Race Sorter – QuickSort Visualization Game

Algorithm Racing & Sorting Visualization Project Report

By:
Daniyal Asghar

SAP:
53731

Instructor:
Mr. Muhammad Usman Sharif

Department of Faculty of Computing
RIPHAH INTERNATIONAL UNIVERSITY,

ISLAMABAD, PAKISTAN

Contents:

1. Introduction

Race Sorter is an interactive visualization game developed using Python and Pygame. In this game, each racer car's attributes (Speed, Acceleration, Handling) are combined to calculate a sort value, which is used by the QuickSort algorithm to determine the racer order.

The purpose of the game is to help students:

- Visually understand sorting algorithms
- Observe the effect of pivot selection
- Learn the difference between Randomized and Deterministic QuickSort
- Visualize algorithm steps in a racing context

This project combines racing with algorithm learning, making sorting intuitive and engaging.

---

2. Problem Statement

Students often find it difficult to understand QuickSort concepts like pivot selection, partitioning, lane swapping, and randomized vs deterministic behavior in theoretical form.

Visual representation of sorting steps can serve as an effective teaching method.

Race Sorter addresses this by mapping:

- Racers = array elements
- Lanes = array indexes
- Lane swaps = array swaps
- Median racer boost = algorithmic insight
- Race progress = sorting effect

---

3. Objectives

The main objectives are:

- Demonstrate QuickSort algorithm visually
- Explain Randomized vs Deterministic pivot selection

- Integrate racing mechanics with sorting
- Provide median racer a temporary boost for insight
- Highlight sorting operations graphically
- Develop an educational and entertaining interface

---

4. Project Scope

The project focuses on educational and interactive visualization:

- Comparison of sorting algorithms
- Pivot visualization
- Real-time racer movement
- Median element power boost
- Performance stats (sort operations, finish times)

Future extensions include:

- MergeSort visualization
- HeapSort races
- Larger arrays
- Multiplayer mode

---

5. Tools & Technologies Used

Tool / Library	Purpose
Python 3.x	Core programming
Pygame	Graphics, animation, rendering
Enum	Algorithm mode definition
Random	Random pivot selection, attributes
Time Module	Finish time measurement
Math Module	Minor calculations

---

6. Algorithms Used

6.1 Randomized QuickSort

- Pivot is randomly chosen.
- Provides better average-case performance.
- Pivot is highlighted with a yellow border.

Steps:

1. Random pivot selection
2. Swap with low index
3. Partition subarray
4. Recursively sort subarrays

6.2 Deterministic QuickSort

- Pivot is always the last element.
- Worst-case $O(n^2)$ is possible.
- Predictable execution.

Steps:

1. Choose last element as pivot
2. Partition subarray
3. Recursively sort subarrays

6.3 Median Finding

The median racer is identified using order statistics:

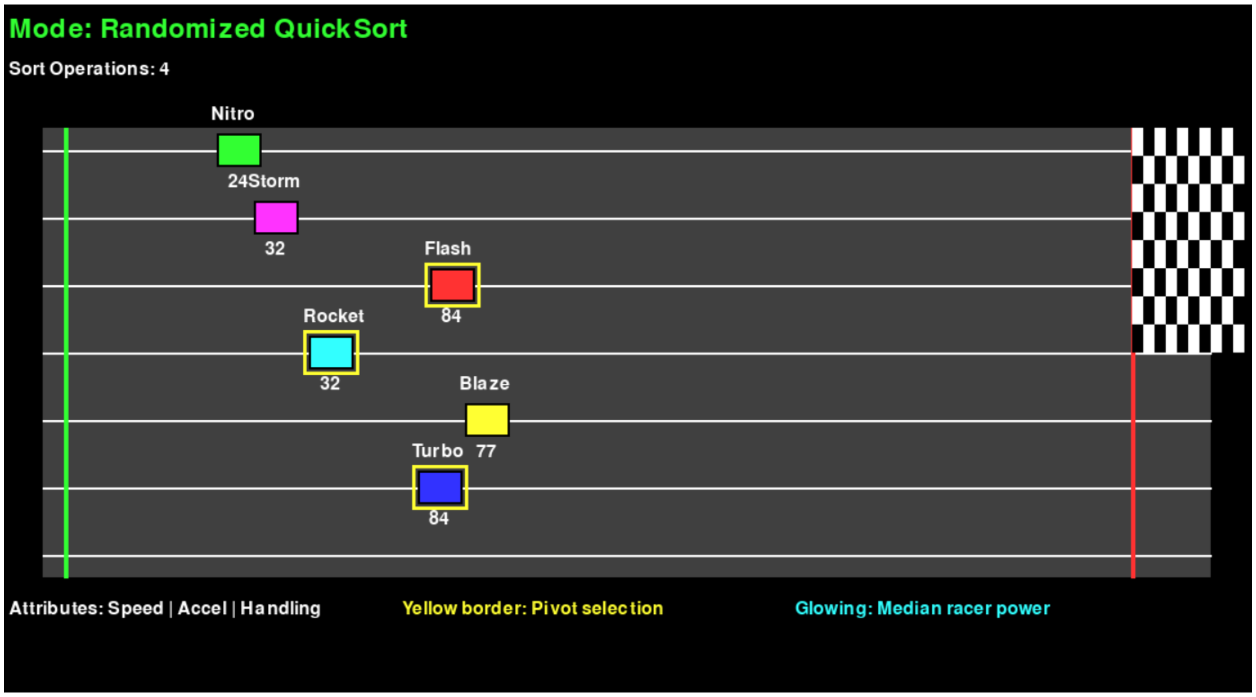
median = sorted(racers by sort_value)[n // 2]

