```
#include <OneMsTaskTimer.h>

int xOutPin = P4_2;
int yOutPin = P5_5;
int xVal;
int yVal;

int selectPin = PUSH1;

int xFlag = 0;
int yFlag = 0;
bool jumpFlag = 0;

int cnt = 0;

void setupPlayerActions()
{
  pinMode(selectPin, INPUT_PULLUP);
  Serial.begin(9600);
  HeroLocation.x = 0;
  HeroLocation.y = 1;

  attachInterrupt(digitalPinToInterrupt(selectPin), jumpISR, FALLING);
}

void loopPlayerActions()
{

  while(PlayerActionFlag == 0)
  {
    delay(10);
  }
  PlayerActionFlag = 0;

  //Serial.print("(PA Thread Wokring) ");


  /*Serial.println(jumpFlag);
  delay(1000);
  Serial.println(jumpFlag);*/

  if(PAS!=gameOver && PAS!=Gamestart && PAS!=GameInit) // detect player movements only
  if the game is in progress
  {
    xVal = analogRead(xOutPin);
    setXflag(xVal);

    yVal = analogRead(yOutPin);
    setYflag(yVal);
  }

  /*Serial.println(xVal);*/
  //Serial.println(xFlag);

  //Serial.println(yVal);
  //Serial.println(yFlag);

  PlayerStateProgress();

  /*Serial.print("Hero position: ");
  Serial.print(HeroLocation.x);
  Serial.print(" ");
  Serial.println(HeroLocation.y);*/
  /*Serial.print("Current Game State: ");
  Serial.println(PAS);*/

  delay(10);

}
```

```

void jumpISR()
{
  Serial.println("ISR - Jump");
  jumpFlag = 1;
}

void setXflag(int xVal)
{
  if(xVal > 730 && xVal < 830)
  {
    xFlag = 0;
  }
  else if(xVal>900)
  {
    xFlag = 1;
  }
  else if(xVal<533)
  {
    xFlag = -1;
  }
}


void setYflag(int yVal)
{
  if(yVal > 730 && yVal < 830)
  {
    yFlag = 0;
  }
  else if(yVal>900)
  {
    yFlag = 1;
  }
  else if(yVal<533)
  {
    yFlag = -1;
  }
}

void PlayerStateProgress()
{
  //switch statements
  switch(PAS)
  {
    case(GameInit):
      PAS = Gamestart;
      break;
    case(Gamestart):
      if(jumpFlag)
      {
        points = 0;
        jumpFlag = 0;
        lcd.clear();
        PAS = WaitingForAction;
      }
      break;
    case(WaitingForAction):
      //Serial.println("Transition Waiting for Action");
      if(xFlag == 1 && HeroLocation.x <= 15)
      {
        PAS = MoveForwards;
        //Serial.println("Moving Forwards");
        break;
      }
      else if(xFlag == -1 && HeroLocation.x >= 0)
      {
        PAS = MoveBackwards;
        break;
      }
      else if(yFlag == -1 && HeroLocation.y == 0)
      {
        Serial.println("Moving Down");
        PAS = MoveDown;
        break;
      }
      else if(yFlag == 1 && HeroLocation.y == 1)
      {
        Serial.println("Moving Up");
        PAS = MoveUp;
        break;
      }
      else
      {
        PAS = WaitingForAction;
        break;
      }
    case(MoveForwards):
        PAS = WaitingForAction;
        break;
    case(MoveBackwards):
        PAS = WaitingForAction;
        break;
    case(MoveDown):
        PAS = WaitingForAction;
        break;
    case(MoveUp):
        PAS = WaitingForAction;
        break;
    default:
        PAS = WaitingForAction;
        break;
    case(gameOver):

        //Serial.println("Game Over");

        if(jumpFlag)
        {
          Serial.println(jumpFlag);
          jumpFlag = 0;
          lcd.clear();
          PAS = Gamestart;
        }
        break;
  }

  //state actions
  switch(PAS)
  {
    case(Gamestart):
      lcd.setCursor(5, 0);
      lcd.print("READY?");
      lcd.setCursor(1, 1);
      lcd.print("press to start");
      HeroLocation.x=0;
      HeroLocation.y=1;
      break;

    case(MoveForwards):
      //Serial.println("Move Forward State");
      xFlag=0;
      (HeroLocation.x) = (HeroLocation.x) + 1;
      break;

    case(MoveBackwards):
      //Serial.println("Move Backward State");
      xFlag=0;
      (HeroLocation.x)--;
      break;

    case(MoveUp):
      yFlag = 0;
      (HeroLocation.y) = 0;
      break;

    case(MoveDown):
      yFlag = 0;
      (HeroLocation.y) = 1;
      break;

    case(gameOver):
      for(int i=0; i<obstcount; i++)
      {
        obsticals[i].position.x = 16; //reset every obstical on game over
        obsticals[i].active = 0;
        obsticals[i].checked = 0;
      }

      bonus.position.x = 16; //also reset everything else
      bonus.position.y = 0;
      bonus.active = 0;
      nukeCount = 0;
      count = 0;
      shieldsInUse = 0;
      maxShields = 2;
      deelaay = 1000;

      //pointsx = 15; //also reset points
      points = 0;
      break;
  }

}
```