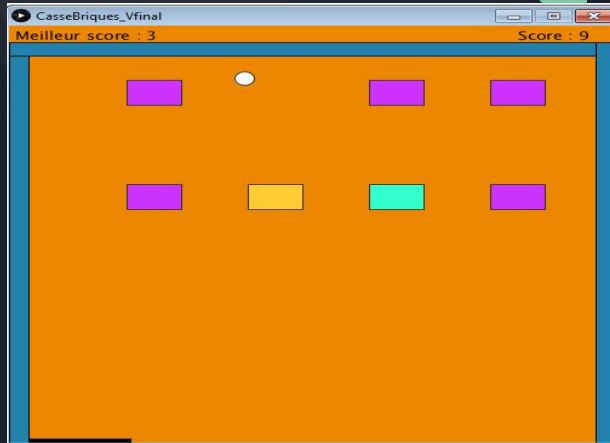


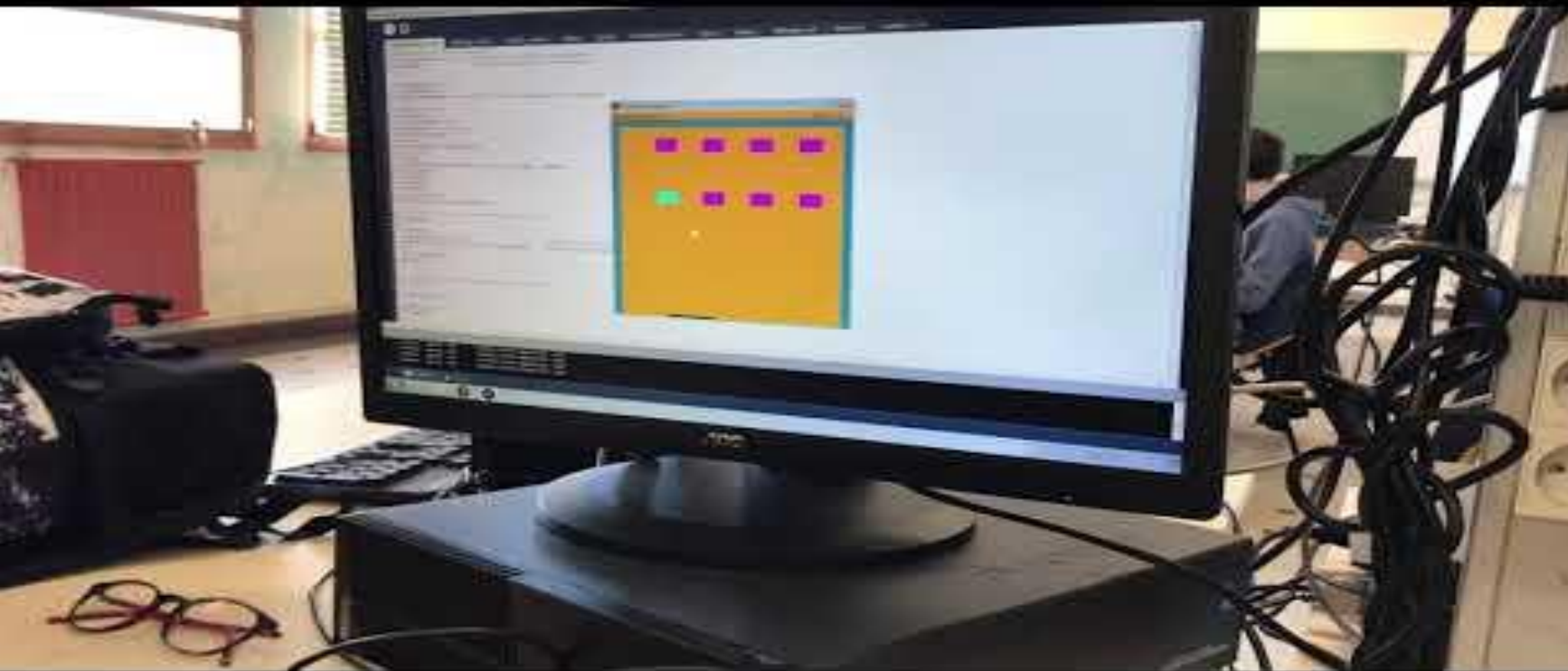
Projet ISN



Le casse brique



Nicolas ROBLET
Arnaud SADION



Les logiciels utilisés



Nom:

-Arduino

-Processing

Langage:

-C/C++

-Java

Fonction:

-Traitement des
données du joystick.

