A Scrabble Artificial Intelligence Game

SJSU SAN JOSÉ STATE UNIVERSITY

Advisor
Dr. Chris Pollett
Committee Members
Dr. Philip Heller
Dr. Robert Chun

By
Priyatha Joji Abraham

Agenda

- Introduction
- Background
- Overview of Maven and Quackle AI
- End game strategies
- Experiments and Observations
- Conclusion and Future work
- Questions

Project Goal

- Build a two-player Scrabble AI game engine.
- Create two AI computer players having different move generation heuristics and endgame strategies.
- Evaluate performance based on the winning criteria, quality of moves, and time consumption.
- Develop variants of world-champion computer AI, ‘Maven,’ and another AI, ‘Quackle.’

Introduction

- Artificial Intelligence (AI) is a collection of techniques used to build intelligent agents that perceive its environment and act rationally to accomplish the goal.
- Games make an ideal test-bed for AI applications.
- Scrabble is a crossword board game, widely popular for its game strategy.
- Scrabble computer players have already surpassed human players in tournaments, however, efficient heuristics are required to defeat another Scrabble Computer AI.
- Maven and Quackle AI compete against each other using its three-ply future look ahead simulation technique and different move generations.

Problem Statement

- Scrabble is an imperfect information game:
  partial information about the game state with a large branching factor.
- An imperfect information state poses distinct challenges for decision making agents, e.g., players are unaware of opponent’s rack, and it's hard to predict next move.
- End-games are complex and crucial to the success of the game.
- End-games are real-world situations where the move options are limited and model strategy to maximize rewards.
- Normal high score yielding approaches are not good enough for endgames.
- Scrabble requires quick move generation techniques.

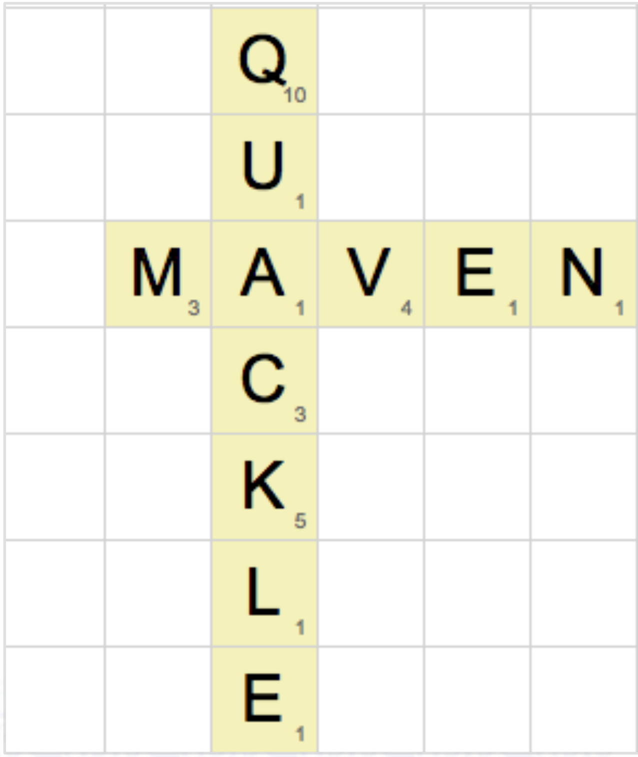
Related Works

- Maven AI created by Brian Sheppard[1] in 2002 defeated several world champion, human players.

- Quackle AI created by Jason Katz-Brown and John[2] in 2006 has beaten world champion David Boys.

- An AI Game engine developed by Stanford students in 2016 between Quackle AI and a new Computer AI with Monte Carlo Simulations of depth 2.

- They focused on mid-game scenarios and not end games due to time limitations.



Scrabble Game Rules and Terminology

- ***Board***: 15x15 grid game board.
- ***Tiles***: Fills each board square; holds letter and value.
- ***Tile Bag***: Holds 100 tiles, 98 are alphabets and 2 blanks
- ***Rack***: Group of seven tiles holds by each player.
- ***Rack Leave***: Letters left out on the rack after one play.
- **Bingo**: Play which uses all the seven letters on the rack.
- ***Premium Squares:*** Squares that earn bonus points.
- ***Legal Moves:*** No diagonal plays, player must extend the words on board.
- ***Hot Spots***: Excellent squares on the board with bonus-scoring opportunities.
- ***Ply***: Player's turn or half a move in a two player game.

