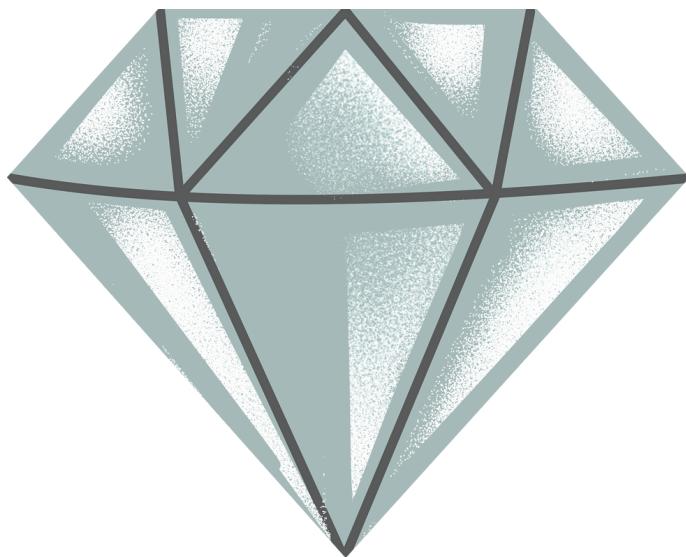


Crystal Capture

Entertainment Game



Supervised by :

Dr:Sara Sweidan

ENG:Yousef Tarek

Our Team

01 Marwa Kotb Saad sec3

02 Nourhan Ahmed Mohamed sec3

03 Eman Yousef Darweish sec1

04 Ayat Mohamed Galal sec1

05 Mohanad Mostafa sec3

Introduction:

Crystal Capture is a simple 2D game developed using Python and the Tkinter library. The game features a graphical interface with animated movements, image-based characters, and sound effects to enhance the user experience. Crystal Capture relies on event-driven programming, canvas-based rendering, and basic game logic to manage player interaction, animation, and competition within the game environment which is (Fully_Observable-discrete-deterministic-static Sequential-Multi-Agent)

Objectives

01

Navigate the player character smoothly across the game grid.

02

Interact with game elements such as crystals and obstacles.

03

Compete against an AI-controlled robot within the same environment

04

Demonstrate basic concepts of game development using Python and Tkinter, including graphics, animation, and event handling.

05

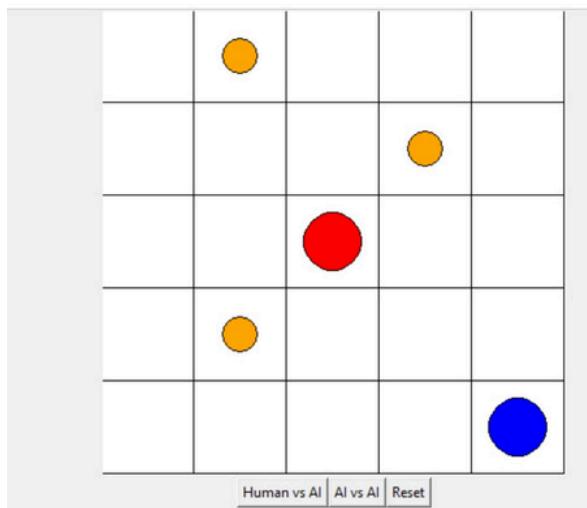
Provide an engaging and interactive 2D gameplay experience

Methodology:

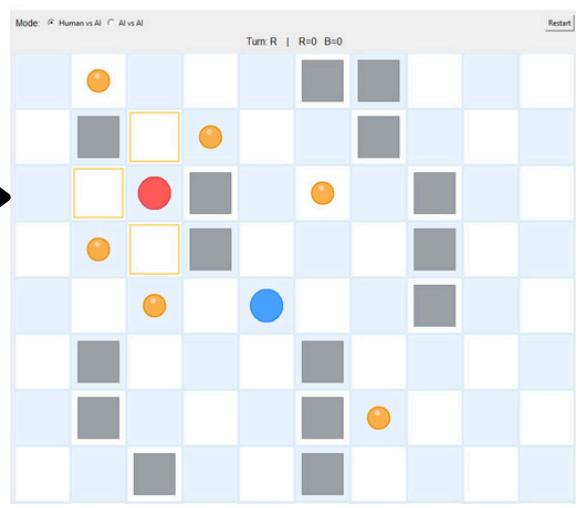
The development of Crystal Capture followed an iterative and problem-driven methodology to gradually improve gameplay, intelligence, and overall game dynamics

