

PROJET ISN

DIGICODE

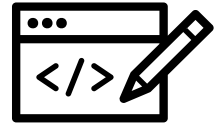
DYLAN LAGRANGE - TS2

PABLOJEANDYLAN.FREE.FR

PRÉSENTATION.

BUT DU PROJET.

Reproduire virtuellement un digicode.
Créer entièrement un programme Java.

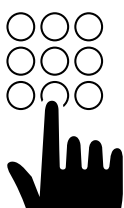
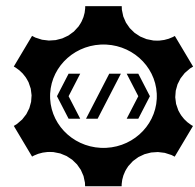


UN DIGICODE.

Nous avons choisis de faire un digicode pour notre projet ISN. Nous avons donc créé une application reproduisant un digicode, avec des boutons de 1 à 9, un écran pour afficher les chiffres, deux boutons Valider et Restart.

LES OUTILS UTILISÉS.

Nous avons mis en oeuvre nos connaissances en codage acquise lors de nos cours d'ISN pour créer ce code, ainsi que le logiciel de codage Processing.



UNE CRÉATION EN TROIS ÉTAPES.

Nous sommes partis de zéro, nous avons donc amélioré un code tout simple au départ, pour arriver à notre digicode complet. On peut distinguer trois grandes étapes dans ce développement.

PREMIÈRE ÉTAPE.



DEUX_CARRÉ.PDE

UN BOUTON.

Le premier programme que nous avons conçu comporte:

- > une interface
- > une fonction booléenne « clic » (fausse au départ).
- > une détection du clic de la souris inverse la valeur de la fonction « clic » qui fait apparaître ou disparaître la croix sur le bouton.
- > une interface composée d'un carré.

Quand la fonction devient vraie, une croix apparaît.

LA FONCTION BOOLÉENNE.

Fonction qui ne peut avoir pour valeur que « false » ou « true ». Ici, elle permet de sélectionner ou désélectionner le carré. Exemple: `boolean clic = false`

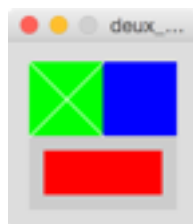
LA FONCTION STRING.

Fonction dont les objets ne peuvent pas changer de valeur. Exemple: `String motdepasse = "1997"`

LA DÉTECTION DE LA SOURIS.

Fonction qui ne peut avoir pour valeur que « false » ou « true ». Ici, elle permet de sélectionner ou désélectionner le carré.

```
void mousePressed(){  
    if ((mouseX>0) && (mouseX<50) &&  
        (mouseY>0) && (mouseY<50)){  
        if(clic==false){clic=true;}  
        else{clic=false;} } }
```



DEUX_CARRÉ_CROIX.PDE

DEUX BOUTONS.

Nous avons ajouté:

- > une interface composée de deux carrés (les deux boutons).
- > un bouton valider.

La fonction « clic » fausse fait apparaître une croix sur le bouton `bleu`. Inversement, si la fonction est vraie, la croix est sur le bouton `vert`.

SECONDE ÉTAPE.

TROIS FONCTIONS BOOLÉENNES.



Ce programme comporte:

- > Trois fonction booléenne (carré1, carré2 et carré3) représentant chacune un carré.
- > Un bouton « Valider »
- > Un message disant si le code est bon ou mauvais.

Il y a 7 possibilités de mot de passe.

Lors du clic sur le bouton « Valider », le programme compare la valeur des fonction booléenne à la valeur qui débloquent le digicode.

Ici, il faut que carré1=faux, carré2=vrai et carré3=faux.

LA FONCTION NOLOOP.

Lors de la validation, le programme réécrivait sur la réponse de la validation. Nous avons donc intégré la fonction « noLoop() » pour arrêter le programme lorsque l'on clique sur le bouton « Valider ».



AJOUT D'UN ÉCRAN ET BOUTON À CHIFFRE.

Nous avons modifier le programme précédent pour avoir:

- > Trois boutons représentant le clavier d'un digicode.
- > Un écran affichant le chiffre sélectionné.

Il y a 3 possibilités de mot de passe.

La fonction booléenne associé à chaque chiffre ne nous permet pas d'avoir un mot de passe avec plusieurs chiffres. Nous avons donc dû résoudre ce problème.

CRÉATION BOUCLE ET FONCTION CODE.

Ce programme comporte:

- > Une boucle pour que chaque chiffre du mot de passe.
- > Une fonction « code » à laquelle ce rajoutera le chiffre sélectionné.
- > Une valeur « motdepasse » pour déterminer le code qui débloquent le digicode.



Cette boucle a 2 étapes, elle permet de sélectionner une première fois un chiffre, et une deuxième fois pour le second chiffre.

Lors du clic sur le bouton « Valider », le programme compare la valeur de « code » à celle de « motdepasse ». Si c'est deux valeur sont égales, le digicode devient vert, sinon il devient rouge.

À LAQUELLE CE RAJOUTERA LE CHIFFRE SÉLECTIONNÉ?

Lorsqu'on clique sur une touche du clavier, le programme va rajouter le caractère du chiffre souhaité à la fonction « code ». Il ne fait pas une addition, sinon nous aurions $1+9+9+7=26$ au lieu de $"1"+"9"+"9"+"7"="1997"$.

