МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. ТАРАСА ШЕВЧЕНКА
ФІЗИЧНИЙ ФАКУЛЬТЕТ

ЗВІТ ДО ЛАБОРАТОРНОЇ РОБОТИ №4 (CRAZY SNAKE AND ULTIMATE TETRIS)

Виконали:

студент 2 курсу 5-А групи спеціальності
104 «Фізика та астрономія»
Свінтозельський Володимир Ярославович

студентка 2 курсу 5-А групи спеціальності
104 «Фізика та астрономія»
Бучинська Марія Євгенівна

студентка 2 курсу 5-А групи спеціальності
104 «Фізика та астрономія»
Топчій Поліна

Науковий керівник:

викладач
Єрмоленко Руслан Вікторович

Київ – 2020

РОЛІ АВТОРІВ

У цій лабораторній роботі брали участь такі студенти:

- Свінтозельський Володимир Ярославович
  - *концепт*
  - *сборка електричної схеми*
  - *написання коду програми*
  - *відеофіксація легендарного досягнення*
  - *проектування джойстика*
  - *виготовлення джойстика*
  - *оформлення звіту*
- Бучинська Марія Євгенівна
  - *відлагодження електричної схеми*
  - *дизайн*
  - *концепт*
  - *виготовлення джойстика*
  - *написання коду програми*
- Топчій Поліна
  - *відлагодження електричної схеми*
  - *дизайн*
  - *концепт*
  - *політична робота*

ЗМІСТ

ВСТУП

У даній роботі нами було створено ігри епохи 00х: змійка та тетріс! Створено це чудо програмної та інженерної думки на базі arduino та світлодіодної матриці 8х8.

Для гри у змійку було виготовлено джойстик 1 із кнопок, проводів, макетної плати та рук, що нагло претендують на звання "прямі". По суті, він являв собою кнопки, які при натисканні притягували відповідні виходи до землі (ардуінка була у режимі $INPUT_PULLUP$). Нажаль, до того часу, як почалась розробка тетрісу, даний джойстик не дожив :(

Рис. 1: Джойстик, що використовувався у змійці.

РОЗДІЛ 1
БЛИЖЧЕ ДО СПРАВИ

Оскільки тут описувати по суті нічого, адже автори щиро вірять у казочку, що усі грали в ці ігри, перейдемо відразу до технічних деталей.

1.1 Підключення матриці

Розпіновка світлодіодної матриці показана на 1.1

VCC	GND	CIN	CS	CLK
+5V	GND	11	9	13

Рис. 1.1: Розпіновка світлодіодної матриці.

1.2 Змійка

Відео гри одного з авторів у отриману змійку знаходиться у нашій папці на гітхабі.

Програмний код:

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>

int pinCS = 9;
int numberOfHorizontalDisplays = 1;
int numberOfVerticalDisplays = 1;

```

```
Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays,
    numberOfVerticalDisplays);
struct coord{
  int x;
  int y;
};
coord eat={random(8), random(8)};
int len=1;
coord*snake= new coord [64];
void setup() {
  randomSeed(analogRead(0));
    matrix.setIntensity(4);
    pinMode(2,INPUT);
    pinMode(3,INPUT);
    pinMode(4,INPUT);
    pinMode(5,INPUT);
    PORTD = 0b00111100;
    Serial.begin(9600);

snake[0]={3,0};
snake[1]={2,0};
snake[2]={1,0};
snake[3]={0,0};
}
int  x=0, y=0;
int dirX=1, dirY=0;


void printdigit(int d)
{
  for(int i=0; i<8; i++)
  {
    for(int j=0; j<8; j++)
      matrix.drawPixel(i,j, LOW);
    matrix.write();
  }
  int a=d/10;
  for(int k=0; k<2; k++)
  {
  switch(a)
  {
    case 0:
    for(int i=1; i<6; i++)
    {
    matrix.drawPixel(i,4*k, HIGH);
    matrix.drawPixel(i,2+4*k, HIGH);
```

