Part 3: Secure Programming

For Part 3, I have chosen Asteroid game from my graphics programming assignment. The function of this game is to make the spaceship attack the incoming asteroids by releasing a series of bullets. The game ends in 3 different scenarios:

1) When the asteroids collide with the Spaceship

2) When the asteroids collide with the Earth

3) When the spaceship collides with the Earth.

Secure Programming Recommendation 1:

One of the secure programming recommendations is to create unclonable classes. Using Java's object cloning technique an attacker could create a class function without having to call any of its constructors. For asteroid game, I have defined a few classes that are cloneable. One of such classes is the class BulletSystem.

```
class BulletSystem {

  constructor(){
    this.bullets = [];
    this.velocity = new createVector(0, -5);
    this.diam = 10;
  }

  run(){
      this.move();
      this.draw();
      this.edges();
  }

  fire(x, y){
    this.bullets.push(createVector(x,y));
  }

  //draws all bullets
  draw(){
    fill(255);
    for (var i=0; i<this.bullets.length; i++){
      ellipse(this.bullets[i].x, this.bullets[i].y, this.diam, this.diam);
    }
  }

  //updates the location of all bullets
  move(){
    for (var i=0; i<this.bullets.length; i++){
      this.bullets[i].y += this.velocity.y;
    }
  }

  //check if bullets leave the screen and remove them from the array
  edges(){
      // YOUR CODE HERE (3 lines approx)
      for (var i=this.bullets.length-1; i>=0; i--){
          if(this.bullets[i].y<0){
              this.bullets.splice(i,1);
          }
      }
  }
}
```

To make my class unclonable, I'm going to add a new method in each of my classes.

public final Object clone () throws java. lang. CloneNotSupportedException {

throw new java. lang. CloneNotSupportedException ();}

I feel implementing the above code in my classes will make my project more secure.

Secure Programming Recommendation 2:

Another recommendation that I have to make my code more secure is to make my classes undeserializeable.

An attacker can create a series of bytes that deserialize to a class instance with the attacker's preferred values. Deserialization is a form of a public constructor which lets an attacker pick the object's state. The impact of insecure deserialization can be very severe. However, project does not have any codes that can prevent deserialization.

```
class Spaceship {

  constructor(){
    this.velocity = new createVector(0, 0);
    this.location = new createVector(width/2, height/2);
    this.acceleration = new createVector(0, 0);
    this.maxVelocity = 5;
    this.bulletSys = new BulletSystem();
    this.size = 50;
  }
```

In order to prevent deserialization, I implemented the following codes:

private final void readObject (ObjectInputStream in) throws java.io.IOException {

throw new java.io. IOException ("Class cannot be deserialized");}

I feel implementing the above code in my classes will make my project more secure.

Secure Programming Recommendation 3: