```
#include <String.h>

XY oldHeroPos;
XY obstaclePos;

//declare variables
int screenTimerFlag = 0;
int score = 0;

byte hero[8] = {
  B01110,
  B01110,
  B00101,
  B11111,
  B10100,
  B00100,
  B11011,
  B00001,
};

OneMsTaskTimer_t screenTimerTask = {250, screenTimerISR, 0, 0};

void setupRefreshScreen(){
  Serial.begin(9600);
  delay(4800);

  // Set your screen state to its initial state
  screenState = screenInit;

  //Timer Interrupt SetUp
  OneMsTaskTimer::add(&screenTimerTask);
  OneMsTaskTimer::start();
  lcd.createChar(0, hero); // set up hero

}

void loopRefreshScreen(){

  while(screenTimerFlag == 0){
    delay(10);
  }
  screenTimerFlag = 0;
  refreshScreen();
  delay(10);
```

```
}

void drawHero(){
  lcd.setCursor(oldHeroPos.x, oldHeroPos.y);
  lcd.print(" ");
  oldHeroPos = heroPos;
  lcd.setCursor(heroPos.x, heroPos.y);
  lcd.write(byte(0));
}

void eraseHero(){
  lcd.setCursor(oldHeroPos.x, oldHeroPos.y);
  lcd.print(" ");
}

void drawObstacle(){
  lcd.setCursor(obstaclePos.x, obstaclePos.y);
  lcd.print(" ");

  (obstaclePos.x)--;
  if (obstaclePos.x < 0){
    obstaclePos.x = 16;
    //obstaclePos.x = rand () % 17;
    obstaclePos.y = 1;
    //obstaclePos.y = rand() % 2;
  }
  lcd.setCursor(obstaclePos.x, obstaclePos.y);
  lcd.print("|");
}

void refreshScreen(){

  //state transitions
  switch(screenState){
    case screenInit:
      screenState = DrawScreen;
      break;

    case DrawScreen:
      if(obstaclePos.x == heroPos.x && obstaclePos.y == heroPos.y){
        screenState = GameOver;
      }

      else if (obstaclePos.x == heroPos.x && obstaclePos.y != heroPos.y){
        score++;
      }
      break;

    case GameOver:
      screenState = Restart;
      break;


    case Restart:
      if (jump == 1){
        clearScreen();
        screenState = screenInit;
      }
      break;
  }

//actual states
  switch(screenState){
    case screenInit:
    //set initial screen
      score = 0;
      heroPos.x = 0;
      heroPos.y = 1;
      obstaclePos.x = 15;
      obstaclePos.y = 1;

      //title screen for when game is restart
      lcd.setCursor(0, 0);
      lcd.print("   Hero Runner  ");
      lcd.setCursor(0, 1);
      lcd.print("                ");
      delay(1000);
      lcd.setCursor(0, 0);
      lcd.print("       3        ");
      lcd.setCursor(0, 1);
      lcd.print("                ");
      tone(buzzer, 2000, 100);
      delay(1000);
      lcd.setCursor(0, 0);
      lcd.print("       2        ");
      lcd.setCursor(0, 1);
      lcd.print("                ");
      tone(buzzer, 1500, 100);
      delay(1000);
      lcd.setCursor(0, 0);
      lcd.print("       1       ");
      lcd.setCursor(0, 1);
      lcd.print("                ");
      tone(buzzer, 1000, 100);
      delay(1000);
      clearScreen();
      break;

    case DrawScreen:
      drawHero();
      drawObstacle();
      break;

    case GameOver:
      gameOver();
      break;

    case Restart:
      restart();
      break;
  }
}

void gameOver(){
  lcd.setCursor(0, 0);
  lcd.print("   Game Over !  ");
  lcd.setCursor(0, 1);
  //two different lcd.prints to account for single or double variable scores
  if(score < 10){
      lcd.print("    Score: ");
      lcd.print(score);
      lcd.print("    ");
  }
  else{
      lcd.print("   Score: ");
      lcd.print(score);
      lcd.print("    ");
  }

  //buzzer when in gameover
  for(int i = 0; i < 10; i++){
  tone(buzzer, 300, 500);
  delay(500);
  }
}

//displays restart prompt
void restart(){
  lcd.setCursor(0, 0);
  lcd.print("   PRESS JUMP   ");
  lcd.setCursor(0, 1);
  lcd.print("   TO RESTART   ");
}

//clears the screen to make it blank
void clearScreen(){
  lcd.setCursor(0, 0);
  lcd.print("                ");
  lcd.setCursor(0, 1);
  lcd.print("                ");
}

void screenTimerISR(){
  screenTimerFlag = 1;
}
```