
}
}
}
//touche shift
fill(255,255,255,64);
rect(50,
  (height/2)+(4*height/11),
  (width-100)/11,
  height/12,
  10);
fill(Confluences);
text("Maj",
  50+((width-100)/11)/2,
  (height/2)+(4.6*height/11)
);

if(
mousePressed == true &&
mouseX>50 &&
mouseX<50+((width-100)/11) &&
mouseY>(height/2)+(4*height/11) &&
mouseY<(height/2)+(5*height/11)){
  if(Shift==0){
    Shift=1;
    Shift2=0;
  }
  else{
    Shift=0;
    Shift2=1;
  }
  delay(delai);
}
}
void remiseazero(){
```

```
Nom1 = "";
Mail = "";
nom="nom";
CodePostal = "";
}

void ok()
{
    background(fond2);

    if(SauveOK2 == false){
        TableRow newRow = exportcsv.addRow();
        newRow.setInt("id", exportcsv.getRowCount() -1);
        newRow.setString("nom", Nom1);
        newRow.setString("mail", Mail);
        newRow.setString("codepostal", CodePostal);
        saveTable(exportcsv, "data/TheDoorsEntree.csv");
        println(Nom1+" "+Mail+" "+CodePostal);
        SauveOK2 = true;
    }
}

// int NombreParticipants=exportcsv.getRowCount();
```

Croisière

Préparation

Conduite

Détails

Débriefing

Site internet

From:
<https://wiki.11h22.be/> -

Permanent link:
<https://wiki.11h22.be/doku.php?id=productions:confluences>

Last update: **2022/03/13 23:34**