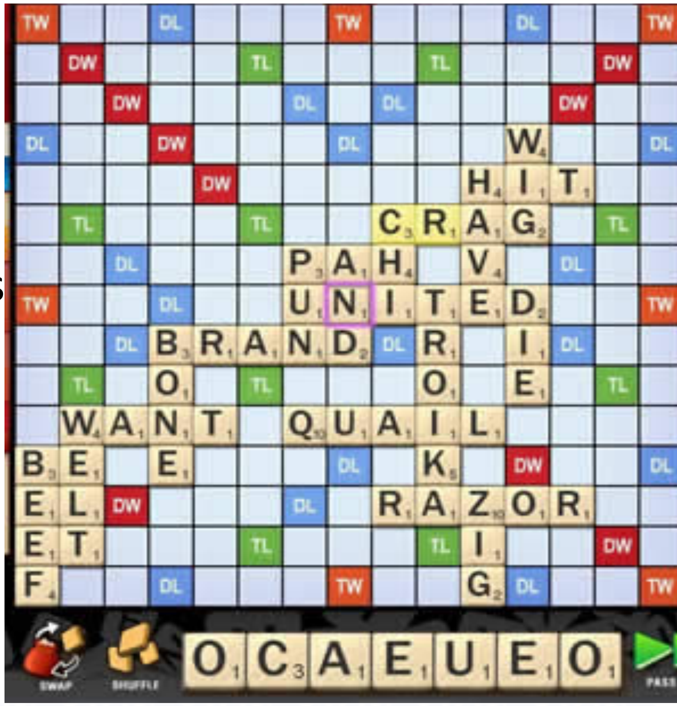


Fig 1. Conventional Scrabble game board [2]

Maven AI and Quackle AI

- We implemented Maven AI and Quackle AI from scratch in Java.
- Maven applies 3-ply look ahead strategy and various heuristics for simulation.
- Quackle calculates win probability estimation using 3-ply look ahead strategy.
- Quackle AI serves as a benchmark to measure the performance of Maven AI variants.

Dictionary and Lexicon Representation

- Official Scrabble Players Dictionary (OSPD) size: 1.8 MB contains >100,000 words.
- Our AI player used a trimmed dictionary of size: 1KB contains 232 words.
- Maven uses a Directed Acyclic Word Graph (DAWG) to represent dictionary that enables fast move-generation.
- DAWG is a minimal representation of a trie data structure.

Trie and DAWG

TRIE

DAWG

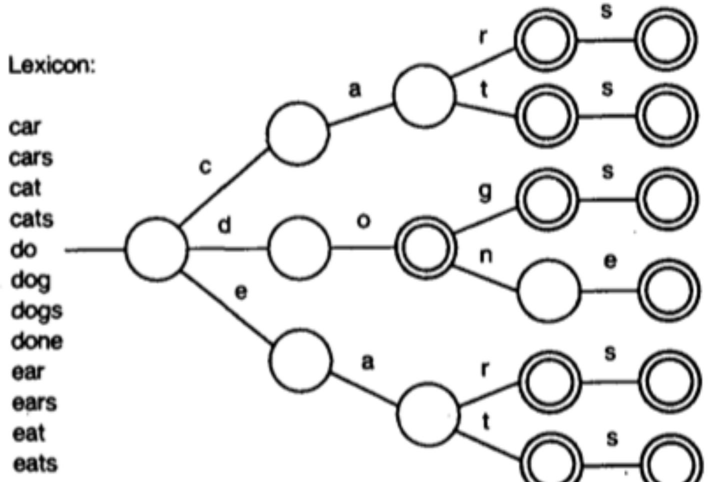


Fig 2. Trie with its lexicon [5]