O1

Initially, the game board was designed to be relatively small to simplify testing and validate the core mechanics. After ensuring basic functionality, the board size was increased to provide a richer and more challenging environment. During this stage, the density of obstacles and crystals was adjusted to balance difficulty and movement freedom.



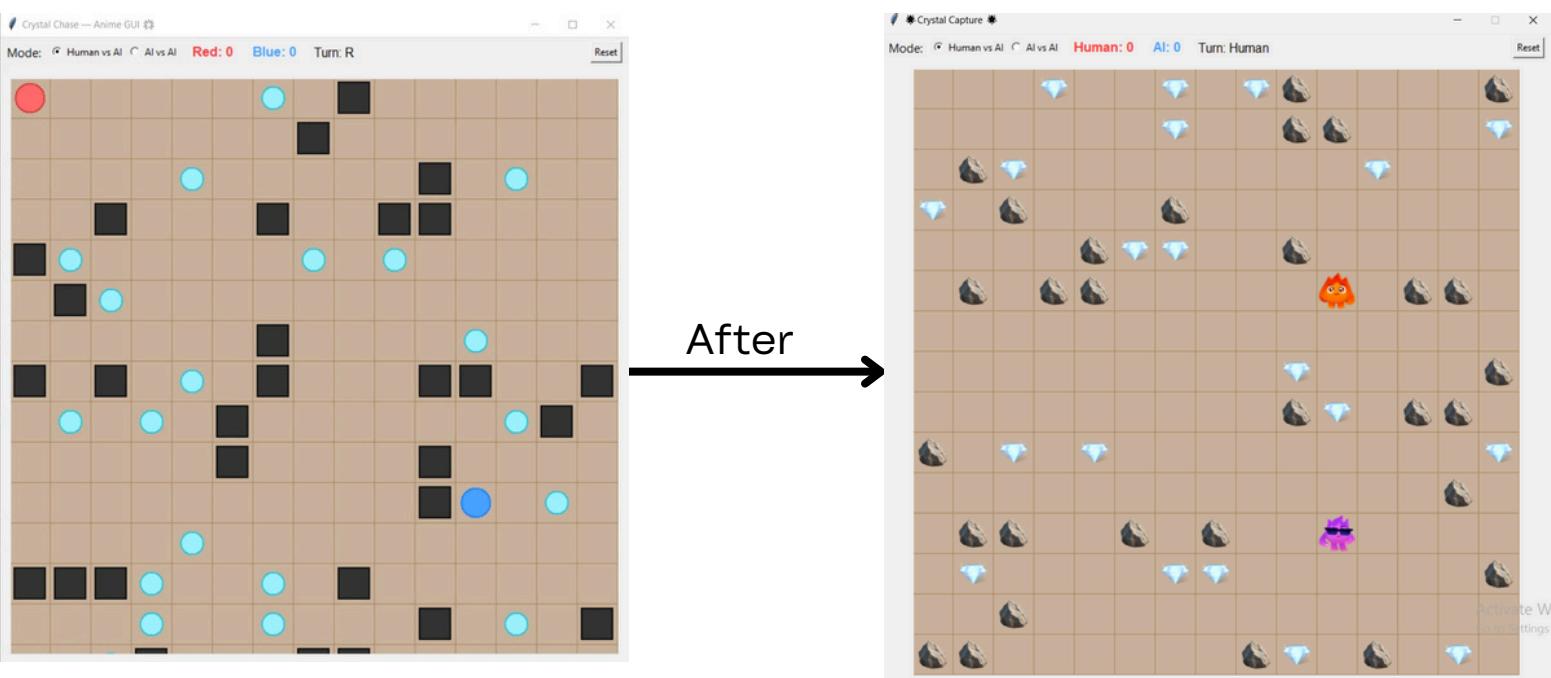
After



Methodology:

