Project Report

# Hexagonal Scrabble – A Strategic Word Game Redefined

FAST NUCES – School of Computing

Course: Artificial Intelligence

Instructor: Ms. Mehak Mazhar

Submission Date: 4th March 2025

## Submitted By

• Muhammad Sohaib (22K-4751)

• Abdul Hadi Khan (22K-4724)

• Zehra Qureshi (22K-4744)

## 1. Introduction

Hexagonal Scrabble is a reimagined version of the traditional Scrabble game, developed using Python and Pygame. It replaces the classic square grid with a hexagonal board, allowing players to place words in six directions, introducing new strategies and complexity. The game includes a basic AI opponent that makes moves using a greedy heuristic-based algorithm.

## 2. Objective

The project aims to:  
- Develop a playable Scrabble variant with a hexagonal layout.  
- Implement an AI opponent capable of making smart moves using heuristic scoring.  
- Validate word formation using a dictionary (SOWPODS) and NLTK.  
- Provide a complete GUI-based game using Pygame.

## 3. Tools and Technologies

• Programming Language: Python  
• Libraries Used:  
 - Pygame: GUI rendering and game loop  
 - NumPy: Mathematical operations and positioning  
 - NLTK: Word validation using the SOWPODS dictionary  
 - Random: For letter rack generation

## 4. Game Mechanics

### 4.1 Board Design

The game board is a hexagonal grid implemented using axial coordinates. Each cell can hold a letter tile. Words can be formed in six directions.

### 4.2 Player Interaction

Players click a hex, type a letter from their rack, and place it. The AI then plays its move automatically. Words are validated after each move, and scores are updated.

### 4.3 Scoring

Words of 3+ letters are valid. Letter values follow traditional Scrabble rules. Each valid word formed adds to the player's or AI's score.

### 4.4 Win Condition

First to reach 150 points wins the game. If both players reach it simultaneously, it’s a tie.

## 5. AI Strategy

### 5.1 Approach

The AI uses a greedy algorithm:  
- It scans every empty cell.  
- Simulates placing every letter (A–Z) in each cell.  
- Uses a scoring function to calculate the value of words that would be formed.  
- Selects the move that gives the maximum immediate score.

### 5.2 Heuristic Design

• Score = Sum of word letter values + directional bonus (for word spread).  
• No planning or blocking — only highest-scoring move is selected per turn.