-Support du jeu

Envoie de ces valeurs
dans le moniteur série


Répartitions des tâches au sein du groupe

Tâches	Élève(s) l'ayant réalisé
Recherche de l'idée et élaboration des objectifs	Arnaud et Nicolas
Design du jeu	Arnaud
Score	Arnaud
Menu pause et victoire	Arnaud
Communication arduino processing	Nicolas
Exploitation des valeurs du joystick	Nicolas
Correction des bugs	Arnaud et Nicolas

Cahier des charges du jeu

Besoin à satisfaire et contraintes à respecter	Fonction créer	Solution
Facilité d'utilisation	Utilisation d'un jeu populaire et simple	Casse-brique
	Utilisation d'un module populaire pour le contrôle	Joystick
	Description des tâches à effectuer afin de débiter une partie	Texte insérer dans le jeu
Rendre le jeu agréable	Limitation des "bugs"	Programme
	fluidité du jeu	/
	Création d'une interface de pause	appuyer sur n'importe quelle touche du clavier

Tableau des variables

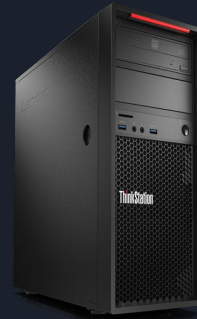
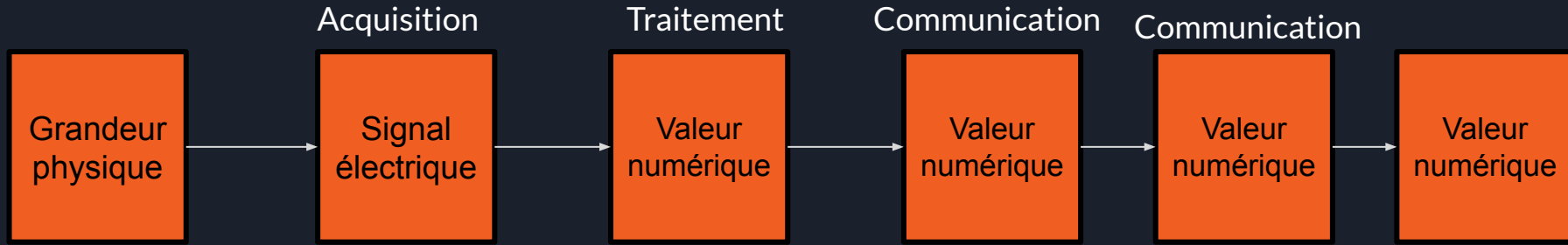
Type de variable	Variable	Fonction
Int (integer/entier)	Score	Compte le nombre de touche entre la balle et les briques
int	Pscore	Garde en mémoire le score de la partie précédente
int	MScore	Garde en mémoire le meilleur score de toutes les parties (réinitialisé à chaque lancement du jeu)
boolean	Marche	Si le jeu est en pause ou non
int	Position_barre	valeur de la position du centre de la barre sur l'axe X
int	hauteur_bandeau	20 pixels de large 
Int	ValeursManettes	valeur analogiques du joystick

Principe du fonctionnement de notre programme

```
CasseBriques_Vfinal  Affichage_Score  Arduino_commu  Brique  Defaite  Mouvement_Barre  Pause  Setup2  affichage_vict  demarrer  victoire ▼

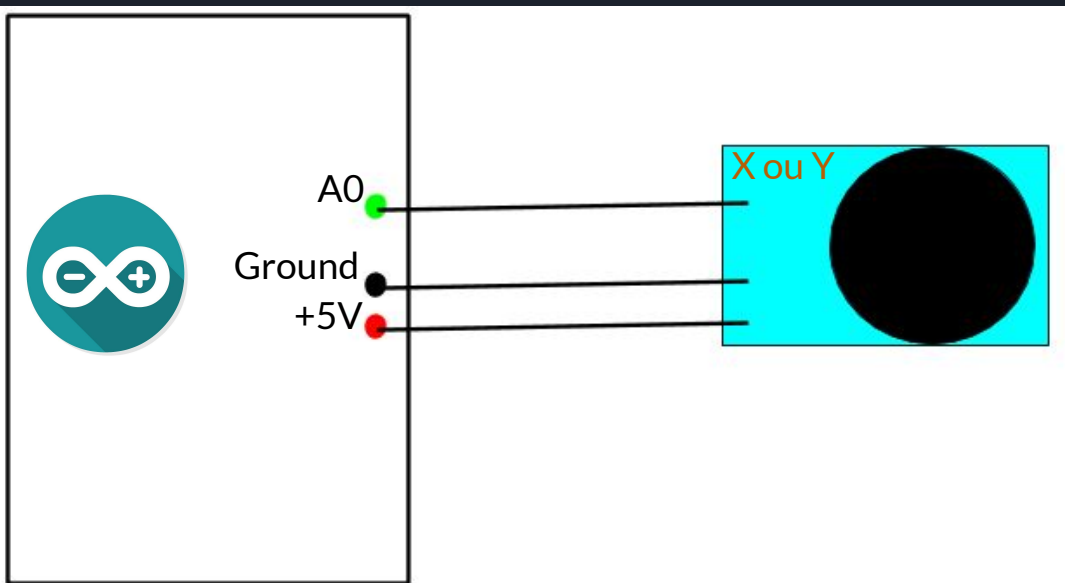
28 void setup()
29 {
30     //joystick
31     printArray(Serial.list());           // Ecrire sur la console serie le port serie disponible
32     myPort=new Serial(this, Serial.list()[1],9600); // Ouvrir le port choisi avec un débit de transmission
33     //
34     size(500, 500);
35     setup2();
36 }
37
38 // =====
39 void draw()
40 {
41     if(Val_pause ==1){
42
43         demarrer ();
44         affichage ();
45     }
46     else{
47
48         communication();
49         mouvement_barres ();
50         nettoyer();
51         bouger();
52         dessiner();
53         defaite();
54         pause();
55         VICTOIRE();
56
57     }
58 }
59
```

