Introduction to Artificial Intelligence-CSL304-Group Project Report

Team members:

Ranga Chandra Naga Venkata Chaitanya Kumar

Dadepogu Rithvika

Gurrala Hansika Reddy

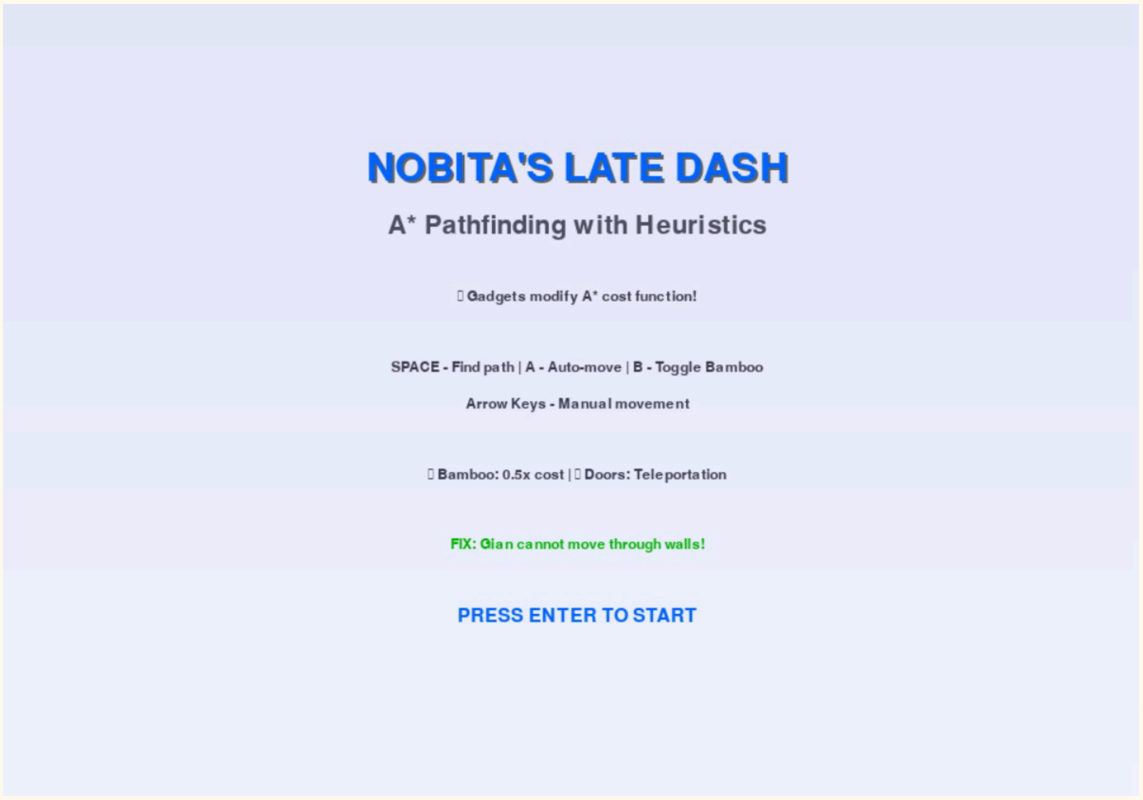
Balthi Reethika

Nobita’s Late Dash – A* Pathfinding with Heuristics

Core idea:implemetation of a path finding game using the assistance A* algorithm

- ***Abstract: Nobita's Race Against the Clock***

The A* pathfinding algorithm was a key component in *Nobita’s Late Dash*, a Pygame-based game. In this game, Nobita must reach his school using the least number of steps. He is equipped with two gadgets— the Bamboo Copter and the Anywhere Door. The Bamboo Copter reduces movement costs while navigating the A* grid, and the Anywhere Door enables instant teleportation across the map. Experimental results show that these tools significantly lower the overall path cost and help the player move more efficiently. Along with a basic enemy added to increase difficulty, the game offers a real-time visualization of A* in action, effectively demonstrating how pathfinding works.



NOBITA'S LATE DASH

A* Pathfinding with Heuristics

 Gadgets modify A* cost function!

SPACE - Find path | A - Auto-move | B - Toggle Bamboo

Arrow Keys - Manual movement

 Bamboo: 0.5x cost | Doors: Teleportation

FIX: Gian cannot move through walls!