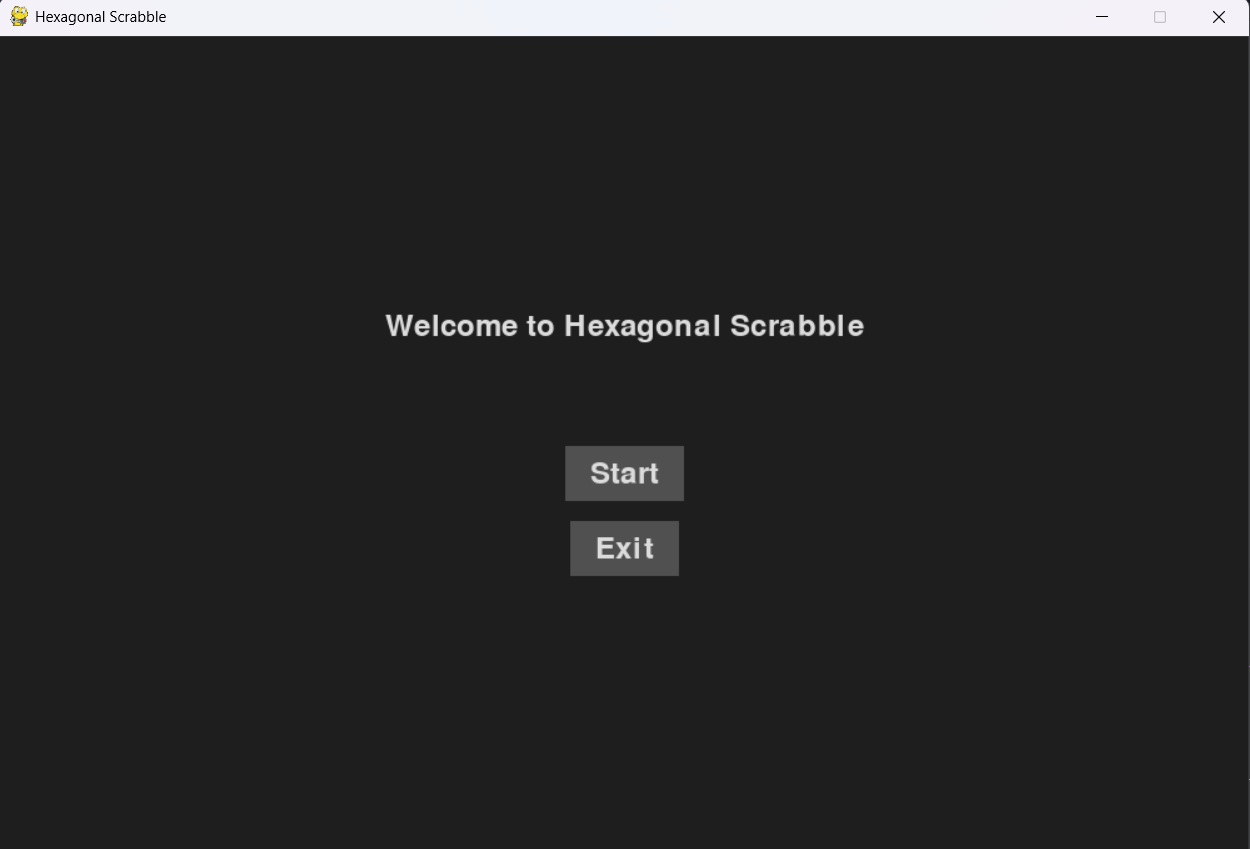
### 5.3 Limitations

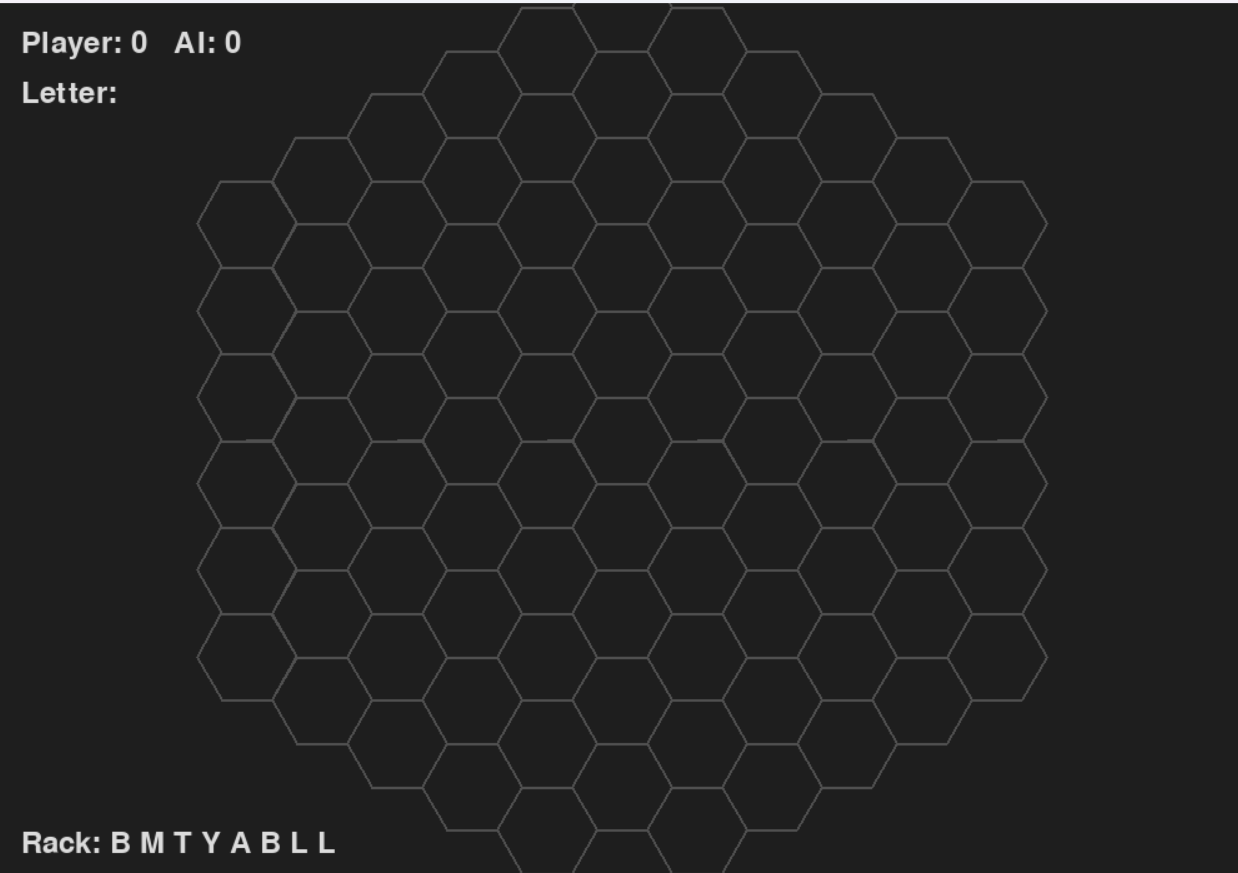
No Minimax or Alpha-Beta pruning implemented. AI does not predict future player moves or defend.