Median racer gets a 3-second power boost, demonstrating the concept of median-of-three pivot selection.

7. Time Complexity Analysis

Algorithm	Best	Average	Worst
Randomized QuickSort	O(n log n)	O(n log n)	$O(n^2)$
Deterministic QuickSort	O(n log n)	O(n log n)	$O(n^2)$
Partition Function	O(n)	O(n)	O(n)

8. Game Design



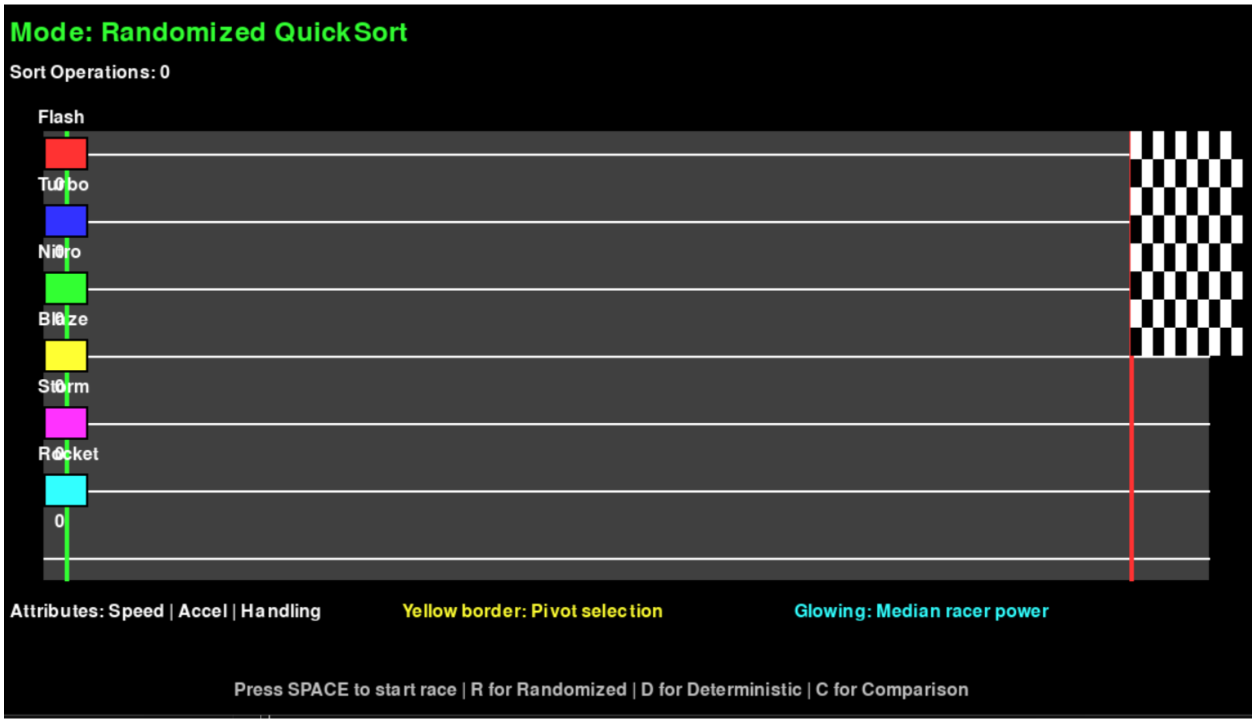
Mode: Randomized QuickSort

Sort Operations: 4

Attributes: Speed | Accel | Handling

Yellow border: Pivot selection

Glowing: Median racer power


Mode: Randomized QuickSort

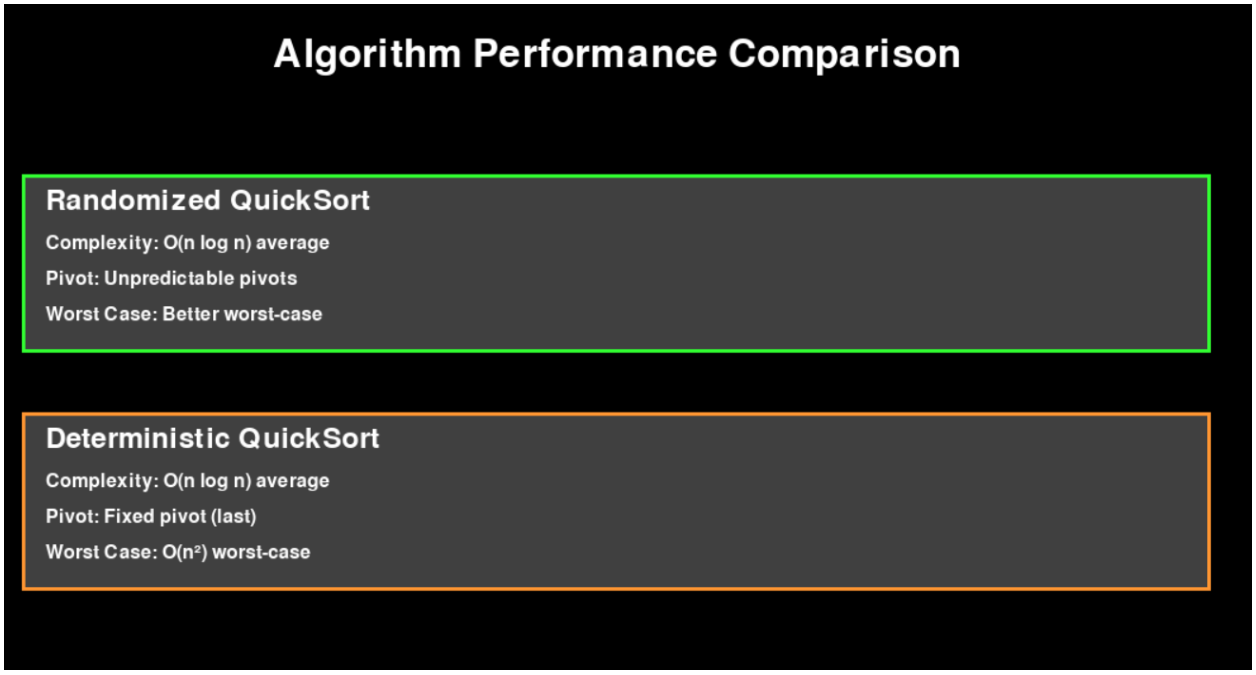
Sort Operations: 0

Attributes: Speed | Accel | Handling

Yellow border: Pivot selection

Glowing: Median racer power

Press SPACE to start race | R for Randomized | D for Deterministic | C for Comparison


Algorithm Performance Comparison

Randomized QuickSort

Complexity: O(n log n) average

Pivot: Unpredictable pivots

Worst Case: Better worst-case

Deterministic QuickSort

Complexity: O(n log n) average

Pivot: Fixed pivot (last)

Worst Case: $O(n^2)$ worst-case

8.1 User Interface Design

UI elements include:

- Track with 6 racing lanes
- Start line, finish line, checkered flag
- Racer names and stats (Speed, Acceleration, Handling)
- Sort operation counter
- Side legend
- Pivot highlight box
- Finish order leaderboard

8.2 Racing Mechanics

- Racers accelerate based on their attributes
- Median racer gets temporary turbo boost
- Handling introduces slight randomness
- Racers stop at finish line
- Finish times are recorded

8.3 Sorting Visualization

Sorting affects:

- Racer lane positions
- Pivot highlights