PRESS ENTER TO START

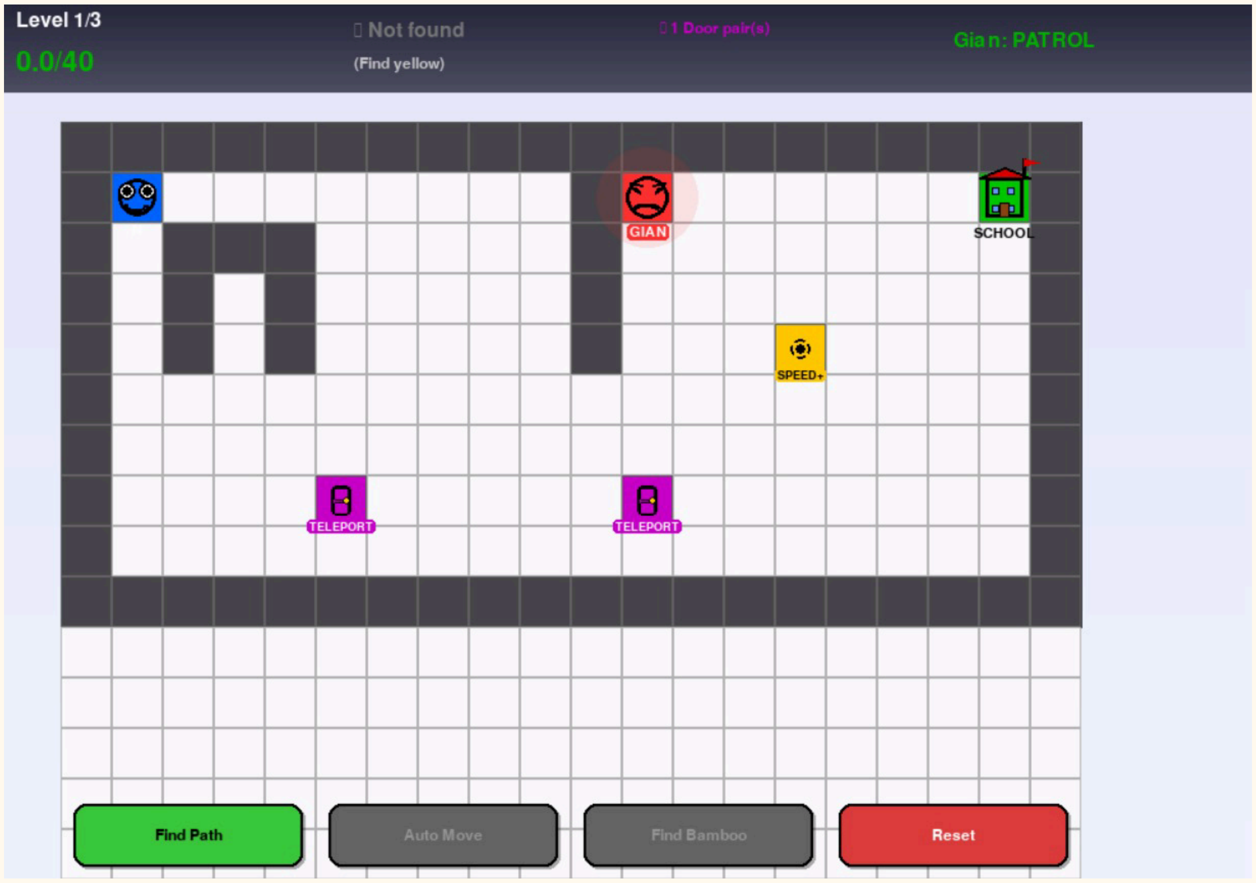
1. Introduction: Pathfinding – The Engine of Game AI

- Pathfinding acts as the decision-making system behind character movement in games, determining the most efficient route between two points. Although Breadth-First Search (BFS) guarantees the shortest path, it becomes slow in larger maps because it explores every possible option. A* improves on this by combining the cost already covered, g(n), with an estimated cost to the destination, h(n). By using both values, A* focuses on the most promising paths first, making it much faster and more efficient than BFS.
- In the game, the player (Nobita) must move through a maze with a strict limit on the number of steps, pushing them to choose the most efficient path. The unique technical feature is that Nobita can collect items that dynamically modify how the A* algorithm behaves while the game is running, instantly altering the pathfinding logic.

Project Objectives

- To build a robust and modular implementation of the A* algorithm.

- To integrate game mechanics (gadgets) as drivers for dynamic heuristic and graph structural changes.
- To introduce a simple enemy AI to create necessary time and spatial constraints.
- To provide an accessible, real-time visualization of the A* search process.
- To use a Star Rating metric for the objective assessment of path efficiency.



2. Methodology: Implementing A* and Dynamic Modification

2.1 The Core A* Algorithm

The maze is represented as a graph where each grid cell acts as a node, and every node begins with a default movement cost of 1.0. A* evaluates each node nnn using the formula:

f(n)=g(n)+h(n)Here, g(n) is the accumulated cost from the starting position to node nnn, while h(n) is the heuristic estimate of the remaining cost to reach the goal.

