int a=0;

int cur\_i=0;

int b=4;

int cur\_j=4;

int i=0,j=0;

int array[16][8]={0};//empty array

int shape1[3][3]={{1,1,0}, {0,1,0},{0,1,1}};//z

int shape2[3][3]={{1,1,1}, {1,1,1},{1,1,1}};//square

int shape3[3][3]={{1,1,1}, {0,1,0},{0,1,0}};//T

int shape4[3][3]={{0,0,0}, {0,0,0},{1,1,1}};//HORIZONTAL BAR

int shape0[3][3]={{1,0,0}, {1,0,0},{1,1,1}};//L

void setup()

{

pinMode(10,OUTPUT); //assuming these are the pins to be connected

pinMode(11,OUTPUT);

pinMode(12,OUTPUT);

pinMode(13,OUTPUT);

pinMode(5,OUTPUT);

pinMode(6,OUTPUT);

pinMode(7,OUTPUT);

randomSeed(20);

pinMode(19,INPUT);

pinMode(20,INPUT);

pinMode(21,INPUT);

attachInterrupt(digitalPinToInterrupt(19),moveleft,CHANGE); //for buttons

attachInterrupt(digitalPinToInterrupt(20),moveright,CHANGE);

attachInterrupt(digitalPinToInterrupt(21),rotate,CHANGE);

}

/\* to display the matrix fed to it\*/

void display(int X[16][8])

{

int i=0,j=0;

for(i=0;i<16;i++)

{

for(j=0;j<8;j++)

{

if(X[i][j]==1)

{

int r0=bitRead(j,0); //A0

int r1=bitRead(j,1); //A1 for cathode mux

int r2=bitRead(j,2); //A2

int k0=bitRead(i,0); //A0

int k1=bitRead(i,1); //A1 for anode mux

int k2=bitRead(i,2); //A2

int k3=bitRead(i,3); //A3

digitalWrite(5,r0);

digitalWrite(6,r1);

digitalWrite(7,r2);

digitalWrite(10,k0);

digitalWrite(11,k1);

digitalWrite(12,k2);

digitalWrite(13,k3);

}

delay(1);

}

}

}

/\*to move down the block\*/

void movedown(int arr[16][8])

{

for(i=a+2;i>=a;i--)

{

for(j=cur\_j;j<cur\_j+3;j++)

{

arr[i+1][j]=arr[i][j];

}

}

for(j=cur\_j;j<cur\_j+3;j++)

arr[a][j]=0;

a++;

cur\_i=a;

}

/\*to check if block could be moved down\*/

int check\_movedown(int arr[16][8])

{

if(cur\_i>=13)return 0;

for(j=cur\_j;j<cur\_j+3;j++)

{

if(arr[cur\_i+2][j]==arr[cur\_i+3][j]&&arr[cur\_i+2][j]==1)

{

if(cur\_i==0)

{

turn\_off(arr);

}

return 0;

}

}

return 1;

}

/\*to check if block could be moved left\*/

int check\_moveleft(int arr[16][8])

{

if(cur\_j==0)return 0;

for(i=cur\_i;j<cur\_i+3;i++)

{

if(arr[i][cur\_j]==arr[i][cur\_j-1]&&arr[i][cur\_j]==1)return 0;

}

return 1;

}

/\*to check if block could be moved right\*/

int check\_moveright(int arr[16][8])

{

if(cur\_j+2==7)return 0;

for(i=cur\_i;j<cur\_i+3;i++)

{

if(arr[i][cur\_j+2]==arr[i][cur\_j+3]&&arr[i][cur\_j+2]==1)return 0;

}

return 1;

}

/\*to move left the block\*/

void moveleft()

{

if(check\_moveleft(array)==0)return;

digitalWrite(19,RISING);

for(j=b;j<b+3;j++)

{

for(i=cur\_i;i<cur\_i+3;i++)

{

array[i][j-1]=array[i][j];

}

}

for(i=cur\_i;i<cur\_i+3;i++)

array[i][b+2]=0;

b--;

cur\_j=b;

for(int f=0;f<10;f++)

display(array);

}

/\*to move right the block\*/

void moveright()

{

if(check\_moveright(array)==0)return;

digitalWrite(20,RISING);

for(j=b+3;j>=b;j--)

{

for(i=cur\_i;i<cur\_i+3;i++)

{

array[i][j+1]=array[i][j];

}

}

for(i=cur\_i;i<cur\_i+3;i++)

array[i][b]=0;

b++;

cur\_j=b;

for(int f=0;f<10;f++)

display(array);

}

void swap(int \*a,int \*b)

{

int tmp=\*a;

\*a=\*b;

\*b=tmp;

}

/\*to rotate the block\*/

void rotate()

{

digitalWrite(21,RISING);

i=cur\_i;

j=cur\_j;

swap(&array[cur\_i+1][cur\_j],&array[cur\_i][cur\_j+1]);

swap(&array[cur\_i+2][cur\_j],&array[cur\_i][cur\_j+2]);

swap(&array[cur\_i+2][cur\_j+1],&array[cur\_i+1][cur\_j+2]);

for(i=cur\_i;i<cur\_i+3;i++)

{

int tmp=array[i][cur\_j];

array[i][cur\_j]=array[i][cur\_j+2];

array[i][cur\_j+2]=tmp;

}

for(int f=0;f<10;f++)

display(array);

}

/\*to add new block to the matrix\*/

void copy(int arr[16][8],int shape[3][3])

{

for(i=cur\_i;i<cur\_i+3;i++)

{

for(j=cur\_j;j<cur\_j+3;j++)

{

arr[i][j]=shape[i-cur\_i][j-cur\_j];

}

}

}

/\*to shift the matrix if whole row is filled\*/

void shift\_matrix(int arr[16][8],int row)

{

for(j=0;j<8;j++)

{

for(i=row-1;i>=0;i--)

{

arr[i+1][j]=arr[i][j];

}

}

for(j=0;j<8;j++)

{

arr[0][j]=0;

}

}

/\*if GAME OVER turn off all LEDs\*/

void turn\_off(int arr[16][8])

{

for(i=0;i<16;i++)

{

for(j=0;j<8;j++)

{

arr[i][j]=0;

}

}

arr[15][2]=0;

while(1)

display(arr);

}

void loop()

{

int randomnum=random(0,5); //to send random shape

switch(randomnum){

case 0:copy(array,shape0);

for(int f=0;f<5;f++)

display(array);

break;

case 1:copy(array,shape1);

for(int f=0;f<5;f++)

display(array);

break;

case 2:copy(array,shape2);

for(int f=0;f<5;f++)

display(array);

break;

case 3:copy(array,shape3);

for(int f=0;f<5;f++)

display(array);

break;

case 4:copy(array,shape4);

for(int f=0;f<5;f++)

display(array);

break;

}

for(i=0;i<16;i++) //check if matrix is to be shifted

{

for(j=0;j<8;j++)

{

if(array[i][j]==0)break;

}

if(j==8)

{

shift\_matrix(array,i);

}

}

while(check\_movedown(array)) //move down the block if possible

{

movedown(array);

for(int f=0;f<10;f++)

display(array);

delay(10);

}

b=3;

cur\_j=3;

a=0;

cur\_i=0;

}