```
54      }
55      matrix.drawPixel(1,1+4*k, HIGH);
56      matrix.drawPixel(5,1+4*k, HIGH);
57      break;
58
59      case 1:
60      for(int i=1; i<6; i++)
61      matrix.drawPixel(i,2+4*k, HIGH);
62      break;
63
64      case 2:
65      for(int i=1; i<6; i++)
66      {
67        if(i!=4)
68        { matrix.drawPixel(i,4*k, HIGH);
69          if(i!=2) matrix.drawPixel(i,1+4*k, HIGH);
70        }
71      if(i!=2)
72      matrix.drawPixel(i,2+4*k, HIGH);
73      }
74      break;
75
76      case 3:
77      for(int i=1; i<6; i++)
78      {
79      matrix.drawPixel(i,2+4*k, HIGH);
80      if(i%2==1)
81      {matrix.drawPixel(i,4*k, HIGH);
82      matrix.drawPixel(i,1+4*k, HIGH);}
83      }
84      break;
85
86      case 4:
87      for(int i=1; i<6; i++)
88      {
89      matrix.drawPixel(i,2+4*k, HIGH);
90      if(i>2) matrix.drawPixel(i,4*k, HIGH);
91      }
92      matrix.drawPixel(3,1+4*k, HIGH);
93      break;
94
95      case 5:
96
97      for(int i=1; i<6; i++)
98      {
99        if(i!=2)
```

```
      { matrix.drawPixel(i,4*k, HIGH);
        if(i!=4) matrix.drawPixel(i,1+4*k, HIGH);
      }
    if(i!=4)
    matrix.drawPixel(i,2+4*k, HIGH);
    }
    break;

    case 6:
    for(int i=1; i<6; i++)
    {
    matrix.drawPixel(i,4*k, HIGH);
    matrix.drawPixel(i,2+4*k, HIGH);
    if(i%2==1) matrix.drawPixel(i,1+4*k, HIGH);
    }
    matrix.drawPixel(4,2+4*k, LOW);
    break;

    case 7:
    for(int i=1; i<6; i++)
    {
    matrix.drawPixel(i,2+4*k, HIGH);
    }
    matrix.drawPixel(5,4*k, HIGH);
    matrix.drawPixel(5,1+4*k, HIGH);
    break;

    case 8:
    for(int i=1; i<6; i++)
    {
    matrix.drawPixel(i,4*k, HIGH);
    matrix.drawPixel(i,2+4*k, HIGH);
    if(i%2==1) matrix.drawPixel(i,1+4*k, HIGH);
    }
    break;

    case 9:
    for(int i=1; i<6; i++)
    {
    matrix.drawPixel(i,4*k, HIGH);
    matrix.drawPixel(i,2+4*k, HIGH);
    if(i%2==1) matrix.drawPixel(i,1+4*k, HIGH);
    }
    matrix.drawPixel(2,4*k, LOW);
    break;
  }
```

```
  a=d%10;
  }
  matrix.write();
}


void loop() {


snake[0].x=(snake[0].x+8)%8;
 snake[0].y=(snake[0].y+8)%8;
for(int i=0; i<len; i++)
    {
      matrix.drawPixel((snake[i].x+8)%8, (snake[i].y+8)%8, LOW);
    }
    matrix.write();

for(int j=len-1;j>0; j--)
{
  snake[j].x=(snake[j].x+8)%8;
  snake[j].y=(snake[j].y+8)%8;
  if(snake[j].x != snake[j-1].x || snake[j].y != snake[j-1].y)
  {
   snake[j].x=snake[j-1].x;
   snake[j].y=snake[j-1].y;
  }
 }


snake[0].x+=dirX;
snake[0].y+=dirY;

snake[0].x = (snake[0].x + 8) % 8;
snake[0].y = (snake[0].y + 8) % 8;

 for(int i=0; i<len; i++)
 {
      matrix.drawPixel((snake[i].x+8)%8, (snake[i].y+8)%8, HIGH);
 }
  matrix.drawPixel(eat.x, eat.y, HIGH);
    matrix.write();
     for(int k=0; k<len; k++)
      {
        for( int j=k+1; j<len; j++)
        {
          if(snake[k].x==snake[j].x&&snake[k].y==snake[j].y)
```

```
          {
            printdigit(len);
            while(1==1)
            {

