

BLOCKLY GAMES: POND BATTLE

Pt. 1 - ANALYSIS OF COMPETITORS:

Describe in plain English the strategy of **the red duck, "Rook."**

(In other words, when and how does it move? How does it locate enemies?)

Rook moves to the center of the y axis and scans north, south, east, and west. When Rook locates enemies, it shuffles between due east and due west to center on the target. It shoots the at the distance of the target.

What are the advantages and disadvantages of this strategy?

The advantages are the fact that anyone following a specific path rotation will get hit by Rook and the simplicity of the tactic. The disadvantages are the fact that there is little to no defence and the fixed scanners can be easily outmaneuvered.

Describe in plain English the strategy of **the green duck, "Counter."**

(In other words, when and how does it move? How does it locate enemies?)

Counter moves at the degree it last scanned a target and moves in a random direction when hit. Counter locates enemies by doing a small cone sweep for maximized accuracy. When it finds a target, it shoots at the distance of the target.

What are the advantages and disadvantages of this strategy?

The advantages are that Counter is very unpredictable, it is not dead-set on a location, and it is very accurate. The disadvantages are the lack of defence and the tendency to go to the middle.

Describe in plain English the strategy of **the black duck, "Sniper."**

(In other words, when and how does it move? How does it locate enemies?)

Sniper moves to a corner and darts between corners when it is shot. While it is in the corner, Sniper does a 90 degree sweep of the corner it is in and locates a target. If it doesn't locate one, it moves to another corner. Once it locates an enemy, it scans a smaller area and keeps scanning and firing until hit.

What are the advantages and disadvantages of this strategy?

The advantages are the extreme defence tactic, the quick darting between corners, and the unpatterned corner movement. The disadvantages are the easily located areas of movement and the lack of an offensive tactic while on the move.

Pt. 2 - DEVELOPING YOUR OWN TACTICS:

Consider the advantages and disadvantages of your opponents' strategies. Identify what you think are the most effective tactics and then combine them with your own original and creative ideas for both offense *and* defense. Describe in plain English what you want your duck to do. Please be as specific as possible.

(For example: *"My duck will scan 360 degrees around and stop scanning when it locates an enemy in range. It will keep firing until the enemy is no longer in range. Then it will swim in that direction and beginning scanning again. If my duck gets hit, it will reverse directions at full speed and swim to the nearest corner before beginning scanning again."*)

I want my duck to quickly scan when moving from close to a corner to close to a corner and stop at each close to a corner. Depending on the health of my duck, I want to scan more areas when I stop the lesser my health is.

Pt. 3 - PSEUDOCODE:

Pseudocode is halfway between speaking English and actual programming. It uses the logic and structure of programming without all the grammar rules. In other words, it conveys the idea of how the program will be organized without needing to worry about semicolons and parentheses, etc.

For example:

My Spinning Scanner Function

Make a variable named "angle" = 0

Loop this forever:

 Scan using my angle variable

 If the scanner says a enemy is in range

Then fire the cannon at that angle and that range
otherwise,
increase the angle variable by one

Please write a pseudocode description for some of your most important tactics -- at least one offense and one defensive tactic.

Pseudocode for Important OFFENSIVE Tactic:

```
stop  
Detect function  
While x < 50 and y < 50  
swim(45 degrees, 100 speed)  
Quickscan function
```

Pseudocode for Important DEFENSIVE Tactic:

```
stop  
Detect function  
While x < 90 and y < 90  
swim(45 degrees, 100 speed)  
Quickscan function
```

Pt. 4: MY POND CODE (Prototype)

Computer programming is a type of Engineering and follows the Engineering Design cycle. At this stage, you've already developed some possible solutions to your problem (i.e., strategies to survive and win), and it's time to develop a working prototype. Copy and paste an early working draft of your code here. (It does not need to be perfect because you're going to continue to reiterate to improve upon it.)

```
while (true) {  
  while (loc_x() < 50 && loc_y() > 50) {  
    swim(315, 100);  
    quickscan();  
  }  
  stop();  
  detect();  
  while (loc_x() < 90 && loc_y() < 90) {  
    swim(45, 100);  
    quickscan();  
  }  
}
```

```

stop();
detect();
while (loc_x() > 50 && loc_y() > 50) {
    swim(225, 100);
    quickscan();
}
stop();
detect();
while (loc_x() < 90 && loc_y() > 10) {
    swim(315, 100);
    quickscan();
}
stop();
detect();
while (loc_x() > 50 && loc_y() < 50) {
    swim(135, 100);
    quickscan();
}
stop();
detect();
while (loc_x() > 10 && loc_y() > 10) {
    swim(225, 100);
    quickscan();
}
stop();
detect();
while (loc_x() < 50 && loc_y() < 50) {
    swim(45, 100);
    quickscan();
}
stop();
detect();
while (loc_x() > 10 && loc_y() < 90) {
    swim(135, 100);
    quickscan();
}
stop();
detect();
}
function quickscan() {
    for (var i = 0; i < 16; i++) {
        if (scan(i * 20, 20) <= 70) cannon(i * 20, scan(i * 20, 20));
    }
}
}
function detect() {
    quickscan();
}