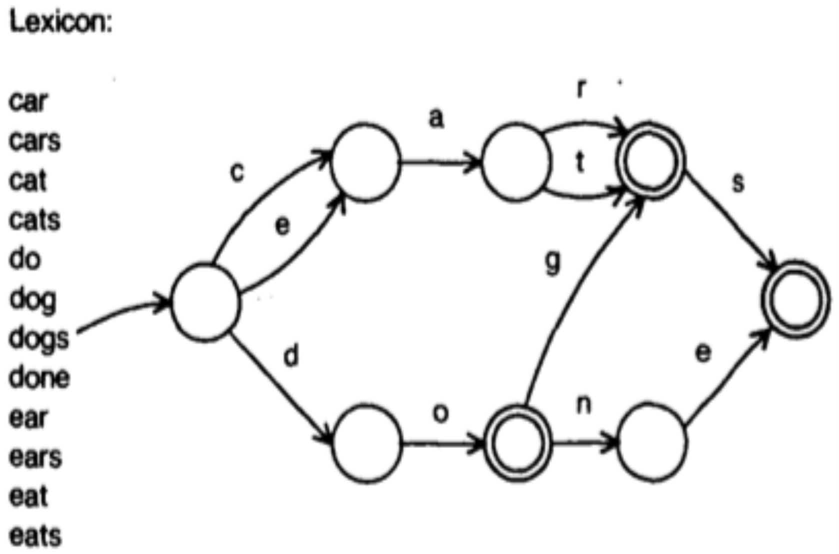
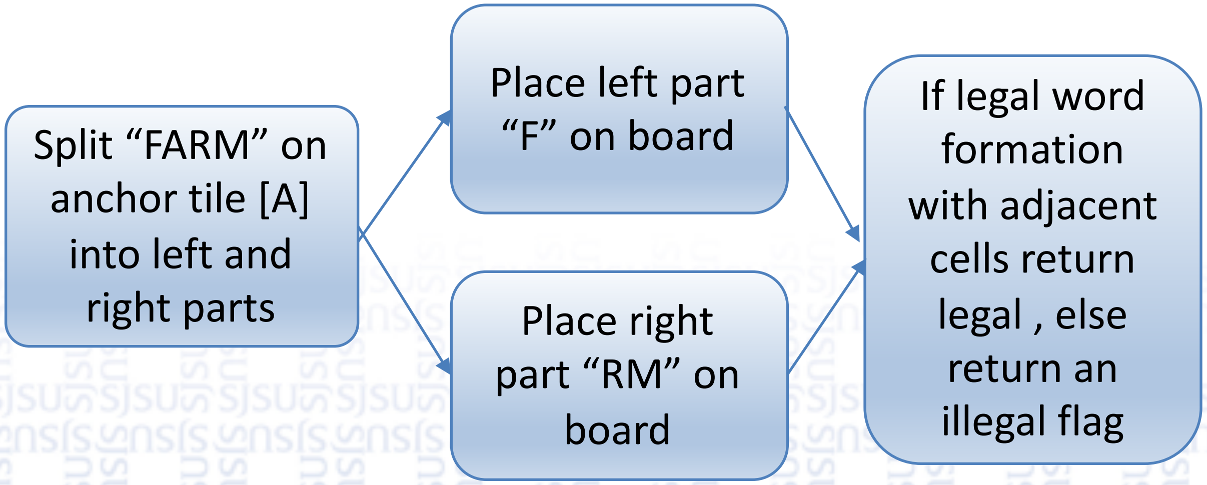


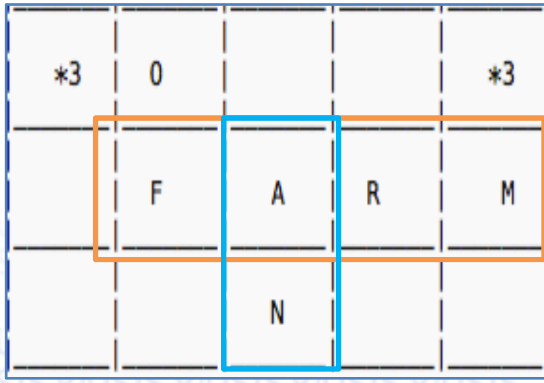
Fig 3. DAWG with its lexicon [5]

- We used trie and a word-list representation for storing the entire dictionary.

Maven Move Generation

- Maven's move generator applies different heuristics on each play to find the best candidate move from 20-30 candidate moves.
- Only legal placements are considered on the board.
- E.g.: The best possible word: "FARM," place across [A]N (Horizontal direction).


Split "FARM" on anchor tile [A] into left and right parts
Place left part "F" on board
Place right part "RM" on board
If legal word formation with adjacent cells return legal , else return an illegal flag



Rack: MFRFGTH

Fig 4. An example of horizontal legal placement

Maven Heuristics

- Heuristics: Set of evaluation parameters to instantly solve the problems when traditional approaches are often slow.

- Four main heuristics used in Maven, implemented in our project:

  1. Vowel-Consonant Balance
  2. U-With-Q-Unseen
  3. Hot spot block
  4. First-Turn Open

SJSU SAN JOSÉ STATE UNIVERSITY

Maven Game Phases

First Move

- 100 tiles in the bag.
- First turn open except Bingo.