COMPARER DES VALEURS.

Les fonctions « motdepasse » et « code » sont des **String**, cela signifie que ce sont des séquences de caractères et non des valeurs numériques.

Nous avons donc utilisé pour le bouton Valider « `if (motdepasse.equals(code))` ».

DERNIÈRE ÉTAPE.



DIGICODE.PDE

LE DIGICODE.

Nous avons modifier le programme précédent pour avoir:

- > un clavier à neuf chiffres.
- > un code à 4 chiffres.
- > un bouton « RESTART » permettant de réinitialiser le Digicode.
- > un graphisme amélioré grâce à la fonction stroke() modifiée pour avoir des coins arrondis.
- > l'affichage d'image pour illustrer la validation du code.
- > un nombre maximal de tentative, le Digicode est bloqué au bout de 4 essais.
- > la redirection vers notre site internet lors de la saisis du bon mot de passe.

LE BOUTON RESTART.

Il permet de relancé le **void setup()**, cela va remettre a zéro les valeurs modifiées quand l'utilisateur a cliqué sur le clavier du Digicode.

```
void mouseClicked(){  
  if ((mouseX>370) && (mouseX<470) &&  
      (mouseY>630) && (mouseY<670)) {  
    setup();  
  }  
}
```

CARRÉ AVEC COINS ARRONDIS.

Pour améliorer la graphisme du Digicode, nous avons arrondi les bords des touches du clavier. Pour cela, nous avons rajouté des bordures avec bord arrondi de la couleur de la touche.

```
fill(32, 148, 250);  
stroke(32, 148, 250);  
strokeWeight(10);  
strokeJoin(ROUND);  
rect(50, 150, 100, 100);
```

INTÉGRATION DE L'IMAGE.

Pour compléter notre digicode, nous avons fais apparaitre une image lors de la validation. chaque image illustre si le code est vrai, faux, ou si le nombre de tentatives maximum est atteint.

```
PImage jean;  
void setup() {  
  jean = loadImage("jean.jpg");  
  void draw() {  
    image(jean,0,-20,500,620);  
  }  
}
```

NOMBRE DE TENTATIVE.

Nous avons intégrer un « mouseClicked » qui permet de compter combien de fois l'utilisateur clique sur le bouton Valider. Si il est supérieur à 3 et que le code est faux, lors de la validation une image apparait et le digicode ce bloque.

```
void mouseClicked(){  
  if ((mouseX>370) && (mouseX<470) &&  
      (mouseY>630) && (mouseY<670)) {  
    restartnb=restartnb+1;  
    validation = false;  
    setup(); } }  
}
```

REDIRECTION VERS UN SITE INTERNET.

Lors de la validation, le digicode ouvre automatiquement la page de notre site internet.

```
link("http://pablojeandylan.free.fr");
```

BILAN.

Ce Digicode est une création de groupe, Jean Pablo et moi avons créé ce Digicode. Nous avons pensé, ensemble, comment résoudre chacun des problèmes que nous rencontrions.



Nous sommes partis d'un simple bouton, qui après amélioration, à former un Digicode.

À la fin, nous avons pu améliorer le graphisme de notre digicode pour que celui ci soit plus original et personnel. Nous avons rajouté des fonctions qui n'était pas utile mais qui faisait partis de nos connaissances (telle que l'intégration des images, le lien vers notre site internet..).



Ce programme nous a fait acquérir de nouvelles connaissance dans le langage JAVA, et dans l'intégration d'un programme dans un site internet en HTML.

ANNEXE. CODE COMPLET DU DIGICODE.