Chaîne de l'information

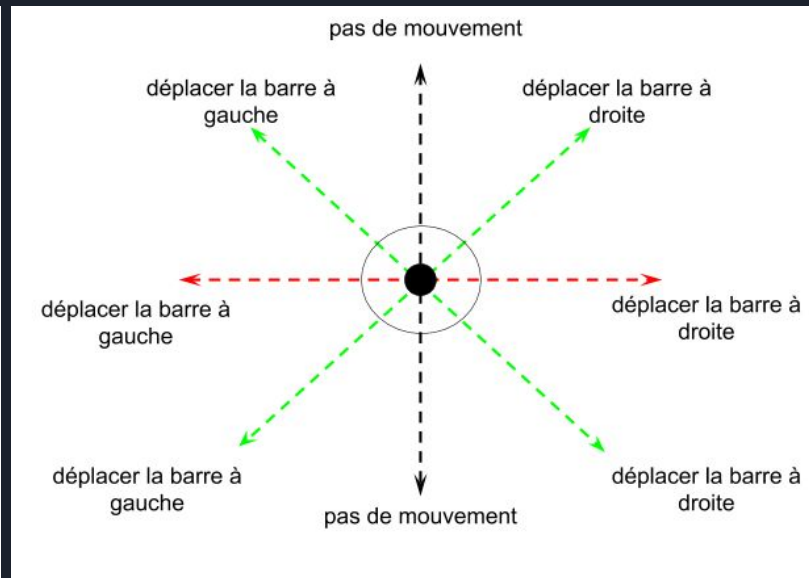


Différent schéma concernant le joystick

Electronique



Fonctionnement



Communication entre arduino et processing

```
28 void setup()
29 {
30   //joystick
31   printArray(Serial.list());           // Ecrire sur la console serie le port serie disponible
32   myPort=new Serial(this, Serial.list()[1],9600); // Ouvrir le port choisi avec un débit de transmission
33   //
```

communication_arduino \$

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int G = analogRead(A0);
  Serial.print(G);
  Serial.print(" ");
  Serial.println(1023);
  delay(50);
}
```

test	Affichage_Score	Arduino_commu	Brique	Defaite	Mouvement_Barre	Pause	Setup2	affichage_vict	demarrer
------	-----------------	---------------	--------	---------	-----------------	-------	--------	----------------	----------

```
1 void communication () {
2   message = myPort.readStringUntil(13);           //lire message mais à creuser au niveau du nombre 13 (saut à la ligne)
3   if (message != null)
4   {
5     //try                                           //le prg peut fonctionner sans
6     {
7       elements = splitTokens(message);           //décomposer le message reçu au niveau de l'espace
8       for (int i=0; i<elements.length && i<2; i++)
9       {
10        ValeursManettes[i] = int(elements[i]);    //stocke première partie du message dans ValeursManettes[0] puis ValeursManettes[1]
11      }
12    }
13    //catch (Exception e){}                       //le prg peut fonctionner sans
14  }
15  print("manette gauche ",ValeursManettes[0]);    //Ecrire valeur numérique manette gauche sur console Processing
16  print(" ");                                       //Ecrire plusieurs espace
17  println("manette droite désactivée ",ValeursManettes[1]); //Ecrire valeur numérique manette droite sur console Processing
18 }
```

manette	gauche	513	manette droite désactivée	1023
manette	gauche	513	manette droite désactivée	1023
manette	gauche	513	manette droite désactivée	1023
manette	gauche	513	manette droite désactivée	1023

Exploitation des valeurs

test

Affichage_Score

Arduino_commu

Brique

Defaite

Mouvement_Barre

```
1 void mouvement_barres () {  
2     if ((ValeursManettes[0]<350)&&(position_barre>50)){  
3         position_barre=position_barre-10;  
4         delay(10);  
5     }  
6     if ((ValeursManettes[0]>600)&&(position_barre<450)){  
7         position_barre=position_barre+10;  
8         delay(10);  
9     }  
10 }  
11  
12
```

Conclusion