Mid-game

- Beginning until 9 or fewer tiles in the bag.
- Greedy scoring strategy.

Pre-endgame

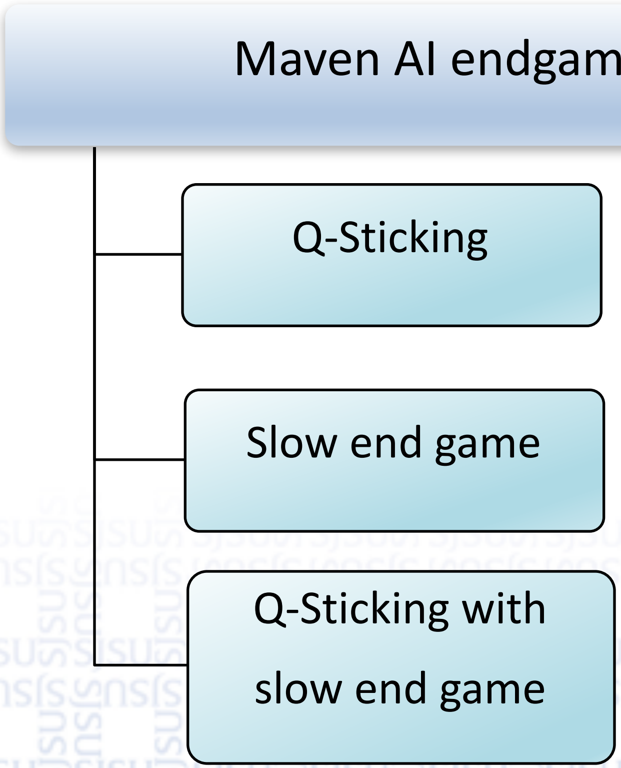
- Total of 16 unseen tiles.
- 9 in bag and 7 in opponent’s rack.
- Combines mid and pre-endgame.

End-game

- Zero tiles in the bag.
- 3 endgame strategies.

Maven End-Game Strategies

- Endgames are complicated as player should make a move in a controlled manner to maximize rewards within the time limit.



- When opponent stuck with unplayable tiles like Q or V.
- Block the big hot spot and play greedy approach.

- Maximize the value of tiles by playing out slowly.
- Delay the game using low-scoring tiles.

- Block the opponent moves using low-scoring tiles if player is behind.
- If player is ahead, close the game by maximizing reward.
- Trap the opponent with awkward tiles.

Fig 5. Maven AI variants

SJSU SAN JOSÉ STATE UNIVERSITY

Maven AI Look- Ahead and Simulation

- Maven computes 3 ply look-ahead of future moves.
- Ply 1: Place the best scoring candidate word for a given rack and board state.
- Ply 2: Find the opponent’s possible candidate word and resultant score if that word is placed.
- Ply 3: Evaluate the current state and find the best word to block that move in an endgame.

Quackle AI Schematic Diagram

- Quackle considers rack leave heuristics.
- Quackle has following two components:

i) Kibitzer:
It applies static evaluation and ranks candidate moves quickly.

ii) Simulation engine:
It applies 3-ply look ahead and repeated 100-300 times for 100 random racks and calculates point differential.
Point diff = Our Total Score – Opponent's Score + Residual rack leave value estimate