Code de l'élève	Code du professeur	Code de l'élève	Code du professeur
int curseur=0;	textSize(40);		textSize(40);
String code="";	fill(0);	if ((mouseX>50) &&	code=code+"6";
String motdepasse="1997";	text("3", 390, 210);	(mouseX<150) &&	}
boolean validation = false;	fill(0);	(mouseY>150) &&	
int restartnb=0;	text("4", 90, 360);	(mouseY<250)) {	if ((mouseX>50) &&
Plmage jean;	textSize(40);		(mouseX<150) &&
Plmage content;	fill(0);	textSize(40);	(mouseY>450) &&
Plmage tentatives;	text("5", 240, 360);	fill(0, 0, 0);	(mouseY<550)) {
	textSize(40);	text("1", 180, 100);	textSize(40);
void setup() {	fill(0);	textSize(40);	fill(0, 0, 0);
size(500, 700);	text("6", 390, 360);	code=code+"1";	text("7", 180, 100);
code="";	fill(0);	}	textSize(40);
curseur=0;	text("7", 90, 510);		code=code+"7";
noStroke();	textSize(40);	if ((mouseX>200) &&	}
fill(242,244,255);	fill(0);	(mouseX<300) &&	
rect(0,0,500,700);	text("8", 240, 510);	(mouseY>150) &&	if ((mouseX>200) &&
jean =	textSize(40);	(mouseY<250)) {	(mouseX<300) &&
loadImage("jean.jpg");	fill(0);	textSize(40);	(mouseY>450) &&
content =	text("9", 390, 510);	fill(0, 0, 0);	(mouseY<550)) {
loadImage("content.jpg");		text("2", 180, 100);	textSize(40);
tentatives =	textSize(40);	textSize(40);	fill(0, 0, 0);
loadImage("tentatives.jpg");	fill(0);	code=code+"2";	text("8", 180, 100);
}	text("_", 183, 104);	}	textSize(40);
			code=code+"8";
void draw() {	textSize(40);	if ((mouseX>350) &&	}
noStroke();	fill(0);	(mouseX<450) &&	
fill(0, 0, 0);	text("_", 220, 104);	(mouseY>150) &&	if ((mouseX>350) &&
rect(0, 0, 500, 50);		(mouseY<250)) {	(mouseX<450) &&
	textSize(40);	textSize(40);	(mouseY>450) &&
if (validation==false){	fill(0);	fill(0, 0, 0);	(mouseY<550)) {
fill(32, 148, 250);	text("_", 260, 104);	text("3", 180, 100);	textSize(40);
stroke(32, 148, 250);		textSize(40);	fill(0, 0, 0);
strokeWeight(10);	textSize(40);	code=code+"3";	text("9", 180, 100);
strokeJoin(ROUND);	fill(0);	}	textSize(40);
rect(50, 150, 100, 100);	text("_", 300, 104);		code=code+"9";
fill(32, 148, 250);	}	if ((mouseX>50) &&	}
rect(200, 150, 100, 100);	fill(255,59,48);	(mouseX<150) &&	}
fill(32, 148, 250);	stroke(255,59,48);	(mouseY>300) &&	
rect(350, 150, 100, 100);	strokeWeight(10);	(mouseY<400)) {	if (curseur==1) {
	strokeJoin(ROUND);	textSize(40);	if ((mouseX>50) &&
fill(32, 148, 250);	rect(30, 630, 300, 40);	fill(0, 0, 0);	(mouseX<150) &&
rect(50, 300, 100, 100);		text("4", 180, 100);	(mouseY>150) &&
fill(32, 148, 250);	fill(0,245,234);	textSize(40);	(mouseY<250)) {
rect(200, 300, 100, 100);	stroke(0,245,234);	code=code+"4";	textSize(40);
fill(32, 148, 250);	strokeWeight(10);	}	fill(0, 0, 0);
rect(350, 300, 100, 100);	strokeJoin(ROUND);		text("1", 220, 100);
	rect(370, 630, 100, 40);	if ((mouseX>200) &&	textSize(40);
fill(32, 148, 250);		(mouseX<300) &&	code=code+"1";
rect(50, 450, 100, 100);	textSize(40);	(mouseY>300) &&	}
fill(32, 148, 250);	fill(255, 255, 255);	(mouseY<400)) {	
rect(200, 450, 100, 100);	text("Saisir le code", 120,	textSize(40);	if ((mouseX>200) &&
fill(32, 148, 250);	40);	fill(0, 0, 0);	(mouseX<300) &&
rect(350, 450, 100, 100);	textSize(30);	text("5", 180, 100);	(mouseY>150) &&
	fill(0);	textSize(40);	(mouseY<250)) {
noStroke();	text("VALIDER", 125, 660);	code=code+"5";	textSize(40);
fill(0);	textSize(20);	}	fill(0, 0, 0);
rect(0, 600, 500, 100);	fill(0);	if ((mouseX>350) &&	text("2", 220, 100);
	text("RESTART", 377, 657);	(mouseX<450) &&	textSize(40);
textSize(40);	}	(mouseY>300) &&	code=code+"2";
fill(0);		(mouseY<400)) {	}
text("1", 90, 210);	void mousePressed() {	textSize(40);	if ((mouseX>350) &&
textSize(40);		fill(0, 0, 0);	(mouseX<450) &&
fill(0);	if (curseur==0) {	text("6", 180, 100);	
text("2", 240, 210);			if ((mouseX>350) &&

[illegible]


```

    }

    if ((mouseX>350) &&
(mouseX<450) &&
    (mouseY>450) &&
(mouseY<550)) {
    textSize(40);
    fill(0, 0, 0);
    text("9", 300, 100);
    textSize(40);
    code=code+"9";
    }
}

    if ((mouseX>30) &&
(mouseX<330) &&
    (mouseY>630) &&
(mouseY<670)) {
    validation = true;

if (motdepasse.equals(code))
{
    image(content,
0,50,500,570);
    link("http://
pablojeandylan.free.fr");

}
else{
    if(restartnb<3){
    image(jean,0,-20,500,620);}
    else{
    image(tentatives,
0,0,500,700);
    noLoop();
    }

}

    }

}

void mouseClicked(){
    if ((mouseX>370) &&
(mouseX<470) &&
    (mouseY>630) &&
(mouseY<670)) {
    restartnb=restartnb+1;
    validation = false;
    setup();

    }
}

void mouseReleased() {
    if ((mouseX>50) &&
(mouseX<450) &&
    (mouseY>150) &&
(mouseY<550)) {
    curseur=curseur+1;
    }
}

```