Graphical User Interface (GUI) Enhancements

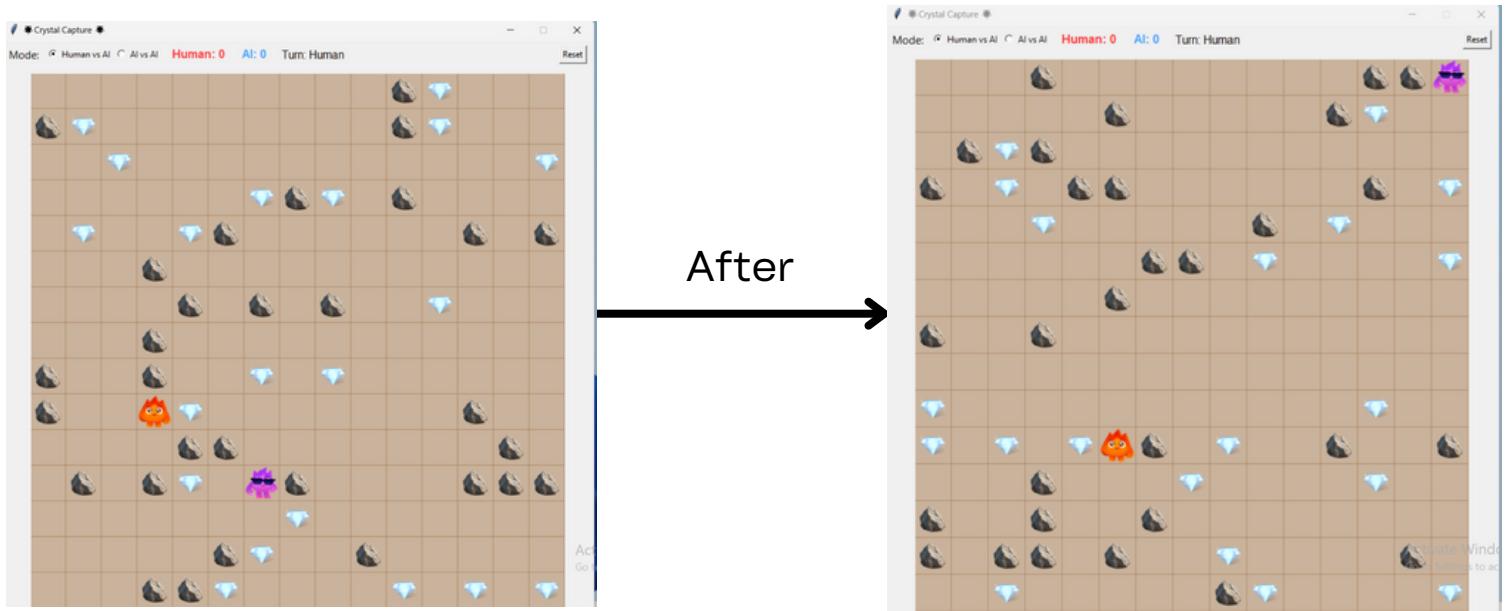
The game's GUI was significantly improved to provide a more engaging and visually appealing experience. Custom graphics were introduced for the agents, obstacles, and crystals, replacing generic placeholders with visually distinct and meaningful sprites. These enhancements not only improved the aesthetic appeal but also made it easier for players to identify game elements and track gameplay in real-time. The updated GUI contributes to a more immersive and polished user experience.



Methodology

02

One of the early challenges encountered was that the game environment was static, meaning that obstacles, crystals, and agents appeared in fixed positions every time the game started. This reduced replayability and predictability. To address this issue, the environment was redesigned to be randomized, so that with every reset, the positions of obstacles, crystals, and agents change dynamically, creating a unique experience in each playthrough.



Methodology

03

As the AI agent was introduced, a major issue emerged: when surrounded by multiple obstacles, the AI began oscillating vertically without making progress. This behavior indicated inefficient path selection. The problem was resolved by integrating the A* (A-Star) pathfinding algorithm, enabling the AI to calculate optimal paths toward targets while avoiding obstacles.

