```
// include the library code:
#include <LiquidCrystal.h>
#include <OneMsTaskTimer.h>

int points = 0;
int pointsx = 15;

OneMsTaskTimer_t timerTask = {100, playActionTimerISR, 0, 0};

typedef struct xy_struct
{
  int x;
  int y;
}xy;

typedef struct obstical
{
  xy position;
  bool active = 0; //check if the obstical is active
  int type = 0; //type of the obstical
  int checked = 0; //if the obstical has already been checked
} obstical;

obstical bonus; //reusing the obstical struct, the "type" memeber is probably not going to be used
obstical nuke;

int nukeCount = 0;
int count = 0; //incrementer
int delaycnt = 0; //incrementer
int shieldsInUse = 0; //amount of obsticals in use
int maxShields = 2; //max amount of obsticals
int deelaay = 1000; //initial delay between each obstical appearing

byte hero[8] =
{
  B00100,
  B00100,
  B01110,
  B01110,
  B01110,
  B11111,
  B00100,
  B00100,
};

byte sticc[8] =
{
  B11111,
  B10001,
  B10001,
  B10101,
  B10001,
  B10001,
  B11111,
  B00000, //all obsticals are floating since it looks better
};

byte slash[8] =
{
  B11000,
  B00100,
  B00010,
  B00001,
  B00001,
  B00010,
  B00100,
  B11000,
};
```

```
69
70  byte RoundLookingThing[8] =
71  {
72    B01000,
73    B00100,
74    B00100,
75    B01110,
76    B11111,
77    B11111,
78    B11111,
79    B01110,
80  };
81
82  byte special[8] =
83  {
84    B11111,
85    B10101,
86    B10101,
87    B11111,
88    B10101,
89    B10101,
90    B11111,
91    B00000,
92  };
93
94  int const obstcount = 10;
95  obstical obsticals[obstcount]; //a bunch of obsticals
96  obstical oldObsticals[obstcount];
97
98  xy HeroLocation;
99  xy OldLocation;
100 xy sticcLocation = {15,1};
101 xy oldSticcLocation;
102
103 enum PlayerActionStates{GameInit, Gamestart, WaitingForAction, MoveForwards,
    MoveBackwards, MoveUp, MoveDown, gameOver};
104 PlayerActionStates PAS;
105
106 bool PlayerActionFlag = 0;
107 bool BonusThreadFlag = 0;
108 bool NukeThreadFlag = 0;
109 bool ScreenThreadFlag = 0;
110
111 // initialize the library with the numbers of the interface pins
112 LiquidCrystal lcd(P6_7, P2_3, P2_6, P2_4, P5_6, P6_6);
113
114 void setup()
115 {
116   lcd.begin(16, 2);
117   lcd.createChar(0, hero);
118   lcd.createChar(1, sticc);
119   lcd.createChar(2, special);
120   lcd.createChar(3, RoundLookingThing);
121   lcd.createChar(4, slash);
122   Serial.begin(9600);
123
124   OneMsTaskTimer::add(&timerTask);
125   OneMsTaskTimer::start();
126
127   for(int i=0; i<obstcount; i++)
128   {
129     obsticals[i].position.x = 16;
130     obsticals[i].active = 0;
131     obsticals[i].type = 0;
132     obsticals[i].checked = 0;
133   }
134
135   for(int i=0; i<(int)(obstcount/2); i++) //half of the obsticals will be on top
136   {
```

```
137       obsticals[i].position.y = 0;
138     }
139
140     for(int i=(int)(obstcount/2); i<obstcount; i++) //half of the obsticals will be on
        the bottom, screw random numbers
141     {
142       obsticals[i].position.y = 1;
143     }
144
145     for(int i=0; i<(int)(obstcount/2); i=i+2) //half of the obsticals on top will be of
        type 0
146     {
147       obsticals[i].type = 0;
148     }
149
150     for(int i=(int)(obstcount/2); i<obstcount; i=i+2) //half of the obsticals on top will
        be of type 1
151     {
152       obsticals[i].type = 1;
153     }
154
155     obsticals[0].type = 0;
156
157   }
158
159   void loop() {
160     delay(100);
161     //Serial.print("(LCD Wokring)");
162   }
163
164   void eraseShield(int i)
165   {
166     lcd.setCursor(oldObsticals[i].position.x, oldObsticals[i].position.y); //these code
        are given, just added an input parameter to help navigation
167     lcd.print(" ");
168   }
169
170   void playActionTimerISR()
171   {
172     PlayerActionFlag = 1;
173     BonusThreadFlag = 1;
174     NukeThreadFlag = 1;
175     ScreenThreadFlag = 1;
176   }
177
```