**Orca Predation Algorithm**

****

**Analysis of Algorithm**

**BSDS 4-1**

**Semester Project**

**Supervised By**

**Mr. Muhammad Usman Sharif**

**Faculty of Computing**

**Riphah International University**

[**1. Introduction**](#) **3**

[**2. Objective**](#) **3**

[**3. Tools and Technologies**](#) **3**

[**4. Key Components**](#) **3**

[4.1 Player (User-Controlled Orca)](#) 3

[4.2 Orcas (Autonomous Agents)](#) 4

[4.3 Fish (Collectibles)](#) 4

[**5. Orca Predation Algorithm (OPA) Integration**](#) **5**

[OPA Steps:](#) 5

[**6. Game Flow**](#) **6**

[**7.Visual Elements**](#) **6**

[**8.Game Controls**](#) **6**

[**9. Features and Enhancements**](#) **6**

[Implemented:](#) 6

[Possible Enhancements:](#) 7

[**10. Conclusion**](#) **7**

[**11. Assets**](#) **7**

[**12. References**](#) **7**

[**13. GitHub**](#) **7**

Orca Predation Implementation

Code Documentation

# **1. Introduction**

This project is a 2D simulation game developed using Python and Pygame, which visually demonstrates the behavior of orcas using the Orca Predation Algorithm (OPA**)**. The player controls an orca avatar to collect fish while being pursued by other orcas whose movements are guided by OPA principles.

# **2. Objective**

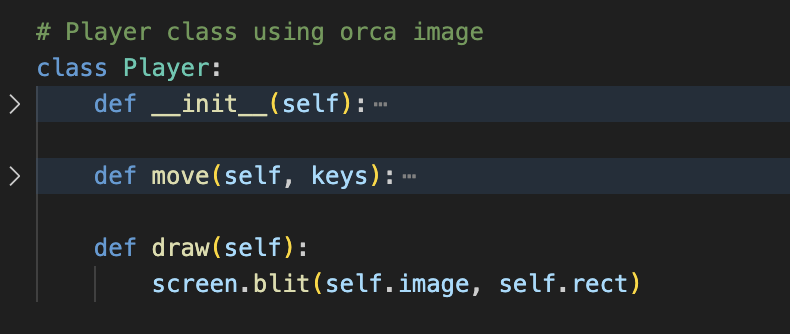
* To create a simple interactive simulation where the behavior of autonomous orcas is governed by a nature-inspired optimization strategy.
* To visually represent the working of the Orca Predation Algorithm (OPA**)** within a game setting.

# **3. Tools and Technologies**

| **Tool** | **Description** |
| --- | --- |
| Python | Programming language |
| Pygame | Game development library |
| Math & Random | Used for vector calculations and randomness |
| **4. Key Components** |  |

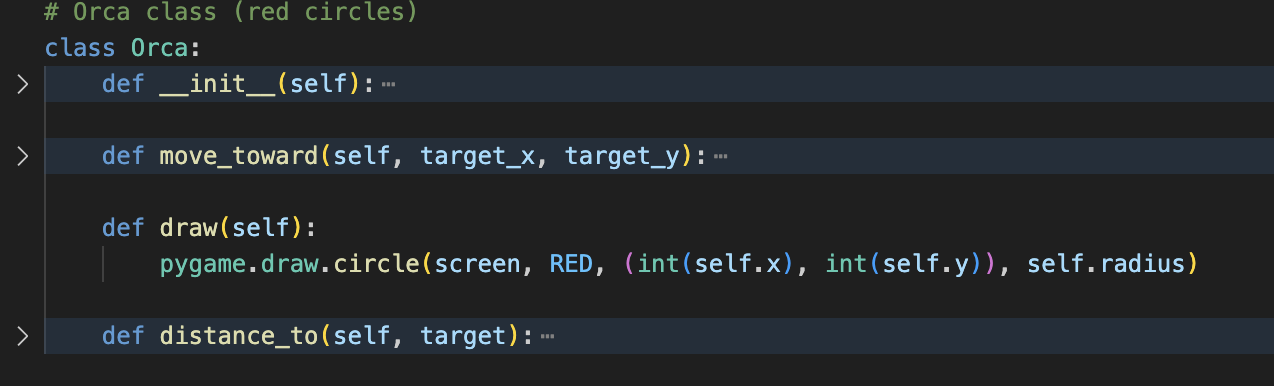
## **4.1 Player (User-Controlled Orca)**

* Controlled via arrow keys.
* Prevented from moving outside the screen.
* Tasked with collecting randomly placed fish.
* Gains score by collecting fish.



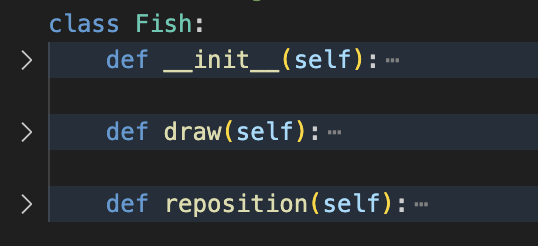
## **4.2 Orcas (Autonomous Agents)**

* Represent enemy orcas.
* Move toward the player using OPA logic.
* Visualized as red circles.



## **4.3 Fish (Collectibles)**

* Green circular objects randomly positioned on screen.
* Reappear at a new position after being collected.

