**National University of Computer & Emerging Sciences**

**Karachi Campus**



**“SMART QWIRKLE”**

**Project Report**

**Artificial Intelligence**

**Section: CS-6J**

**Group Members:**

1. **22K-4393 Syed Muhammad Hassan Naqvi**
2. **22K-4254 Mishaal Bheraiya**

# 1. Introduction

Qwirkle is a tile-matching game where players aim to score points by creating lines of tiles that share common attributes—either color or shape. Our project, titled Smart Qwirkle, elevates the classic board game by introducing AI opponents with adjustable difficulty, a hexagonal grid layout for increased strategic complexity, and engaging power-ups. The game supports both human and AI players, providing an interactive and intelligent gaming experience.

**2. Objectives**

The primary objectives of the Smart Qwirkle project are to enhance and modernize the traditional Qwirkle game by integrating advanced features and artificial intelligence. Key goals include:

* Hexagonal Grid: Replace the square grid with a hexagonal one using axial coordinates to increase spatial complexity and strategic possibilities.
* Tile Logic: Implement rule-based tile placement, ensuring moves match by color or shape and adhere to adjacency rules.

**Multi-level AI:**

* Easy: Uses uninformed search to select the first valid tile.
* Medium: Uses heuristic logic to choose moves that prioritize score or power-up positions.
* Hard: Uses Minimax with alpha-beta pruning to simulate and evaluate future board states.
* Power-ups: Introduce strategic power-ups—UNDO (revert move), DOUBLE (double score), and WILD (ignore matching rules).
* User Interface: Create a responsive, interactive UI using Pygame for intuitive gameplay.

**3. Problem Statement**

Traditional Qwirkle games are fun but have several limitations:

Grid Limitation: Square grids restrict gameplay diversity and pattern creation.

Basic AI: Rule-based AIs become predictable and non-challenging over time.

No Strategic Depth: There’s little complexity for players who seek advanced gameplay mechanics.

**Smart Qwirkle addresses these issues by:**

* Enabling more movement options with a hexagonal layout.
* Implementing AI that can adapt and respond strategically.
* Adding elements like power-ups and optional fog-of-war to increase depth.

**4. Game Rules Summary**

* Players take turns placing tiles adjacent to existing ones.
* Tiles in a line must share either color or shape, not both.
* Points are awarded based on how many tiles are aligned in a line.
* Power-ups alter gameplay: UNDO, DOUBLE, and WILD.
* The game ends when the board is full or no valid moves are available.

**5. Game Architecture and Features**

**Board Representation:**

* Uses a hexagonal grid managed through axial coordinates.
* Allows movement in six directions and more flexible placements.

**Tiles:**

* 72 unique tiles: 6 colors × 6 shapes (2 of each combination).
* Colors: Red, Yellow, Green, Cyan, Magenta, Blue.
* Shapes: Triangle, Diamond, Square, Circle, Star, Spiral.

**Power-Ups:**

* UNDO: Reverses opponent’s last move.
* DOUBLE: Doubles the player’s current score.
* WILD: Allows ignoring the usual placement rules.

**Placement & Scoring:**

* Validated through adjacency and rule checking.
* Points are calculated by line length, with modifiers from power-ups.

**6. Artificial Intelligence Design**

**Easy (Uninformed Search):**

* Finds the first valid move and plays it.
* Fast but not smart; useful for beginners.

**Medium (Heuristic Evaluation):**

* Evaluates all valid moves and picks the best one based on score potential.
* Prefers power-up tiles and maximizing tile line extensions.

**Hard (Minimax + Alpha-Beta Pruning):**

* Considers future turns for both player and AI.
* Uses pruning to avoid unnecessary calculations.
* Makes strategic decisions based on possible opponent responses.

**7. Technologies Used**

* Language: Python
* GUI: Pygame for visual display and event handling.
* AI Techniques: Minimax (with pruning), heuristic evaluation.
* Optional AI: Monte Carlo Tree Search (future scope).
* Support: Runs on both Windows and Linux systems.

**8. Development Timeline**

|  |  |
| --- | --- |
| Week | Milestone |
| Week 1 | **Game rules finalized; hexagonal grid and tile placement coded** |
| Week 2 | **Basic rule-based AI implemented** |
| Week 3 | **Minimax AI added with pruning** |
| Week 4 | **Power-ups integrated; improved UI** |
| Week 5 | **Testing, bug fixing, report and presentation prepared** |

**9. Challenges Faced**

**Hexagonal Grid Complexity:**

Unlike square grids where neighbors are fixed (up, down, left, right), hexagonal grids require axial or cube coordinate logic.

Understanding how to calculate neighbors, line lengths, and valid tile placements took time and involved building utility functions for conversion and rendering.

**Power-Up Integration:**

Handling special-case tiles like WILD or DOUBLE created exceptions in tile validation and scoring.

For example:

* WILD ignores color/shape rules, so the placement logic needed bypass mechanisms.
* DOUBLE changes how scores are calculated after move validation, not during it.

**AI Performance Bottlenecks:**

Initial versions of Minimax were too slow for larger boards or deeper lookahead.

**Solved by:**

* Adding alpha-beta pruning.
* Limiting depth dynamically based on tile count.
* Prioritizing early pruning for invalid/low-score moves.

**Testing AI Interactions:**

* Ensuring AI and power-ups didn’t conflict required manual and automated test cases.
* Edge cases like AI using WILD, placing on DOUBLE, or being UNDO’ed by the player were hard to simulate and debug.

# 10. Conclusion

Smart Qwirkle transforms a classic board game into a modern AI-powered experience. By integrating intelligent opponents, strategic power-ups, and a spatially dynamic grid, the project demonstrates core principles of artificial intelligence in game design. The system is extensible for future improvements, such as fog-of-war mode or reinforcement learning agents.