Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

Projet PONG : ANNEXES

Alexandre LUU

Nicolas MARTIN

Année 2013/2014

Groupe P1

Thierry LEQEU

Véronique AUGER

Études et Réalisation

Université François-Rabelais de Tours

Institut Universitaire de Technologie de Tours

Département Génie Électrique et Informatique Industrielle

UNIVERSITE FRANCOIS-RABELAIS
TOURS

Institut Universitaire de Technologie

Département
GENIE ELECTRIQUE ET
INFORMATIQUE INDUSTRIELLE

Projet PONG : ANNEXES

Alexandre LUU
Nicolas MARTIN
Année 2013/2014
Groupe P1

Thierry LEQEU
Véronique AUGER
Études et Réalisation

Annexe 1 : Planning prévisionnel

Semaine	1	2	3	4	5	6	7	8	9
Choix du projet, mise en place du planning	🟦			🟨	🟨				
	🟥			🟨	🟨				
Etude du matériel		🟦		🟨	🟨				
		🟥	🟥	🟨	🟨	🟥			
Programmation		🟦	🟦	🟨	🟨	🟦			
				🟨	🟨	🟥	🟥		
Programmation du joystick				🟨	🟨	🟦	🟦		
				🟨	🟨			🟥	
Rédaction du dossier				🟨	🟨		🟦	🟦	
				🟨	🟨			🟥	🟥
Remise du dossier				🟨	🟨				🟦
				🟨	🟨				🟥