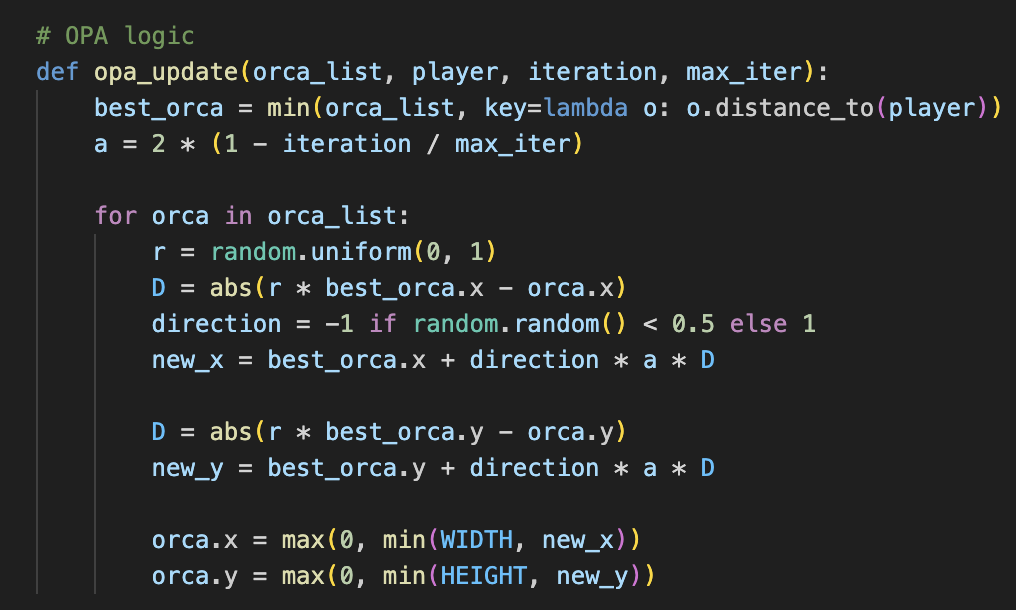


# **5. Orca Predation Algorithm (OPA) Integration**

OPA is integrated into the game loop via the opa\_update() function. This function updates the position of each orca by simulating natural predator behavior:

## **OPA Steps:**

1. Identify best orca – closest to the player.
2. Coefficient aa decreases over time to balance exploration and exploitation.
3. Each orca updates its position toward or away from the best orca using a directionally randomized vector.



This mimics:

* **Hunting toward prey** (exploitation).
* **Random deviation** to explore other options (exploration).

# **6. Game Flow**

1. **Initialize** game assets and variables.
2. **Player movement** is captured and handled.
3. **OPA updates** orca positions each frame.
4. **Collision check** between player and fish.
5. **Score is updated** and displayed.
6. **All elements are drawn** onto the screen.
7. **Loop continues** until the user exits.

# **7.Visual Elements**

| **Element** | **Appearance** | **Color / Asset** |
| --- | --- | --- |
| Player | Orca image | Orca.png scaled |
| Orcas | Red circles | RED |
| Fish | Green circles | GREEN |
| Score | Displayed at top-left | Text in BLACK |

# **8.Game Controls**

| **Key** | **Action** |
| --- | --- |
| Arrow Keys | Move player orca |
| Close Button | Quit game |

# **9. Features and Enhancements**

## **Implemented:**

* Smooth orca image rendering.
* Scalable number of enemy orcas.
* Dynamic fish repositioning.
* Score tracking system.

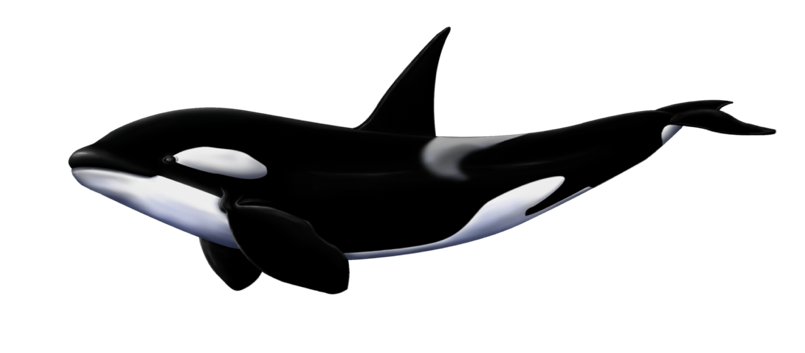
## **Possible Enhancements:**

* Add game timer or health system.
* Sound effects and background music.
* Levels or increasing difficulty.
* Enhanced graphics for fish and orcas.

# **10. Conclusion**

This game provides an interactive and visual demonstration of the Orca Predation Algorithm (OPA). It showcases how a biologically inspired optimization technique can be creatively applied in a simulated environment. By combining Pygame and OPA, the project not only engages users in gameplay but also provides educational value regarding swarm intelligence.

# **11. Assets**



# **12. References**

* Orca Predation Algorithm Theory.
* Pygame Documentation: <https://www.pygame.org/docs/>
* Swarm Intelligence Research Papers.

# **13. GitHub**

**Link:** [**https://github.com/Kaleem205/AoA-Project.git**](https://github.com/Kaleem205/AoA-Project.git)

**Submitted By**

Muhammad Kaleem (56614)