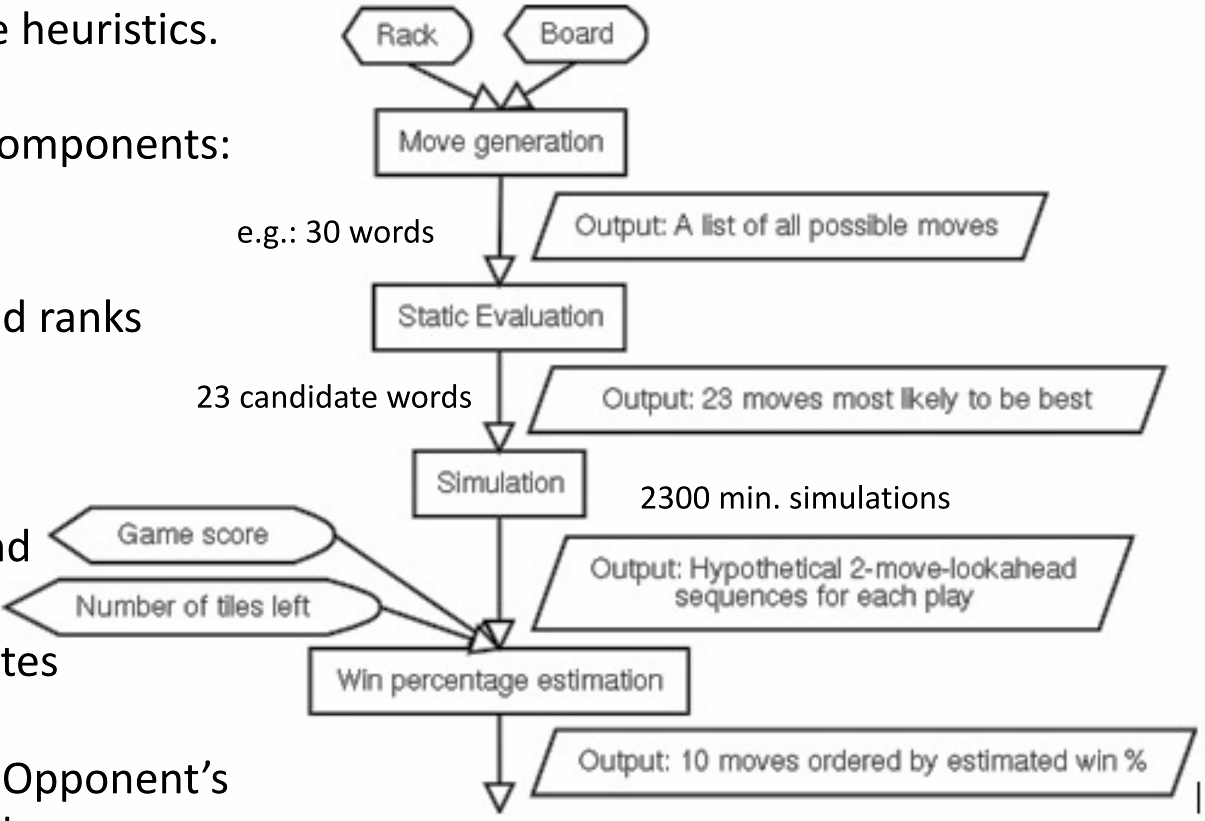
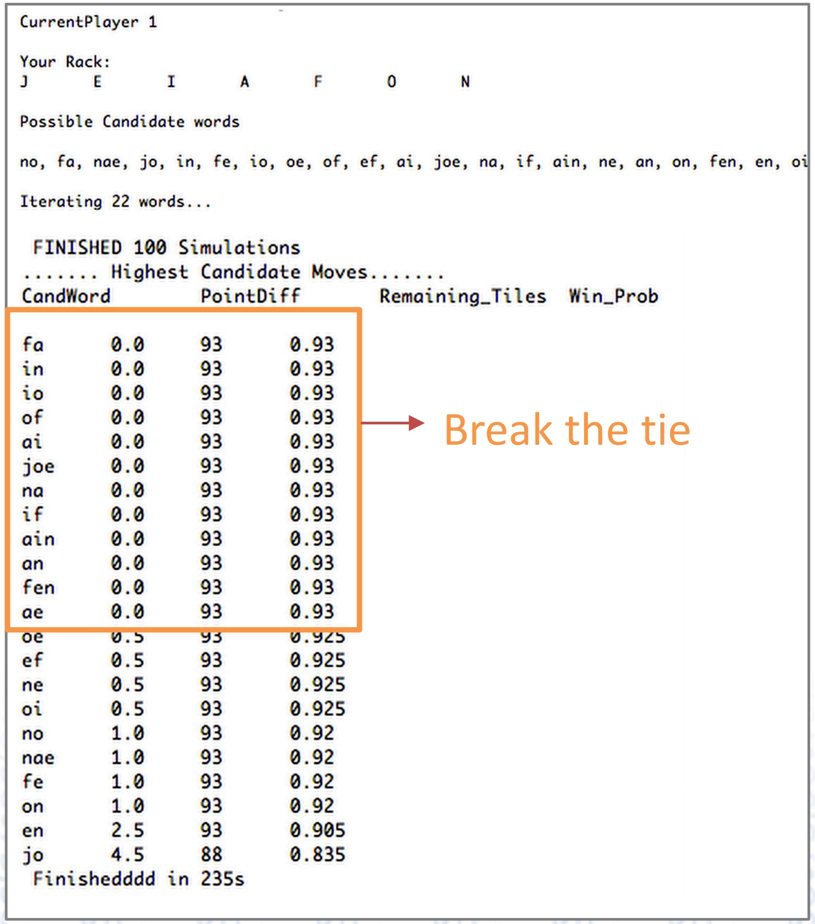