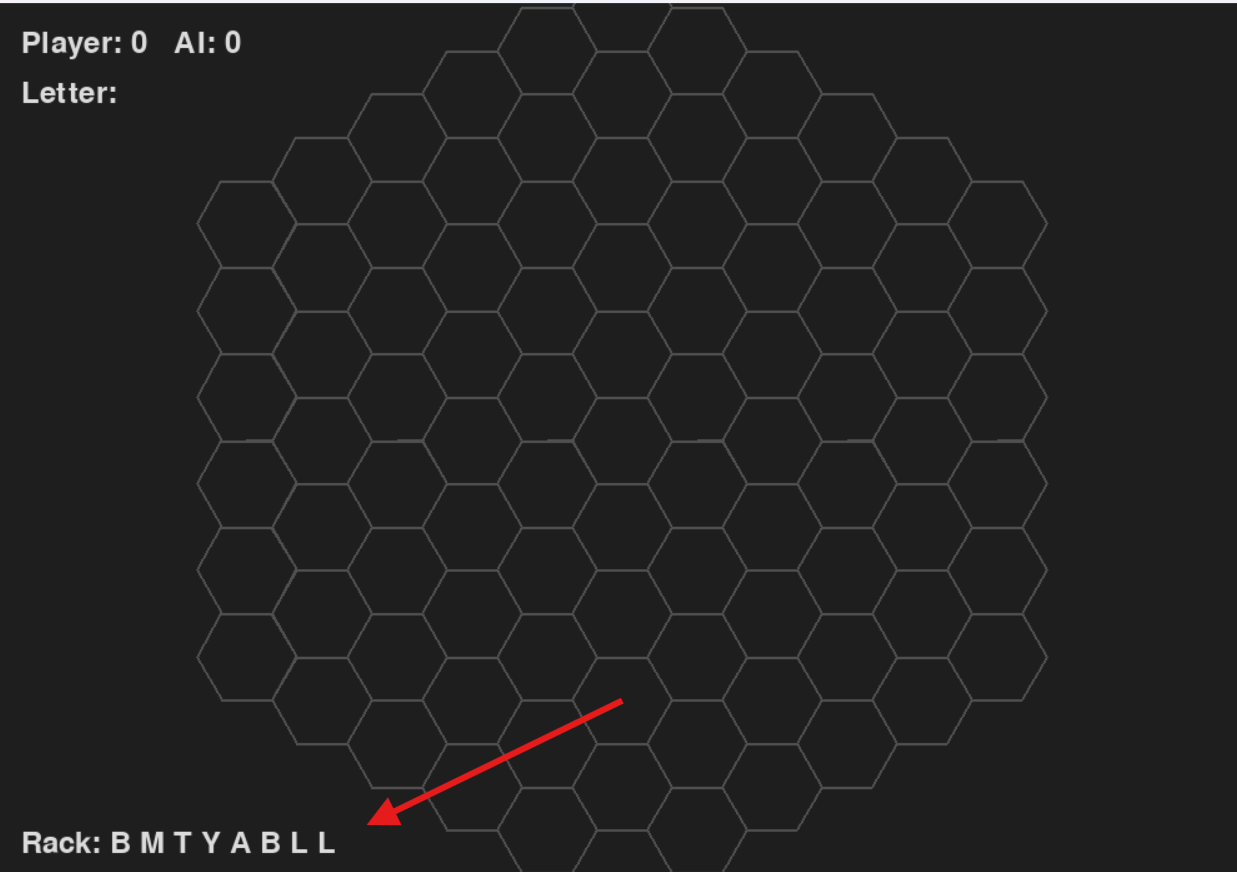
## 6. Implementation Plan

|  |  |
| --- | --- |
| Phase | Task |
| Week 1–2 | Game rules & hex grid setup |
| Week 3–4 | AI logic & greedy move scoring |
| Week 5–6 | GUI, scoring, and word validation |
| Week 7 | AI integration & testing |
| Week 8 | Final testing, demo recording, reporting |

