```
#include <LiquidCrystal.h>
#include <OneMsTaskTimer.h>

int p = 0;

xy OldBonusPosition;

void setupBonus()
{
  bonus.position.x = 16;
  bonus.position.y = 0;
  OldBonusPosition = bonus.position;
  Serial.begin(9600);
}

void loopBonus()
{

  while(BonusThreadFlag == 0)
  {
    delay(10);
  }
  BonusThreadFlag = 0; //bonus is on a faster clock compared to the obsticals, means it will move a lot faster than the obsticals
  //Serial.print("(Bonus Thread Wokring) ");

  if(PAS!=gameOver && PAS!=Gamestart && PAS!=GameInit) // if the game is not over and not at the starting screen
  {
    srand(millis());
    p = (rand()%200);

    if(PAS!=gameOver && PAS!=Gamestart)
    {
      if(count >= 10000)  //designed to create a bonus every 10 seconds (on average);
      {
        //Serial.print("Creating bonus");

        createBonus();
        count = 0;

        //Serial.print("Finish Creating Bonus");
      }
    }

    if(bonus.active == 1)
    {
      advanceBonus(); //move the bonus across the screen
      deleteBonus(); //deleting the bonus when it is off the screen
    }

    if(HeroLocation.x==bonus.position.x && HeroLocation.y==bonus.position.y) //collision
    {
      //Serial.print("Bonus Collected");

      nukeCount++;  //inrement bonus count
      points = points + 2; //bonus also awards points, two of them
      delaycnt = delaycnt + 2; //since it awards point, it should also reduce the delay

      lcd.setCursor((HeroLocation.x+1), HeroLocation.y); //indicate that the bonus has been collected
      lcd.print(nukeCount); //this is fine, I wanted the nuke count of flash instead of permenant but this is fine
      delay(100);
      lcd.print(" ");

      eraseBonus();
      bonus.active = 0; //make the bonus inactive
      bonus.position.x = 16; //reset position
      srand(millis());
      bonus.position.y = rand()%2; //random the y position of the bonus, between 0 and 1, for some reason it feels really broken
    }

    count = count + p;
  }
}

void createBonus()
{
  if(bonus.active == 0) //only draws when the bonus isn't already on the screen, can also check for active in the main loop, but here is fine
  {
    bonus.active = 1;
    lcd.setCursor(bonus.position.x, bonus.position.y);
    lcd.write(byte(3));
  }
}

void advanceBonus()
{
  eraseBonus();
  drawBonus();
}

void eraseBonus()
{
  lcd.setCursor(OldBonusPosition.x, OldBonusPosition.y); //these code are given
  lcd.print(" ");
}

void drawBonus()
{
    bonus.position.x = bonus.position.x - 1;
    lcd.setCursor(bonus.position.x, bonus.position.y);
    lcd.write(byte(3));
    OldBonusPosition = bonus.position;
}

void deleteBonus()
{
  if(bonus.position.x < -1) //-1 to prevent the object disappearing before making collision
    {
      eraseBonus();

      bonus.active = 0; //make the bonus inactive
      bonus.position.x = 16; //reset position

      srand(millis());
      bonus.position.y = rand()%2; //random the y position of the bonus, between 0 and 1, for some reason it feels really broken

      Serial.print("Bonus Y: ");
      Serial.println(bonus.position.y);
    }
}
```