Rack
Board
Move generation
e.g.: 30 words
Output: A list of all possible moves
Static Evaluation
23 candidate words
Output: 23 moves most likely to be best
Simulation
2300 min. simulations
Game score
Number of tiles left
Output: Hypothetical 2-move-lookahead sequences for each play
Win percentage estimation
Output: 10 moves ordered by estimated win %

Fig 6. Quackle work flow [2]

Quackle Win Percentage Estimation

- Quackle determines winning probability based on the point differential and the remaining tile count for each candidate move.
- Candidate move list is sorted in descending order of winning probability percentage.
- If words are found with the same probability, break the tie by selecting the word with lowest rack leave value.



```
CurrentPlayer 1

Your Rack:
J       E       I       A       F       O       N

Possible Candidate words

no, fa, nae, jo, in, fe, io, oe, of, ef, ai, joe, na, if, ain, ne, an, on, fen, en, oi

Iterating 22 words...

 FINISHED 100 Simulations
....... Highest Candidate Moves.......
CandWord             PointDiff           Remaining_Tiles  Win_Prob

fa       0.0       93       0.93
in       0.0       93       0.93
io       0.0       93       0.93
of       0.0       93       0.93
ai       0.0       93       0.93
joe      0.0       93       0.93
na       0.0       93       0.93
if       0.0       93       0.93
ain      0.0       93       0.93
an       0.0       93       0.93
fen      0.0       93       0.93
ae       0.0       93       0.93
oe       0.5       93       0.925
ef       0.5       93       0.925
ne       0.5       93       0.925
oi       0.5       93       0.925
no       1.0       93       0.92
nae      1.0       93       0.92
fe       1.0       93       0.92
on       1.0       93       0.92
en       2.5       93       0.905
jo       4.5       88       0.835
 Finishedddd in 235s
```

Break the tie

Fig 7. Quackle simulation example

Experiments and Observations

- Experiments are conducted on 10 randomly generated Scrabble endgames.
- Executed on a Mac Machine with 16 GB RAM-quad-core processor running Java version 7.
- We loaded real-world end-game scenarios [8, 11, 13] used in tournaments into our project considering current game board state, player rack, player score, tile bag and score lead.

Experiment 1: Maven Q-Sticking AI versus Quackle AI

Experiment 2: Maven slow end game AI versus Quackle AI

Experiment 3: Maven Q-Sticking-slow end game AI versus Quackle AI

Experiment 4: Maven Q-Sticking-slow end game AI versus No-Q-Sticking

Experiments and Observations

- **How to run the experiment?**
- Create Maven and Quackle AIs in two separate classes.
- Make function calls to firstmove() of game open AI and bestmove() of both AIs for subsequent plays alternatively until aRun experiment for 10 random Scrabble end-games simulated in the program.
- Winner is detected.
- Mark the scores and calculate the average mean score score and standard deviation (SD).
- Determine the winner and compare the point score spread.

Experiments and Observations

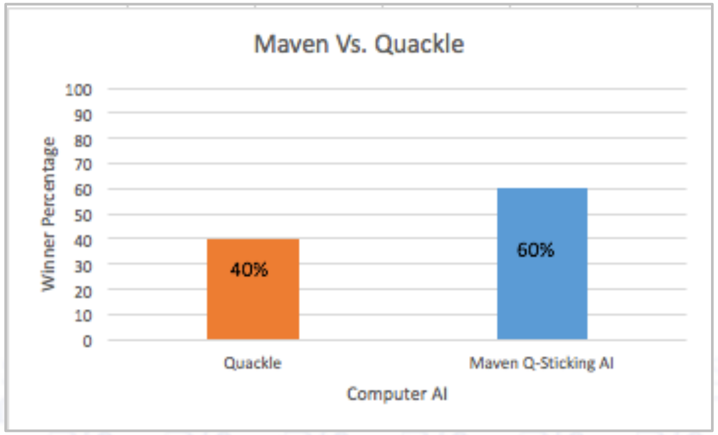
I. **Experiment 1:** Maven Q-Sticking AI versus Quackle AI for End game conducted on randomly generated Scrabble endgames.

Hypothesis: We predicted that Maven Q-Sticking AI will outscore Quackle as Maven AI can block the hot spots of Quackle when Quackle is stuck with Q and play with a greedy approach.