For the heuristic, we used Manhattan distance, which measures the minimum number of horizontal and vertical steps required on the grid. A priority queue is then used to always select and expand the node with the smallest f(n), ensuring efficient path exploration.

2.2 Gadgets: Dynamic Algorithmic Modifiers

The gadgets function as in-game tools that manipulate the variables A* uses for its calculation.

Bamboo Copter – Cost Reduction (Modifying (g(n)))

This item, once collected, cuts the movement cost for all future cells from 1.0 down to 0.5. This directly reduces the (g(n)) component for every subsequent step.

Hypothesis: Halving the per-step cost will lead to a proportional reduction in the total computed path cost.

Anywhere Door – Topology Modification (Modifying the Graph)

This gadget introduces a teleportation link between two far-apart locations, effectively adding a special edge with a fixed cost of 1.0. As a result, the maze's graph structure is altered by providing new, low-cost neighboring nodes during A*'s search.

Hypothesis: Adding such shortcuts that skip large portions of the maze should allow A* to find significantly shorter and more efficient overall paths.

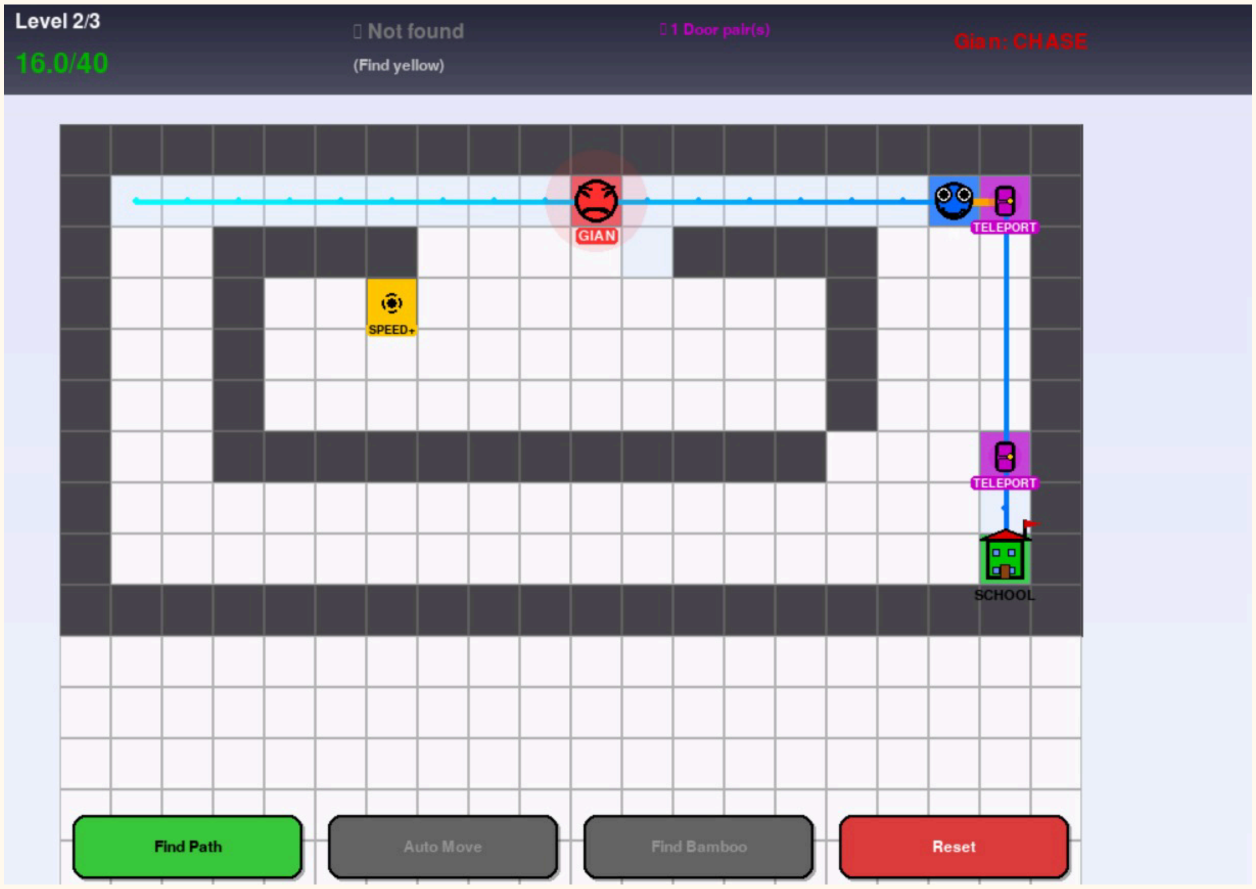
2.3 Enemy AI: Gian's Threat

Gian, the enemy AI, creates real-time pressure on the player's path execution. His behavior uses a simple state machine:

Patrol: Gian follows a preset path when Nobita is outside his detection range.

