#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 7, 6, 5, 4);

boolean dinoOnGround = true; //состояние определяющие находится ли ИГРОК на земле или нет

int buttonPin = 8;

int buzzerPin = 9;

int buttonState = 0; //состояние кнопки

int highScore = 0; //переменная максимального счета

boolean playState = false; //состояние игрового процесса

int dist = 0; //начальное состояние переменной dist

int distTwo = 0; //начальное состояние переменной distTwo

int score = 0; //начальное состояние счета ИГРОКа score

//массивы данных dino и tree

//определяющие образ ИГРОКа и препятствия

byte dino [8] //массив символа dino

{

B00000,

B00111,

B00101,

B10111,

B11100,

B11111,

B01101,

B01100,

};

byte tree [8] //массив символа tree

{

B00011,

B11011,

B11011,

B11011,

B11011,

B11111,

B01110,

B01110,

};

//запускаем функцию setup

//определяющие настройки ввода, вывода и

//координаты c-r вывода символа ИГРОКа

void setup()

{

lcd.begin(16, 2); //инициализируем параметры lcd

lcd.createChar(7, dino); //массив данных сохраняем как символ в 7 ячейку с именем dino

lcd.createChar(6, tree); //массив данных сохраняем как символ в 6 ячейку с именем tree

lcd.setCursor(0, 1); //размещаем курсор в c0-r1

lcd.write(7); //выводим символ находящийся в ячейке памяти 7

pinMode(buttonPin, INPUT); //определяем 8 пин как вход

pinMode(buzzerPin, OUTPUT); //определяем 9 пин как выход

}

//запускаем главный цикл loop - ядро программы

//определяющий состояние playState - кнопки запуска игры

void loop()

{

lcd.clear(); //очищаем lcd экран

if (digitalRead(buttonPin) == HIGH) //считываем состояние пина (кнопки)

{ //из п.функции digitalRead и

playState = true; //определяем состояние playState как true

}

if (!playState) //если состояние playState как НЕ true (НЕ ВЕРНО),

{ //то

lcd.setCursor(0, 0); //размещаем курсор в c0-r0 и

lcd.print("Press the button"); //выводим текст "Press the button",

lcd.setCursor(0, 1); //размещаем курсор в c0-r1 и

lcd.print("to start :)"); //выводим текст "to start :)"

}

if (playState) //если состояние playState как true (ВЕРНО),

{ //то

startGame(); //происходит запуск функции startGame()

}

delay(100); //определим задержку ядра программыs

} //точка выхода из loop

//функция описывающая ЦИКЛ игрового процесса

void startGame() {

lcd.clear(); //очищаем lcd экран

dist = random(4, 9); //определяем переменную dist как рандомное число от 4 до 9

distTwo = random(4, 9); //определяем переменную distTwo как рандомное число от 4 до 9

//ЦИКЛ игрового процесса

//интерфейс ЦИКЛа содержит кол-во итерации от 16 до суммы -(dist + distTwo) переменных

//содержащие рандомные числа

for (int i = 16; i >= -(dist + distTwo); i--)

{

//в начале итерации ЦИКЛа выводим состояние СЧЕТа -

lcd.setCursor(13, 0); //размещаем курсор в координатах c13-r0,

lcd.print(score); //выводим счет из переменой score

//score имеет приращение ++ после каждой итерации ЦИКЛа for

//определяем состоянии state

int state = digitalRead(buttonPin);

if (state == HIGH) //если состояние state как HIGH,

{ //то

buttonState = 1; //определяем состояние buttonState как 1 и

tone(buzzerPin, 700, 100); //выводим сигнал на buzzer

}

else if (state == LOW) //если состояние state как LOW,

{ //то

buttonState = 0; //определяем состояние buttonState как 0

}

//согласно состояния state -> buttonState

//определим действия символа ИГРОК - (7) при нажатии на КНОПКУ

if (buttonState == 1) //если состояние buttonState как лог. 1,

{ //то

lcd.setCursor(1, 0); //размещаем курсор в координатах c1-r0 и

lcd.write(7); //выводим символ ИГРОКа,

lcd.setCursor(1, 1); //перемещаем курсор в координатах c1-r1 и

lcd.print(" "); //выводим символ пробела, что бы затереть след символа (7),

dinoOnGround = false; //отмечаем состояние dinoOnGround как false

}

else //иначе,

{ //то

lcd.setCursor(1, 1); //размещаем курсор в координатах c1-r1,

lcd.write(7); //выводим символ ИГРОКа (7)

lcd.setCursor(1, 0); //перемещаем курсор в координаты c1-r0,

lcd.print(" "); //выводим символ пробела, что бы затереть след символа (7)

dinoOnGround = true; //отмечаем, что ИГРОК находится НА ЗЕМЛЕ

}

//выводим препятствия (6)

//используя для движения по координате c итерации i ЦИКЛа

lcd.setCursor(i, 1);

lcd.write(6);

lcd.setCursor(i + 1, 1);

lcd.print("1");

lcd.setCursor(i + dist, 1);

lcd.write(6);

lcd.setCursor(i + dist + 1, 1);

lcd.print("2");

lcd.setCursor(i + dist + distTwo, 1);

lcd.write(6);

lcd.setCursor(i + dist + distTwo + 1, 1);

lcd.print("3");

//опишем условия столкновения,

//вывода текста Game Over на экран по звершении ИГРЫ и вывода максимального счета

//условие 1

if (i == 1 && (dinoOnGround == true)) //если итерация i равна 1 и состояние dinoOnGround как true,

{ //то

lcd.clear(); //вызываем п.функцию lcd.clear для очиски экрана и

lcd.print("Game Over!!!"); //выводим текст Game Over на экран

if (score > highScore) //если счет ИГРОКа score больше счета highScore,

{ //то

highScore = score; //присваиваем значения из score в highScore

}

lcd.setCursor(0, 1); //размещаем курсор в координатах c0-r1,

lcd.print("HS: "); //выводим текст "HS: " и

lcd.print(highScore); //выводим значение из переменной highScore на lcd

playState = false; //изменим состояние playState на false

delay(5000); //выполним задержку кода на 5000 мс

score = 0; //обнулим счёт

break; //выполним выход из ЦИКЛа for

}

//условие 2

else if (i + dist == 1 && (dinoOnGround == true))

{

lcd.clear();

lcd.print("Game Over!!!");

if (score > highScore)

{

highScore = score;

}

lcd.setCursor(0, 1);

lcd.print("HS: ");

lcd.print(highScore);

playState = false;

delay(5000);

score = 0;

break;

}

//условие 3

else if (i + dist + distTwo == 1 && (dinoOnGround == true))

{

lcd.clear();

lcd.print("Game Over!!!");

if (score > highScore)

{

highScore = score;

}

lcd.setCursor(0, 1);

lcd.print("HS: ");

lcd.print(highScore);

playState = false;

delay(5000);

score = 0;

break;

}

//при каждой итерации i ЦИКЛа необходимо

//увеличить счет score на +1

score++; //увеличиваем состояние счета score на +1

delay(300); //задержка ЦИКЛА ИГРЫ

} //точка выхода из ЦИКЛА ИГРЫ

}