Observations:

Many factors influenced the winner.

- Blocked hot-spots
- Score difference
- Rack tile value
- Number of rack tiles
- Kind of unplayable tile



Player AI	Mean Score + Standard Deviation
Maven Q-Sticking AI	473.2 ± 70
Quackle	450.4 ± 59.2

Result: Maven outscores Quackle. Hypothesis confirmed.

SJSU SAN JOSÉ STATE UNIVERSITY

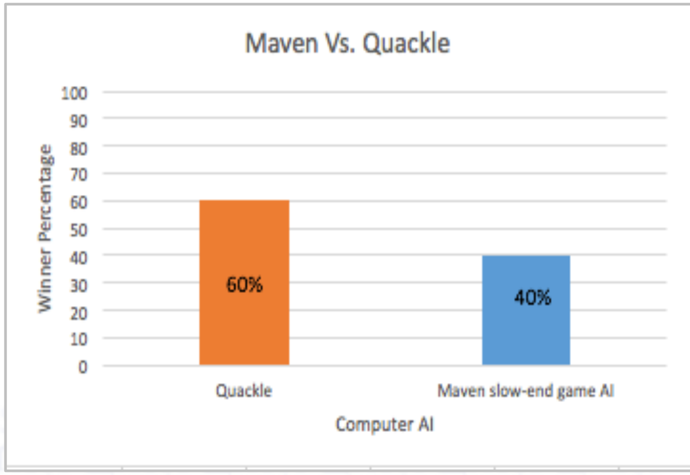
Experiments and Observations

II. Experiment 2: Maven slow endgame AI versus Quackle AI for End game conducted on 10 randomly generated Scrabble endgames.

Hypothesis: We predicted that Maven will outscore Quackle.

Observations:

- High Score lead
- Maven tiles finished soon, could not drag endgame
- High point unplayable tile



Result: Quackle outscores Maven.
Hypothesis invalid.

Player AI	Mean Score + Standard Deviation
Maven slow-endgame AI	461.9 ± 80
Quackle	442.6 ± 60

SJSU SAN JOSÉ STATE UNIVERSITY

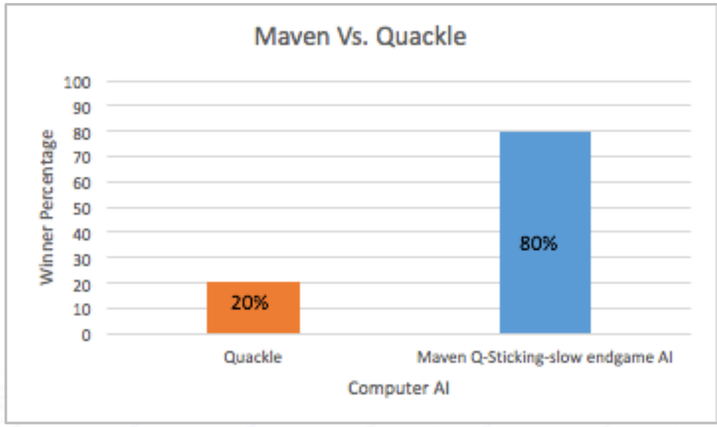
Experiments and Observations

III. **Experiment 3:** Maven Q-Sticking-slow end game AI versus Quackle for 10 random end games.

Hypothesis: We predicted that Maven will outscore Quackle as it block all the spots

Observations:

- Blocked hot spots
- Blocked single opportunity
- Low scoring tiles maximized rewards
- Placement on premium bonus
- Gained penalty score
- High score variation



Player AI	Mean Score + Standard Deviation
Maven Q-Sticking slow-end game AI	472.2 ±86.79
Quackle	417.9 ± 54.63

Result: Maven outscore Quackle. Hypothesis confirmed.

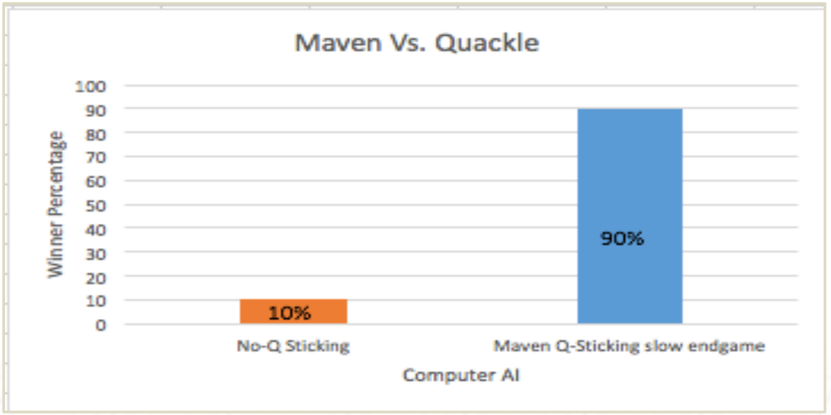
Experiments and Observations

III. **Experiment 3:** Maven Q-Sticking-slow end game AI versus No-Q-Sticking for 10 random games.

Hypothesis: We predicted that Maven will outscore Quackle as it block all the spots.