Chase: When Nobita enters an 8-cell radius, Gian pursues him using wall-aware movement.

This design ensures that optimal paths must be both fast and reliable.



3. Results: Objective Path Performance

3.1 Scoring Metrics

- To evaluate path efficiency across different scenarios, we track three key parameters for each level:
- **Move Limit:** The maximum path cost (g(n)) allowed before failure.
- **Optimal Moves:** The total cost required to achieve a 3-Star rating.
- **Actual Path Cost:** The total (g(n)) accumulated by Nobita to reach the school.

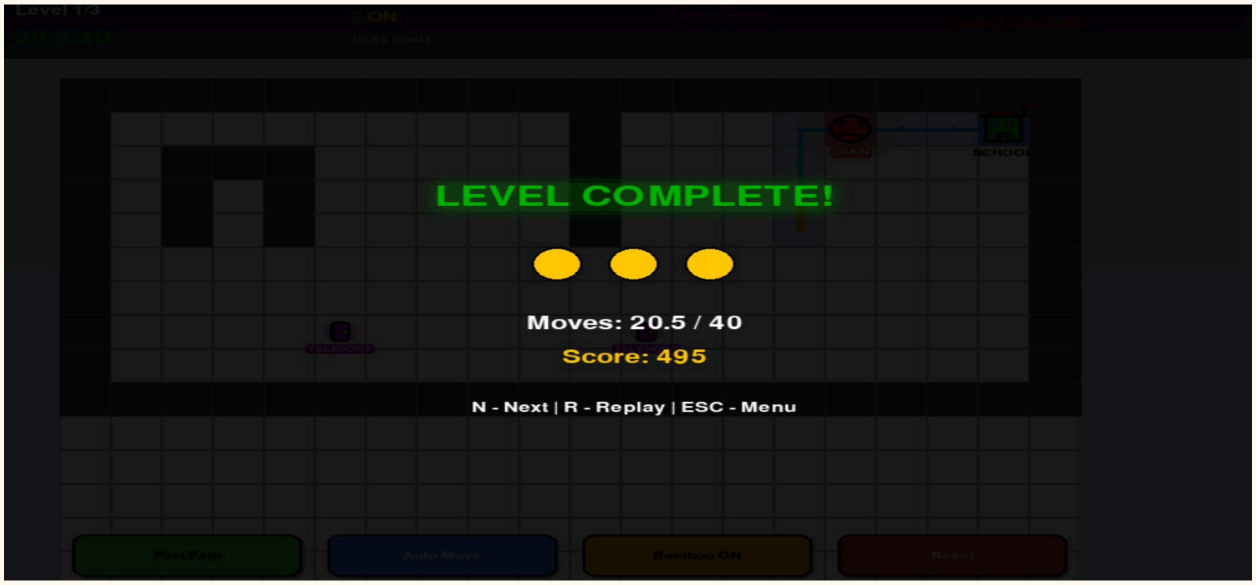
- Star Rating:
- ★★★ if cost ≤{Optimal}
- ★★ if cost ≤{Optimal} + 5
- ★ if cost ≤{Limit}

<u>3.2 Bamboo Copter Results</u>

Condition	**Cost (g(n))**	**Stars**
Copter OFF	20.0	★★★
Copter ON	10.0	★★★

  ○

- **Result:**
  The Bamboo Copter effectively cuts the total path cost in half without changing the actual distance traveled. This clearly demonstrates how modifying the cost function within A* can significantly influence the algorithm's evaluation of the optimal path.


LEVEL COMPLETE!