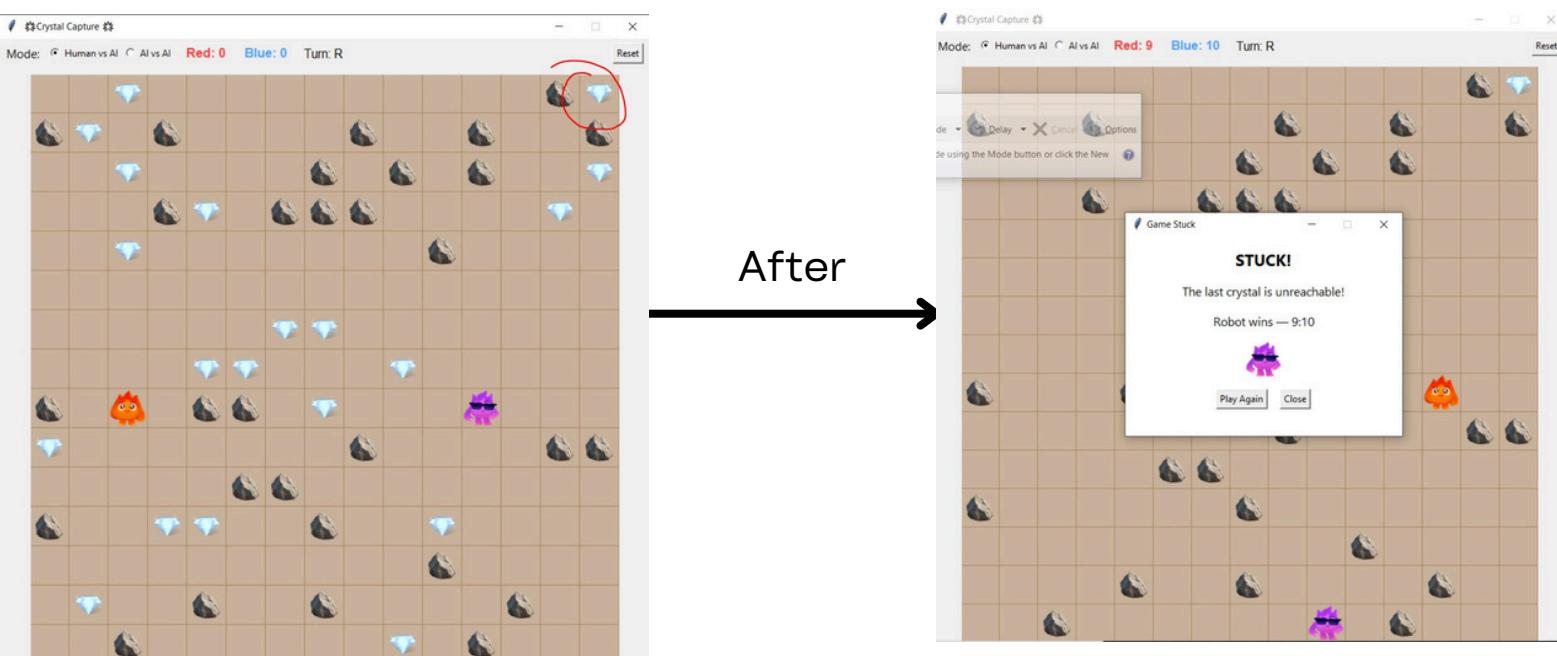
04

Another challenge occurred when the AI agent occasionally stopped on the same cell as the human player, causing movement conflicts. This issue was addressed by refining the AI movement logic using the `ai_turn_step()` and `get_moves()` functions, which ensured valid, non-conflicting moves for both agents.

Methodology

05

Finally, an edge case was identified where crystals could become completely enclosed by obstacles, making them unreachable for both agents. This situation caused the game to stall because distance calculations failed. To resolve this, the `show_unreachable_dialog()` mechanism was implemented. This function detects unreachable scenarios, stops the game gracefully, and declares the winner based on the higher score.



Thanks

we appreciate your attentions