```

```
}
```

Pt. 5: MY POND CODE (Optimized)

After your initial rough draft, you kept testing it to make small changes and improvements. The modifications to your code were meant to optimize it to make it as efficient and effective as possible. Paste your final optimized draft here. (This is the final version of the code you're submitting for competition and for a grade.)

```
while (true) {
  while (loc_x() < 50 && loc_y() > 50) {
    swim(315, 100);
    quickscan();
  }
  stop();
  detect();
  while (loc_x() < 90 && loc_y() < 90) {
    swim(45, 100);
    quickscan();
  }
  stop();
  detect();
  while (loc_x() > 50 && loc_y() > 50) {
    swim(225, 100);
    quickscan();
  }
  stop();
  detect();
  while (loc_x() < 90 && loc_y() > 10) {
    swim(315, 100);
    quickscan();
  }
  stop();
  detect();
  while (loc_x() > 50 && loc_y() < 50) {
    swim(135, 100);
    quickscan();
  }
  stop();
  detect();
  while (loc_x() > 10 && loc_y() > 10) {
    swim(225, 100);
    quickscan();
  }
}
```

```

}
stop();
detect();
while (loc_x() < 50 && loc_y() < 50) {
    swim(45, 100);
    quickscan();
}
stop();
detect();
while (loc_x() > 10 && loc_y() < 90) {
    swim(135, 100);
    quickscan();
}
stop();
detect();
}
function quickscan() {
    for (var i = 0; i < 16; i++) {
        if (scan(i * 20, 20) <= 70) cannon(i * 20, scan(i * 20, 20));
    }
}
function sweep() {
    for (var i = 0; i < 72; i++) {
        if (scan(i * 5) <= 70) cannon(i * 5, scan(i * 5));
    }
}
function dualSweep() {
    for (var i = 0; i < 72; i++) {
        if (scan(i * 5) <= 70) cannon(i * 5, scan(i * 5));
        if (scan(i * 5 + 180) <= 70) cannon(i * 5 + 180, scan(i * 5 +
180));
    }
}
function fullOnAttack() {
    for (var i = 0; i < 72; i++) {
        if (scan(i * 5) <= 70) cannon(i * 5, scan(i * 5));
        if (scan(i * 5 + 90) <= 70) cannon(i * 5 + 90, scan(i * 5 + 90));
        if (scan(i * 5 + 180) <= 70) cannon(i * 5 + 180, scan(i * 5 +
180));
        if (scan(i * 5 + 270) <= 70) cannon(i * 5 + 270, scan(i * 5 +
270));
    }
}
function detect() {
    if (health() >= 75) quickscan();
    else if (health() >= 50) sweep();
    else if (health() >= 25) dualSweep();
}

```

```
else fullOnAttack();  
}
```

Pt. 6: SELF-EVALUATION

After you have optimized your code and competed against classmates in the tournament bracket, evaluate the effectiveness and efficiency of your own code. What tactics or aspects of the program do you feel were relatively strong or particularly effective? In what ways did your program fall short? How would you like to modify or improve upon your program in the future?

The effectiveness of my code was good against Rook and Counter, due to their lack of defensive tactic. However, I was not very effective against Sniper, due to my inability to get around him and inability to accurately hit him. I was very efficient, keeping the code simple, but the movements could probably be condensed. My strong point was my scanning tactics. I scanned more as my health dropped, so the generality of my scans became more accurate as I needed it. I did have a lack of accuracy while on the run, resulting in a few bad hits from everyone right off the bat. I would like to implement a better scan system when sitting still, as well as lengthening the time I sit still to about 5 seconds.