            }
          }
        }
      }

    if(eat.x==snake[0].x&&eat.y==snake[0].y)
    {
      snake[len].x=snake[len-1].x;
      snake[len].y=snake[len-1].y;
      len++;
       matrix.drawPixel(eat.x, eat.y, LOW);
      eat={random(8), random(8)};
       matrix.drawPixel(eat.x, eat.y, HIGH);
      matrix.write();
      }
    for(int k=0;k<500;k++)
    {
      if(k%100==0)
      {
       matrix.drawPixel(snake[0].x, snake[0].y, LOW);
       matrix.drawPixel(eat.x, eat.y, LOW);
       matrix.write();
      }
      if(k%100==50)
      {
       matrix.drawPixel(snake[0].x, snake[0].y, HIGH);
       matrix.drawPixel(eat.x, eat.y, HIGH);
       matrix.write();
      }


if(digitalRead(2)==LOW&&dirX!=-1)
{  dirX=1;
  dirY=0;}
  if(digitalRead(3)==LOW&&dirY!=-1)
{  dirX=0;
  dirY=1;}
  if(digitalRead(4)==LOW&&dirX!=1)
{  dirX=-1;
  dirY=0;}
  if(digitalRead(5)==LOW&&dirY!=1)
```

```
{  dirX=0;
  dirY=-1;}
  delay(1);
  }



}
```

1.3 Тетріс

Відео гри одного з авторів у отриманий тетріс знаходиться у нашій папці на гітхабі.

Програмний код:

```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>

int pinCS = 9;
int numberOfHorizontalDisplays = 1;
int numberOfVerticalDisplays = 1;

Max72xxPanel matrix = Max72xxPanel(pinCS, numberOfHorizontalDisplays,
    numberOfVerticalDisplays);

// 4 5 6
// 2 3
int** matrixMask = new int*[8];
short score = 0;

void setup() {
  randomSeed(analogRead(A0));
  pinMode(4,INPUT_PULLUP);
  pinMode(5,INPUT_PULLUP);
  pinMode(6,INPUT_PULLUP);
  pinMode(2,INPUT_PULLUP);
  pinMode(3,INPUT_PULLUP);
  matrix.setIntensity(0);
  for(int i =0;i<8;i++){
    matrixMask[i] = new int[8];
    for(int j = 0;j<8;j++)
      matrixMask[i][j]=0;
  }
  matrix.fillScreen(LOW);
  matrix.write();
```

```
  Serial.begin(9600);
}

void printdigit(int d)
{
  matrix.setRotation( 0, 1 );
  for(int i=0; i<8; i++)
  {
    for(int j=0; j<8; j++)
      matrix.drawPixel(i,j, LOW);
    matrix.write();
  }
  int a=d/10;
  for(int k=0; k<2; k++)
  {
  switch(a)
  {
    case 0:
    for(int i=1; i<6; i++)
    {
    matrix.drawPixel(i,4*k, HIGH);
    matrix.drawPixel(i,2+4*k, HIGH);
    }
    matrix.drawPixel(1,1+4*k, HIGH);
    matrix.drawPixel(5,1+4*k, HIGH);
    break;

    case 1:
    for(int i=1; i<6; i++)
    matrix.drawPixel(i,2+4*k, HIGH);
    break;

    case 2:
    for(int i=1; i<6; i++)
    {
      if(i!=4)
      { matrix.drawPixel(i,4*k, HIGH);
        if(i!=2) matrix.drawPixel(i,1+4*k, HIGH);
      }
    if(i!=2)
    matrix.drawPixel(i,2+4*k, HIGH);
    }
    break;

    case 3:
    for(int i=1; i<6; i++)
```

```
77      {
78      matrix.drawPixel(i,2+4*k, HIGH);
79      if(i%2==1)
80      {matrix.drawPixel(i,4*k, HIGH);
81      matrix.drawPixel(i,1+4*k, HIGH);}
82      }
83      break;
84
85      case 4:
86      for(int i=1; i<6; i++)
87      {
88      matrix.drawPixel(i,2+4*k, HIGH);
89      if(i>2) matrix.drawPixel(i,4*k, HIGH);
90      }
91      matrix.drawPixel(3,1+4*k, HIGH);
92      break;
93
94      case 5:
95
96      for(int i=1; i<6; i++)
97      {
98        if(i!=2)
99        { matrix.drawPixel(i,4*k, HIGH);
100         if(i!=4) matrix.drawPixel(i,1+4*k, HIGH);
101       }
102     if(i!=4)
103     matrix.drawPixel(i,2+4*k, HIGH);
104     }
105     break;
106
107     case 6:
108     for(int i=1; i<6; i++)
109     {
110     matrix.drawPixel(i,4*k, HIGH);
111     matrix.drawPixel(i,2+4*k, HIGH);
112     if(i%2==1) matrix.drawPixel(i,1+4*k, HIGH);
113     }
114     matrix.drawPixel(4,2+4*k, LOW);
115     break;
116
117     case 7:
118     for(int i=1; i<6; i++)
119     {
120     matrix.drawPixel(i,2+4*k, HIGH);
121     }
122     matrix.drawPixel(5,4*k, HIGH);
```