Observations:

- Blocked hot spots
- Blocked single opportunity
- Low scoring tiles maximized rewards
- Fast play-out
- Penalty score
- High score lead



Player AI	Mean Score + SD
Maven Q-Sticking slow-end game AI	479.9 ± 75
No-Q Sticking AI Score	415.1 ± 45

Result: Maven outscored Quackle.
Hypothesis confirmed.

Conclusions and Future Work

- Maven AI with Q-Sticking-slow end game AI plays the best.
- Maven blocks hot spots as well as earn bonus points by trapping the opponent.
- No Q-Sticking AI are most favorable during mid-games.
- Project takes 100-255 s to finish 100 simulations of Quackle, originally 19-50 s.
- Maven AI takes 0-2000 ms, originally 0-30 ms for one move generation.
- In future work, we will reduce the simulation time and implement an interactive graphical UI for the project.
- Quackle data analysis features can be incorporated into our AI which simulate deeper plies up to 1000 iteration.

References

[1] B. Sheppard, "World-championship-caliber Scrabble," *Artificial Intelligence*, vol. 134, pp. 241- 275, Jan. 2002. [Online]. Available: https://smp.uq.edu.au/sites/smp.uq.edu.au/files/WorldChampionshipScrabble.pdf

[2] J.K. Brown and J. O'Laughlin, "How Quackle Plays Scrabble," 2007. [Online].Available: http://people.csail.mit.edu/jasonkb/quackle/doc/how_quackle_plays_scrabble.html

[3] S. Russell and P. Norvig, "Adversarial search," in *Artificial Intelligence: A Modern Approach*, 3rd ed. New Jersey: Pearson, 2010, Ch. 5, pp. 161-189.

[4] Review: Scrabble for iPad, [Online]. Available: https://appadvice.com/appnn/2010/04/review-scrabble-ipad

[5] C. Josephson and R. Greene, "CS221 Fall 2016 project final report: Scrabble AI," 2016. [Online]. Available: https://web.stanford.edu/class/cs221/2017/restricted/p-final/cajoseph/final.pdf

[6] N. Gupta *et. al,* "AI Agent to play Scrabble," 2017, [Online]. Available: https://web.stanford.edu/class/cs221/2017/restricted/p-final/rsbauer/final.pdf

References

[7] S.A. Gordon, “A faster Scrabble move generation algorithm,” Software: Practice and Experience, vol. 24, no. 2, pp. 219- 232, Feb. 1994. [Online]. Available: http://ericsink.com/downloads/faster-scrabblegordon.pdf

[8] “Endgame finesse”, Australian Scrabble® Players Association, Mar. 2005. [Online]. Available: http://www.scrabble.org.au/strategy/endgame.htm

[9] J. Mandziuk, “Maven and Quackle-The top level machine scrabble players,” in Knowledge-Free and Learning-Based Methods in Intelligent Game Playing, Chennai, India: Springer, 2010, Ch.4, pp. 45 - 47.

[10] A. W. Appel and G. J. Jacobson, “The world's fastest Scrabble program,” *Communications of the ACM* pp. 572-576, May. 1988. [Online]. Available: http://www.gtoal.com/wordgames/jacobson+appel/aj.pdf

[11] “Mastering the endgame”, BattleLine Games LLC, 2013. [Online]. Available: http://www.2letterwords.com/Mastering_the_End_Game.html

[12] “Breaking the game,” 2014. [Online]. Available: http://www.breakingthegame.net/strategy

[13] J. Edley and J.Williams, “Everything Scrabble: Third Edition,” 2009, pp. 255- 315. [Online].Available: https://books.google.com/books?id=Uml_kpOO64gC&lpg=PP1&pg=PR5#v=onepage&q=end%20game&f=false

Questions

SAN JOSÉ STATE UNIVERSITY *powering* SILICON VALLEY

Thank You!