- Swap animations via lane switches
- Randomized vs Deterministic pivot behavior

---

9. Implementation Details

9.1 Code Structure

Component	Description
Racer class	Stores attributes, movement logic, sort value
RaceSorterGame class	Main controller
quicksort_randomized()	Random pivot QuickSort
quicksort_deterministic()	Last-element pivot QuickSort
partition()	Lane swapping visualization
draw_racers()	Draw cars with pivot highlight
update()	Racer movement and boost logic

9.2 Key Functions

calculate_sort_value() – Combines racer attributes for sorting.

partition() – Performs array swap, visually represented as lane swap.

activate_power() – Gives median racer temporary boost.

start_race() – Applies sorting and initiates race.

draw_comparison() – Shows performance comparison of Randomized vs Deterministic QuickSort.

---

10. Real-World Applications

- Teaching sorting algorithms
- Visual algorithm education tools
- AI/ML data ordering systems
- Animation-based learning modules
- Game-based learning

11. Challenges & Solutions

Challenge	Solution
Pivot visualization	Yellow highlight border
Lane swapping clarity	Animated lane assignments
Attribute balancing	Weighted formula for sort values
Algorithm comparison	Dedicated comparison screen
Median racer insight	Special power mechanism

12. Conclusion

Race Sorter is an engaging educational project that:

- Demonstrates QuickSort visually and intuitively
- Explains Randomized vs Deterministic pivot behavior
- Uses race mechanics to showcase sorting steps interactively

It can be extended into a multi-algorithm educational platform for teaching purposes.

GITHUB LINK