

```

#include <FastLED.h>

#define NUM_LEDS 100
#define DATA_PIN 5
#define PIN_ENCODER_A 2
#define PIN_ENCODER_B 3
#define X_SIZE 14
#define Y_SIZE 15

int buttonState = 0;
CRGB leds[NUM_LEDS];
unsigned char encoder_A;
unsigned char encoder_B;
unsigned char encoder_A_prev=0;

struct Cell {
    int8_t pin;
    bool food;
};

struct PacMan {
    int8_t x;
    int8_t y;
    int8_t directionFacing;
    int8_t lives;
} pacman = {10,10,4,3};

struct Cell cells[15][15];

void setup()
{
    FastLED.addLeds<WS2812B, DATA_PIN, GRB>(leds, NUM_LEDS);
    pinMode(PIN_ENCODER_A, INPUT_PULLUP);
    pinMode(PIN_ENCODER_B, INPUT_PULLUP);

    for(int x = 0; x < X_SIZE; x++) {
        for(int y = 0; y < Y_SIZE; y++) {
            cells[x][y] = { -1, false };
        }
    }

    cells[0][0] = { 0, true };
    cells[0][1] = { 1, true };
    cells[0][2] = { 2, true };
    cells[0][3] = { 3, true };
    cells[0][4] = { 4, true };
    cells[0][5] = { 5, true };
    cells[0][6] = { 6, true };
    cells[0][7] = { 7, true };
    cells[0][8] = { 8, true };
    cells[0][9] = { 9, true };
    cells[0][10] = { 10, true };
    cells[0][11] = { 11, true };
    cells[0][12] = { 12, true };
    cells[0][13] = { 13, true };
    cells[0][14] = { 14, true };

```

```
cells[1][0] = { 18, true };
cells[1][4] = { 17, true };
cells[1][10] = { 16, true };
cells[1][14] = { 15, true };
```

```
cells[2][0] = { 19, true };
cells[2][4] = { 20, true };
cells[2][10] = { 21, true };
cells[2][14] = { 22, true };
```

```
cells[3][0] = { 26, true };
cells[3][4] = { 25, true };
cells[3][10] = { 24, true };
cells[3][14] = { 23, true };
```

```
cells[4][0] = { 27, true };
cells[4][1] = { 28, true };
cells[4][2] = { 29, true };
cells[4][3] = { 30, true };
cells[4][4] = { 31, true };
cells[4][5] = { 32, true };
cells[4][6] = { 33, true };
cells[4][7] = { 34, true };
cells[4][8] = { 35, true };
cells[4][9] = { 36, true };
cells[4][10] = { 37, true };
cells[4][11] = { 38, true };
cells[4][12] = { 39, true };
cells[4][13] = { 40, true };
cells[4][14] = { 41, true };
```

```
cells[5][0] = { 45, true };
cells[5][4] = { 44, true };
cells[5][10] = { 43, true };
cells[5][14] = { 42, true };
```

```
cells[6][0] = { 46, true };
cells[6][4] = { 47, true };
cells[6][10] = { 48, true };
cells[6][14] = { 49, true };
```

```
cells[7][0] = { 53, true };
cells[7][4] = { 52, true };
cells[7][10] = { 51, true };
cells[7][14] = { 50, true };
```

```
cells[8][0] = { 54, true };
cells[8][4] = { 55, true };
cells[8][10] = { 56, true };
cells[8][14] = { 57, true };
```

```
cells[9][0] = { 72, true };
cells[9][1] = { 71, true };
cells[9][2] = { 70, true };
cells[9][3] = { 69, true };
cells[9][4] = { 68, true };
cells[9][5] = { 67, true };
```

```

cells[9][6] = { 66, true };
cells[9][7] = { 65, true };
cells[9][8] = { 64, true };
cells[9][9] = { 63, true };
cells[9][10] = { 62, true };
cells[9][11] = { 61, true };
cells[9][12] = { 60, true };
cells[9][13] = { 59, true };
cells[9][14] = { 58, true };

cells[10][0] = { 73, true };
cells[10][4] = { 74, true };
cells[10][10] = { 75, true };
cells[10][14] = { 76, true };

cells[11][0] = { 80, true };
cells[11][4] = { 79, true };
cells[11][10] = { 78, true };
cells[11][14] = { 77, true };

cells[12][0] = { 81, true };
cells[12][4] = { 82, true };
cells[12][10] = { 83, true };
cells[12][14] = { 84, true };

cells[13][0] = { 99, true };
cells[13][1] = { 98, true };
cells[13][2] = { 97, true };
cells[13][3] = { 96, true };
cells[13][4] = { 95, true };
cells[13][5] = { 94, true };
cells[13][6] = { 93, true };
cells[13][7] = { 92, true };
cells[13][8] = { 91, true };
cells[13][9] = { 90, true };
cells[13][10] = { 89, true };
cells[13][11] = { 88, true };
cells[13][12] = { 87, true };
cells[13][13] = { 86, true };
cells[13][14] = { 85, true };

for(int x = 0; x < X_SIZE; x++) {
    for(int y = 0; y < Y_SIZE; y++) {
        if(cells[x][y].pin != -1) {
            leds[cells[x][y].pin] = CHSV( 360, 235, 70);
        }
    }
}

leds[cells[pacman.x][pacman.y].pin] = CHSV( 281, 235, 255);
FastLED.show();
Serial.begin(9600);
}

void loop()
{
    encoder_A = digitalRead(PIN_ENCODER_A);    // Read encoder pins
    encoder_B = digitalRead(PIN_ENCODER_B);