Moves: 20.5 / 40

Score: 495

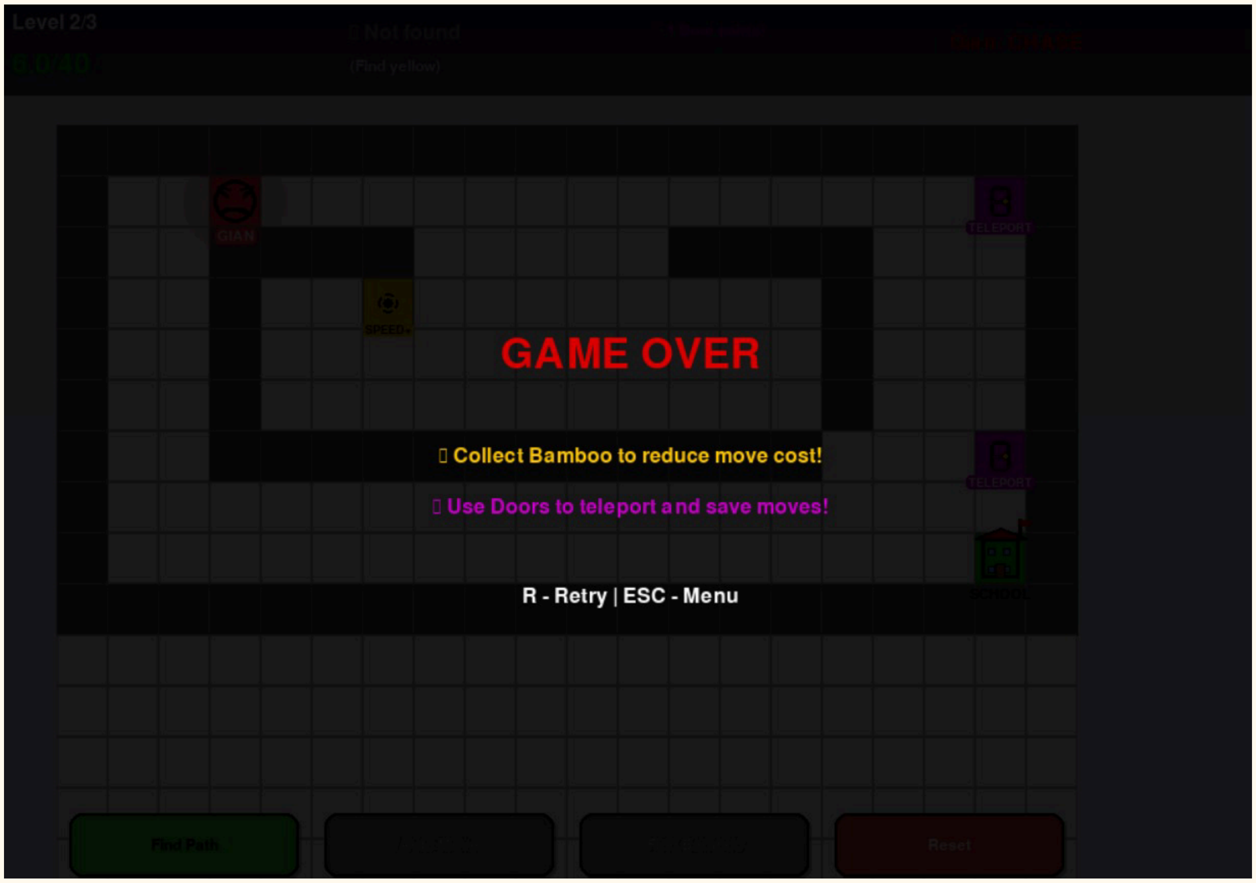
N - Next | R - Replay | ESC - Menu

<u>3.3 Anywhere Door Results</u>

Without the Anywhere Door:
A* is forced to navigate long detours around walls and narrow corridors, resulting in a higher overall cost.

With the Anywhere Door:
A single teleportation edge instantly connects two distant points, bypassing large sections of the maze. This dramatically reduces the total traversal cost.

Conclusion:
By altering the maze's topology, the Anywhere Door enables A* to discover shortcuts that would never be available through cost reduction alone. This highlights the powerful impact of structural graph changes on pathfinding efficiency.



GAME OVER

Collect Bamboo to reduce move cost!

Use Doors to teleport and save moves!

R - Retry | ESC - Menu

3.4 <u>Visualization and Performance</u>

- The real-time visualization of A* makes its behavior easy to observe:

- The algorithm expands far fewer nodes than BFS due to its guided heuristic.
- Optimal paths on 20×20 grids are computed almost instantly.
- The Manhattan heuristic produces clear, goal-directed search patterns that visually illustrate how A* prioritizes its explorations.

4. Conclusion

4.1 Summary

Nobita's Late Dash demonstrates how the A* algorithm can be significantly improved through dynamic heuristic modifiers and structural shortcuts.

The **Bamboo Copter** reduces movement cost, making every step more efficient.

The **Anywhere Door** changes the graph itself, offering entirely new shortcuts.

Together, these mechanics lead to more optimal paths and better performance scores. The game as a whole provides an intuitive, interactive environment for understanding how pathfinding algorithms behave and how they can be enhanced through cost and topology modifications.