🟨 Vacances

🟦 Planning prévisionnel

🟥 Planning réel

Annexe 2 : Bibliothèque OLED128.h

```
1    //uOLED-128-Glh Driver for Arduino
2    //
3    //from uOLED-160-GMD1 driver
4    //original code: Oscar Gonzalez
5    //August 2007
6    //
7    //update: Alexandre LUU et Nicolas MARTIN
8    //For Game PONG
9    //Reference document: GOLDELOX-SGC-COMMANDS-SIS-Rev6.pdf
10   //March 2014
11   //work with Arduino 1.0.5
12   //
13   #define OLED_BAUDRATE 19200
14   #define OLED_RESETPIN 7 //reset pin
15   #define JOYSTICK_UP            6       //touche UP du joystick
16   #define JOYSTICK_PRESS         5       //touche PRESS du joystick
17   #define JOYSTICK_RIGHT         4       //touche RIGHT du joystick
18   #define JOYSTICK_LEFT          3       //touche LEFT du joystick
19   #define JOYSTICK_DOWN          2       //touche DOWN du joystick
20
21   //
22   void OLED_ResetDisplay()
23   {
24     digitalWrite(OLED_RESETPIN, LOW);
25     delay(20);
26     digitalWrite(OLED_RESETPIN, HIGH);
27     delay(20);
28   }
29
30   //Initialisation display. Must be first activate a serial comunication!
31   void OLED_Init()
32   {
33     OLED_ResetDisplay();
34     delay(5000);
35     Serial.write(0x55); //AutoBaud
36   }
37
38   // Clear Screen
39   void OLED_Clear()
40   {
41     Serial.write(0x45);
42     delay(40);
43   }
44
45
```

```
46
47    void OLED_PenSize(char filled)
48    {
49      if (filled == 1)
50      {
51        Serial.write(0x70);
52        Serial.write(byte(0x00)); //filled(solid)
53      }
54      else
55      {
56        Serial.write(0x70);
57        Serial.write(0x01); //outline(wire-frame)
58      }
59      delay(20);
60    }
61
62    void OLED_DrawCircle(char x,char y,char radius,char filled,int color)
63    {
64      OLED_PenSize(filled);
65      Serial.write(0x43);
66      Serial.write(byte(x));
67      Serial.write(byte(y));
68      Serial.write(radius);
69      Serial.write(color>>8); //MSB
70      Serial.write(color&0xFF); //LSB
71      delay(20);
72
73    }
74    //
75    void OLED_DrawTriangle(char x1,char y1,char x2,char y2,char x3,char y3,char filled,int color)
76    {
77      OLED_PenSize(filled);
78      Serial.write(0x47);
79      Serial.write(byte(x1));
80      Serial.write(byte(y1));
81      Serial.write(byte(x2));
82      Serial.write(byte(y2));
83      Serial.write(byte(x3));
84      Serial.write(byte(y3));
85      Serial.write(color>>8); //MSB
86      Serial.write(color&0xFF); //LSB
87      delay(20);
```

```
47    void OLED_PenSize(char filled)
48    {
49      if (filled == 1)
50      {
51        Serial.write(0x70);
52        Serial.write(byte(0x00)); //filled(solid)
53      }
54      else
55      {
56        Serial.write(0x70);
57        Serial.write(0x01); //outline(wire-frame)
58      }
59      delay(20);
60    }
61
62    void OLED_DrawCircle(char x,char y,char radius,char filled,int color)
63    {
64      OLED_PenSize(filled);
65      Serial.write(0x43);
66      Serial.write(byte(x));
67      Serial.write(byte(y));
68      Serial.write(radius);
69      Serial.write(color>>8); //MSB
70      Serial.write(color&0xFF); //LSB
71      delay(20);
72
73    }
74    //
75    void OLED_DrawTriangle(char x1,char y1,char x2,char y2,char x3,char y3,char filled,int color)
76    {
77      OLED_PenSize(filled);
78      Serial.write(0x47);
79      Serial.write(byte(x1));
80      Serial.write(byte(y1));
81      Serial.write(byte(x2));
82      Serial.write(byte(y2));
83      Serial.write(byte(x3));
84      Serial.write(byte(y3));
85      Serial.write(color>>8); //MSB
86      Serial.write(color&0xFF); //LSB
87      delay(20);
88    }
```

```
90    void OLED_DrawLine(char x1,char y1,char x2,char y2,int color)
91     {
92       Serial.write(0x4C);
93       Serial.write(x1);
94       Serial.write(y1);
95       Serial.write(x2);
96       Serial.write(y2);
97       Serial.write(color>>8); //MSB
98       Serial.write(color&0xFF); //LSB
99       delay(20);
100    }
101    //
102    void OLED_DrawRectangle(char x,char y,char width,char height,char filled,int color)
103     {
104       OLED_PenSize(filled);
105       Serial.write(0x72);
106       Serial.write(x); //x1
107       Serial.write(y); //y1
108       Serial.write(x+width); //x2
109       Serial.write(y+height); //y2
110       Serial.write(color>>8); //MSB
111       Serial.write(color&0xFF); //LSB
112       delay(20);
113    }
114    //
115    void OLED_Opacity(char mode)
116     {
117       Serial.write(0x4F);
118       Serial.write(byte(mode));
119       delay(20);
120    }
```

```
122    void OLED_DrawStringText(char x,char y,char font,int stringColour,int width,int height,char *string)
123    {
124      Serial.write(0x53);
125      Serial.write(byte(x));
126      Serial.write(byte(y));
127      Serial.write(font);
128      Serial.write(stringColour>>8);   //MSB
129      Serial.write(stringColour&0xFF);   //LSB
130      Serial.write(width);
131      Serial.write(height);
132      for (int i=0;i<strlen(string);i++)
133      {
134        Serial.write(string[i]);
135      }
136      Serial.write(byte(0x00));
137      delay(20);
138    }
139
140    void ConfigJoystick(void)
141  {
142    pinMode(JOYSTICK_UP,INPUT);//broche D6 en entrée (joystick UP)
143    digitalWrite(JOYSTICK_UP,HIGH); //R pull-up active !
144    pinMode(JOYSTICK_PRESS,INPUT);// joystick PRESS
145    digitalWrite(JOYSTICK_PRESS,HIGH);
146    pinMode(JOYSTICK_RIGHT,INPUT); // joystick RIGHT
147    digitalWrite(JOYSTICK_RIGHT,HIGH);
148    pinMode(JOYSTICK_LEFT,INPUT); // joystick LEFT
149    digitalWrite(JOYSTICK_LEFT,HIGH);
150    pinMode(JOYSTICK_DOWN,INPUT); // joystick DOWN
151    digitalWrite(JOYSTICK_DOWN,HIGH);
152  }
```

Annexe 3 : Programme du jeu

Annexe 4 : Schéma de l'arduino