```

```

if(encoder_A > 0) {
    switch(pacman.directionFacing) {
        case 1: case 3:
            goLeft();
            break;
        case 2: case 4:
            goUp();
            break;
    }
} else {
    switch(pacman.directionFacing) {
        case 1: case 3:
            goRight();
            break;
        case 2: case 4:
            goDown();
            break;
    }
}

bool shouldDelay = true;
switch(pacman.directionFacing) {
    case 1:
        shouldDelay = movingUp();
        break;
    case 2:
        shouldDelay = movingRight();
        break;
    case 3:
        shouldDelay = movingDown();
        break;
    case 4:
        shouldDelay = movingLeft();
        break;
}

if(shouldDelay) {
    delay(500);
}
FastLED.show();
}

void goUp() {
    if(cells[pacman.x][pacman.y+1].pin != -1 && pacman.y+1 < Y_SIZE) {
        pacman.directionFacing = 1;
    }
}

void goRight() {
    if(cells[pacman.x+1][pacman.y].pin != -1 && pacman.x+1 < X_SIZE) {
        pacman.directionFacing = 2;
    }
}

void goDown() {

```

```

    if(cells[pacman.x][pacman.y-1].pin != -1 && pacman.y-1 >= 0) {
        pacman.directionFacing = 3;
    }
}

void goLeft() {
    if(cells[pacman.x-1][pacman.y].pin != -1 && pacman.x-1 >= 0) {
        pacman.directionFacing = 4;
    }
}

bool movingUp() {
    if(cells[pacman.x][pacman.y+1].pin != -1 && pacman.y+1 < Y_SIZE) {
        leds[cells[pacman.x][pacman.y].pin] = CHSV( 360, 235, 70);
        leds[cells[pacman.x][pacman.y+1].pin] = CHSV( 281, 235, 255);
        pacman.y += 1;
        return true;
    }
}

bool movingRight() {
    if(cells[pacman.x+1][pacman.y].pin != -1 && pacman.x+1 < X_SIZE) {
        leds[cells[pacman.x][pacman.y].pin] = CHSV( 360, 235, 70);
        leds[cells[pacman.x+1][pacman.y].pin] = CHSV( 281, 235, 255);
        pacman.x += 1;
        return true;
    }
}

bool movingDown() {
    if(cells[pacman.x][pacman.y-1].pin != -1 && pacman.y-1 >= 0) {
        leds[cells[pacman.x][pacman.y].pin] = CHSV( 360, 235, 70);
        leds[cells[pacman.x][pacman.y-1].pin] = CHSV( 281, 235, 255);
        pacman.y -= 1;
        return true;
    }
}

bool movingLeft() {
    if(cells[pacman.x-1][pacman.y].pin != -1 && pacman.x-1 >= 0) {
        leds[cells[pacman.x][pacman.y].pin] = CHSV( 360, 235, 70);
        leds[cells[pacman.x-1][pacman.y].pin] = CHSV( 281, 235, 255);
        pacman.x -= 1;
        return true;
    }
}

```