Pinball Template Code

```
#include <Pinball.h>

#include "arrays.h"

//_________________________________________PIN DEFINITIONS, FLAGS_______

int spkr_pin = 13;
Pb_speaker spkr(spkr_pin);                 // Speaker
Pb_outputs shregs(10, 12, 11, 2);    // Shift registers
// (data, clk, latch, number of registers)
Pb_scoreboard myboard(8, 9);         // Scoreboard (clock, data)

byte serdata[2];                     // For the shift registers
// serdata[1,0] are each 8 independent LEDs

int ir_pin = A0, piezo_pin = A1;    // IR, Piezo pins
int roll_pin = A3, drain_pin = 7;  // roller, drain switch pins

// Switches for roll and drain
Pb_switch roll_sw(50), drain_sw(50);

// Flags for the same
int roll_flag, drain_flag;

// Game specific global variables
int ii, num_lives = 4, score = 0, score_flag = 0;
int ir_thresh = 800, piezo_thresh = 500;
int ir_val, piezo_val, ir_delay, piezo_delay = 1000;
int ir_flag = 0, piezo_flag = 0;

// Timed events
Pb_timedevent LEDflash(flash);
Pb_timedevent scoreflash(flashscore);

// Stopwatch for ir and piezo debounce
Pb_stopwatch mywatch, mywatch_ir, mywatch_piezo;


//__________________________________UPDATE FUNCTION_____

void update_music_and_events() {

  spkr.update();
  LEDflash.update();
  scoreflash.update();

}
```

```
//_____________________________________________SETUP___________

void setup() {
  // put your setup code here, to run once:

  pinMode(roll_pin, INPUT); pinMode(drain_pin, INPUT);
  // Enable pullup resistors on digital input pins
  digitalWrite(roll_pin, HIGH); digitalWrite(drain_pin, HIGH);

    serdata[0] = 0b11111111; // blue LEDs
    serdata[1] = 0b00000000; // red LEDs

  shregs.update(serdata);
  delay(500);

  spkr.loopstart(beep_vals, beep_time, beep_len);

  myboard.setpartition(1); // Use scoreboard to keep track of lives
  myboard.predisplay(num_lives);
  myboard.postdisplay(score);
  delay(250);
  LEDflash.loopstart(flashloop, flashtime, 2);
  spkr.start(startup_vals, startup_time, startup_len);
  LEDflash.start(startup_vals, startup_time, startup_len);

}

//____________________________________________THE LOOP___________

void loop() {
  // put your main code here, to run repeatedly:

  if (num_lives > 0) {
    readinputs();
    dologic();
    writeoutputs();
  }

  update_music_and_events();

}



//____________________________________________INPUTS___________

void readinputs() {

  roll_flag = 0; drain_flag = 0;

  roll_flag = roll_sw.pushed(digitalRead(roll_pin));
  drain_flag = drain_sw.pushed(digitalRead(drain_pin));

  ir_val = analogRead(ir_pin);
  piezo_val = analogRead(piezo_pin);

}
```

```
//____________________________________________LOGIC___________

void dologic() {

  score_flag = 0; // Used to decide whether to update scoreboard

  if (roll_flag == 1) { score = score + 1; score_flag = 1; }

  if (ir_val > ir_thresh) {
    if (ir_flag == 0) {
      score = score + 5; score_flag = 2;
      ir_flag = 1;
      mywatch_ir.start();
    }
  } else if (ir_flag > 0) {
    if (mywatch_ir.time() > ir_delay) {
      ir_flag = 0;
      mywatch_ir.stop();
    }
  }

  if (piezo_val > piezo_thresh) {
    if (piezo_flag == 0) {
      score = score + 5; score_flag = 3;
      piezo_flag = 1;
      mywatch_piezo.start();
    }
  } else if (piezo_flag > 0) {
    if (mywatch_piezo.time() > piezo_delay) {
      piezo_flag = 0;
      mywatch_piezo.stop();
    }
  }


  if (drain_flag == 1) { num_lives = num_lives - 1; score_flag = 4;}

}
```

```
//____________________________________________OUTPUTS___________

void writeoutputs() {

  int shreg_flag = 0;

  switch (score_flag) {
    case 1:
      spkr.start(coin_vals, coin_time, 3);
      break;
    case 2:
      spkr.start(coin_vals, coin_time, 15);
      break;
    case 3:
      spkr.start(oneup_vals, oneup_time, oneup_len);
      break;
    // You can add more cases
  }

  if (drain_flag == 1) {
    shreg_flag = 1;
    spkr.start(life_vals, life_time, life_len);
    if (num_lives > 0) {
      LEDflash.start(lifeflash, lifetime, 20);
    } else {
      LEDflash.loopstop();
      LEDflash.start(deathLED, deathtime, 17);
      scoreflash.loopstart(scflashvals, scflashtime,2);
      spkr.loopstop();
      spkr.start(death_vals, death_time, death_len);
    }
  }

  if (roll_flag > 0) {
    LEDflash.start(shiftpatvals, shiftpattime, 17);
    spkr.start(scoreone_vals, scoreone_time, scoreone_len);
  }

  myboard.predisplay(num_lives);
  myboard.postdisplay(score);

  if (shreg_flag > 0) { shregs.update(serdata); }
  if (score_flag > 0) {
    myboard.predisplay(num_lives);
    myboard.postdisplay(score);
  }

}
```

```
//___________________________________________SPECIAL FUNCTIONS___________


void flash(int val) {
  // Flash the LEDs
    if (serdata[0] == 0b00000000) { serdata[0] = 0b11111111; }
    else { serdata[0] = 0b00000000; }
    if (serdata[1] == 0b00000000) { serdata[1] = 0b11111111; }
    else { serdata[1] = 0b00000000; }

  shregs.update(serdata);

}


void flashscore(int val) {
 // Flash the scoreboard

  if (val == 1) {
    myboard.blankpredisplay();
    myboard.blankpostdisplay();
  }
  else {
    myboard.predisplay(num_lives);
    myboard.postdisplay(score);
  }

}
```