## Results and Screenshots

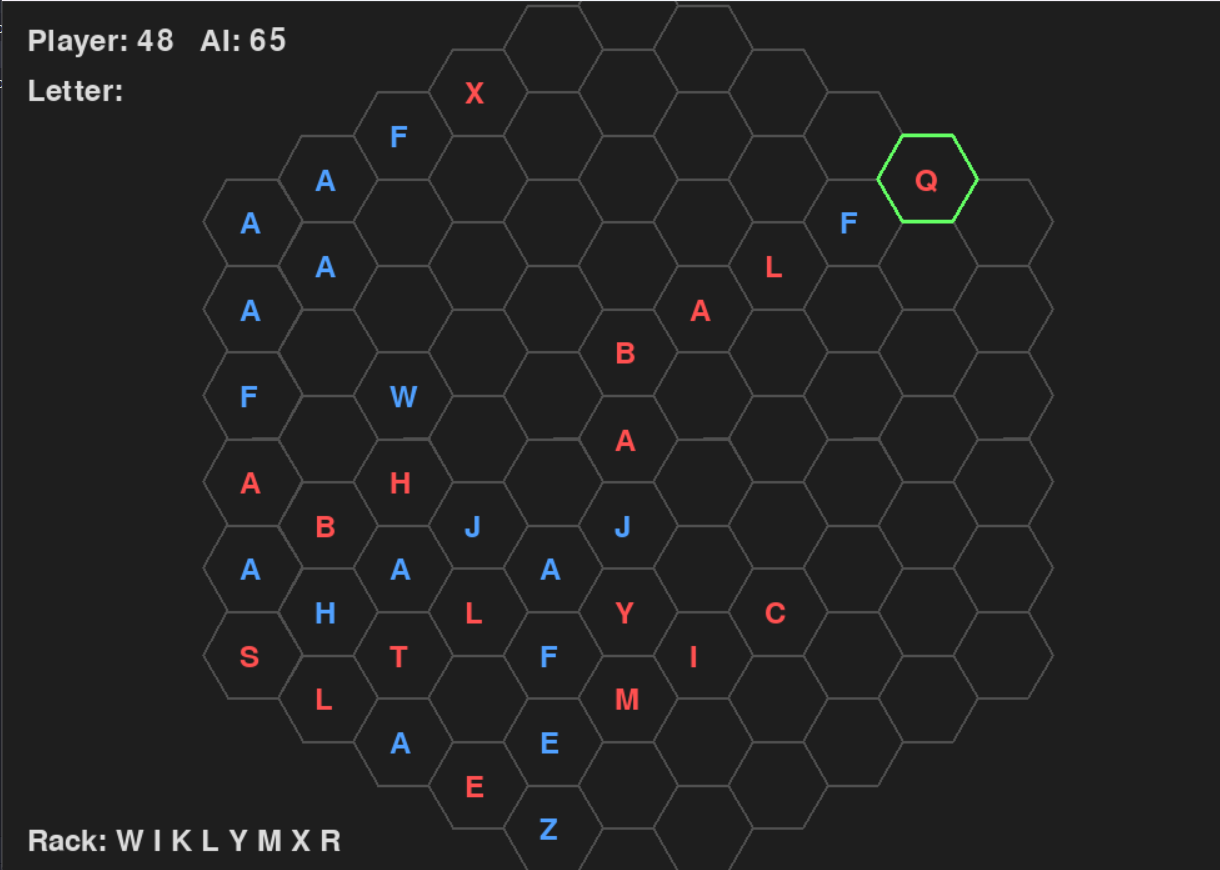






Letters rack





Ai taking the lead

## 8. Challenges Faced

• Accurate hex-to-pixel conversion for rendering hexagons.  
• Validating words in multiple directions dynamically.  
• Ensuring AI doesn’t take too long to compute best move.

## 9. Conclusion

Hexagonal Scrabble was a successful implementation of a strategic word game with basic AI. The use of a hexagonal grid and greedy AI decision-making created a fun and competitive experience. While a full Minimax implementation was out of scope, the project lays the groundwork for future enhancement using deeper AI algorithms.

## 10. References

• Scrabble Official Rulebook  
• Python Docs (Pygame, NumPy, NLTK)  
• Research on Heuristic AI for board games  
• SOWPODS Word List

## Appendix

• Source code available at: [GitHub Link]

• Demo video: [YouTube or Google Drive Link]