```
123     matrix.drawPixel(5,1+4*k, HIGH);
124     break;
125 
126     case 8:
127     for(int i=1; i<6; i++)
128     {
129     matrix.drawPixel(i,4*k, HIGH);
130     matrix.drawPixel(i,2+4*k, HIGH);
131     if(i%2==1) matrix.drawPixel(i,1+4*k, HIGH);
132     }
133     break;
134 
135     case 9:
136     for(int i=1; i<6; i++)
137     {
138     matrix.drawPixel(i,4*k, HIGH);
139     matrix.drawPixel(i,2+4*k, HIGH);
140     if(i%2==1) matrix.drawPixel(i,1+4*k, HIGH);
141     }
142     matrix.drawPixel(2,4*k, LOW);
143     break;
144   }
145   a=d%10;
146   }
147   matrix.write();
148 }
149 
150 
151 struct Figure {
152   int nx;
153   int ny;
154   int** mask;
155 
156   ~Figure(){
157     for(int i = 0;i<ny;i++)
158       delete [] mask[i];
159     delete[] mask;
160   }
161   Figure(int nx, int ny, int** mask): nx(nx),ny(ny),mask(mask){}
162   Figure(int type){
163     mask = new int*[2];
164     for(int i =0;i<2;i++){
165       mask[i] = new int[3];
166       for(int j =0;j<3;j++)
167         mask[i][j]=0;
168     }
```

```
169     if(type == 1){
170       nx = 1;
171       ny =1;
172       mask[0][0] = 1;
173     }
174     if(type == 2){
175       nx = 3;
176       ny =2;
177       mask[0][0] = 1;
178       mask[0][1] = 1;
179       mask[1][1] = 1;
180       mask[1][2] = 1;
181     }
182     if(type == 3){
183       nx = 3;
184       ny =2;
185       mask[1][0] = 1;
186       mask[0][1] = 1;
187       mask[1][1] = 1;
188       mask[0][2] = 1;
189     }
190     if(type == 4){
191       nx = 3;
192       ny =2;
193       mask[0][1] = 1;
194       mask[1][0] = 1;
195       mask[1][1] = 1;
196       mask[1][2] = 1;
197     }
198     if(type == 5){
199       nx = 3;
200       ny =2;
201       mask[0][0] = 1;
202       mask[1][0] = 1;
203       mask[1][1] = 1;
204       mask[1][2] = 1;
205     }
206     if(type == 6){
207       nx = 3;
208       ny =2;
209       mask[0][0] = 1;
210       mask[1][0] = 1;
211       mask[0][1] = 1;
212       mask[0][2] = 1;
213     }
214     if(type == 7){
```

```
      nx = 3;
      ny =1;
      mask[0][0] = 1;
      mask[0][1] = 1;
      mask[0][2] = 1;
    }
    if(type == 8){
      nx = 2;
      ny =2;
      mask[0][0] = 1;
      mask[0][1] = 1;
      mask[1][0] = 1;
    }
    if(type == 9){
      nx = 2;
      ny =2;
      mask[0][0] = 1;
      mask[0][1] = 1;
      mask[1][0] = 1;
      mask[1][1] = 1;
    }
  }
};

bool hasActiveFig = false;
Figure* activef;
int x=0;
int y=0;

bool isSomethingUnder(){
  if(y == activef->ny-1) {Serial.println("GROUND");return true;}
  for(int j =0;j<activef->nx;j++){
    for(int i = activef->ny-1;i>=0;i--){
      if(activef->mask[i][j]){
        if(matrixMask[y-i-1][x+j])
          return true;
        break;
      }
    }
  }
  return false;
}

bool isSomethingRight(){
  if(x == 8-activef->nx) {Serial.println("Right border");return true;}
  for(int j =0;j<activef->ny;j++){
```

