

Name: _____

Boids II Worksheet

Exercise 1a

In order to have 100 prey boids, which of the animation variables in `starter_advboids` will become array variables?

In this case, the neighborhood radius and angle and the maximum acceleration and speed will be species-specific (shared by all boids of the same type) rather than different for individual boids. Consider prey and predator to be different species.

```
// prey
PVector preypos, preyvel;           // position and velocity
float preyhunger;                   // hunger
float preyradius, preyangle;        // defines what the prey can see
float preymaxforce, preymaxspeed;   // other prey parameters

// predator
PVector predpos, predvel;           // position and velocity
float predradius, predangle;        // defines what the predator can see
float predmaxforce, predmaxspeed;   // other predator parameters

// food
float food;                          // amount of food left
PVector foodpos;                     // food position
```

Draw a box around the following chunks of code below: (some comments have been omitted to save space; see the provided code)

- the initialization of each animation variable that becomes an array variable
- drawing the prey boid
- updating the animation variables for the prey boid

When considering what goes into each box, note the steps 2-4 are all part of updating the boid and logically should be kept together. Also note that the local variable `steer` is involved in both step 2 and step 3 so that those steps must be kept together.

```
void setup () {
  size(800, 800);

  preypos = new PVector(random(0, width), random(0, height));
  preyvel = new PVector(random(-1, 1), random(-1, 1));
  predpos = new PVector(random(0, width), random(0, height));
  predvel = new PVector(random(-1, 1), random(-1, 1));

  preyhunger = random(100, 1000);

  preyradius = 60;
  preyangle = radians(135);
  preymaxforce = .3;
  preymaxspeed = 3;
  predradius = 300;
  predangle = radians(135);
  predmaxforce = .1;
  predmaxspeed = 2;

  food = 255;
  foodpos = new PVector(random(50, width-50), random(50, height-50));
}
```

(continued on the next page)

```

void draw () {

    // 1 - draw scene
    background(255);
    // food
    fill(255-food);
    ellipse(foodpos.x, foodpos.y, 20, 20);
    // prey boid
    drawBoid(preypos, preyvel, 255, 255-preyhunger, 255-preyhunger);
    // predator boid
    drawBoid(predpos, predvel, 0, 0, 255);

    // -- update prey -----

    if ( preyhunger < 1000 &&
        dist(foodpos.x, foodpos.y, preypos.x, preypos.y) < 10 ) {
        preyhunger = preyhunger+100;
        food = food-.5;
    } else {
        preyhunger = preyhunger-1;
    }
    // update prey's position and velocity
    {
        // 2 - compute net steering force
        // a - compute steering force for each behavior

        // b - combine forces (one behavior)
        PVector steer = new PVector(0, 0);

        // c - limit the size of the force that can be applied
        steer.limit(preymaxforce);

        // 3 - update boid's velocity
        // a - add net steering force
        preyvel.add(steer);
        // b - limit boid's max speed
        preyvel.limit(preymaxspeed);

        // 4 - update boid's position
        // a - update position by adding velocity
        preypos.add(preyvel);
        // b - wrap at edges of window
        if ( preypos.x > width ) {
            preypos.x = 0;
        } else if ( preypos.x < 0 ) {
            preypos.x = width;
        }
        if ( preypos.y > height ) {
            preypos.y = 0;
        } else if ( preypos.y < 0 ) {
            preypos.y = height;
        }
    }
}

```

(continued on the next page)

```

// -- update predator
-----
// update predator's position and velocity
{
    // 2 - compute net steering force
    // a - compute steering force for each behavior

    // b - combine forces (one behavior)
    PVector steer = new PVector(0, 0);

    // c - limit the size of the force that can be applied
    steer.limit(predmaxforce);

    // 3 - update boid's velocity
    // a - add net steering force
    predvel.add(steer);
    // b - limit boid's max speed
    predvel.limit(predmaxspeed);

    // 4 - update boid's position
    // a - update position by adding velocity
    predpos.add(predvel);
    // b - wrap at edges of window
    if ( predpos.x > width ) {
        predpos.x = 0;
    } else if ( predpos.x < 0 ) {
        predpos.x = width;
    }
    if ( predpos.y > height ) {
        predpos.y = 0;
    } else if ( predpos.y < 0 ) {
        predpos.y = height;
    }
}

// -- update food -----
// move food if depleted
if ( food < 0 ) {
    food = 255;
    foodpos = new PVector(random(50, width-50), random(50, height-50));
}
}

```

Exercise 1b

The rules for how each prey boid moves are given in the handout:

- If the prey boid is within 250 pixels of the predator, it should evade the predator and flock with the other prey boids. (This means four behaviors are active in this step — *evade, separation, alignment, and cohesion.*)
- Otherwise if the prey is not near the food (more than 100 pixels away) and there's potential for colliding with another prey boid, avoid the collision.
- Otherwise if the prey boid is hungry (hunger is 0 or lower) and is within 300 pixels of the food, it should arrive at the food.
- Otherwise the prey boid should wander.

This is already written in language that closely matches the structure of an `if` statement. Identify how you are going to translate each condition into code – write the specific condition (in Processing) that you will use in your sketch.

the prey boid is within 250 pixels of the predator	
prey boid is more than 100 pixels from the food	
prey boid is hungry	

Exercise 1c

The rules for how each predator boid moves are given in the handout:

- If there's prey in sight, the predator should pursue it.
- Otherwise the predator should wander.

This is already written in language that closely matches the structure of an `if` statement. Identify how you are going to translate each condition into code – write the specific condition (in Processing) that you will use in your sketch.

there's prey in sight	
-----------------------	--