```
    for(int i = activef->nx-1;i>=0;i--){
      if(activef->mask[j][i]){
        if(matrixMask[y-j][x+i+1])
          return true;
        break;
      }
    }
  }
  return false;
}

bool isSomethingLeft(){
  if(x == 0) {Serial.println("Left border");return true;}
  for(int j =0;j<activef->ny;j++){
    for(int i = 0;i<activef->nx;i++){
      if(activef->mask[j][i]){
        if(matrixMask[y-j][x+i-1])
          return true;
        break;
      }
    }
  }
  return false;
}

void removeLine(int line){
  for(int j = line;j<6;j++)
    for(int i =0;i<8;i++)
      matrixMask[j][i] = matrixMask[j+1][i];
}

void checkIsFullLineExist(){
  bool isfull;
  for(int j =0;j<6;j++){
    isfull = true;
    for(int i =0;i<8;i++){
      if(!matrixMask[j][i]){
        isfull = false;
        break;
      }
    }
    if(isfull)
      removeLine(j);
  }
}

```

```
307 void endGame(){
308   matrix.fillScreen(LOW);
309   printdigit(score);
310   matrix.write();
311   while(true){}
312 }
313 
314 void moveActive(){
315   bool check = isSomethingUnder();
316   Serial.println(check);
317   if(check){
318     if(y >= 6) endGame();
319     for(int i = y;i>-activef->ny+y;i--)
320       for(int j =x;j<activef->nx+x;j++)
321         matrixMask[i][j] = (activef->mask[y-i][j-x] || matrixMask[i][j]);
322     hasActiveFig=false;
323     checkIsFullLineExist();
324     return;
325   }
326   y--;
327 }
328 
329 void falldown(){
330   while(hasActiveFig)
331     moveActive();
332 }
333 
334 void graph(){
335   matrix.fillScreen(LOW);
336   for(int i = 0;i<8;i++)
337     for(int j = 0;j<8;j++)
338       matrix.drawPixel(i, j, matrixMask[j][i]);
339   for(int i = 0;i<activef->nx;i++)
340     for(int j = 0;j<activef->ny;j++)
341       if(activef->mask[j][i])
342         matrix.drawPixel(x+i,y-j, activef->mask[j][i]);
343   matrix.write();
344 }
345 
346 void rotateRight(){
347   int xsize = activef->nx;
348   int ysize = activef->ny;
349   Figure* rotatedF = new Figure(ysize,xsize,0);
350   rotatedF->mask = new int*[xsize];
351   for(int i = 0;i<xsize;i++){
352     rotatedF->mask[i] = new int[ysize];
```

```
353     for(int j =0;j<ysize;j++)
354       rotatedF->mask[i][j]=0;
355   }
356   for(int i = 0;i<xsize;i++)
357     for(int j = 0;j<ysize;j++)
358       rotatedF->mask[i][j]=activef->mask[ysize-j-1][i];
359 
360   activef = rotatedF;
361 }
362 
363 void wait(){
364   for(int t = 0;t<4;t++){
365     bool turnRight = !digitalRead(4);
366     bool turnDown = !digitalRead(3);
367     bool turnLeft =! digitalRead(2);
368     bool rotRight =! digitalRead(5);
369     bool rotLeft = !digitalRead(6);
370     if(turnRight && !turnLeft)
371       if(!isSomethingRight()){
372         x++;
373         graph();
374       }
375     if(!turnRight && turnLeft)
376       if(!isSomethingLeft()){
377         x--;
378         graph();
379       }
380     if(turnDown){
381       falldown();
382       graph();
383     }
384     if(rotRight && !rotLeft){
385       rotateRight();
386       graph();
387     }
388     delay(150);
389   }
390 }
391 
392 void loop() {
393   if(!hasActiveFig){
394     score ++;
395     Serial.print("creating new figure with type: ");
396     hasActiveFig=true;
397     int type = random(1,10);
398     Serial.print(type);
```

```
399     Serial.print("    ");
400     activef = new Figure(type);
401     x = 4;
402     y = 7;
403     Serial.println(type);
404     graph();
405     wait();
406   }
407 
408   moveActive();
409   Serial.print("moved  ");
410   Serial.print(x);
411   Serial.print("  ");
412   Serial.println(y);
413 
414   graph();
415   wait();
416 }
```

ВИСНОВКИ

В процесі виконання даної роботи автори відточили навички з написання не завжди корисних програм на дитячій платформі arduino. Ну і звісно поморозважалися! :)

Автори надзвичайно вдячні викладачу Єрмоленку Р. В. за надану ардуінку, матрицю, та кучу барахла до неї! ~~Ну хоч одне речення без сарказму..~~