CS540 Artificial Intelligence: Project Proposal
Casey Pore
Spring 2015

The focus of my project will be the application of modern techniques for creating a formidable agent capable of playing Scrabble. You are probably familiar with the game of Scrabble. Scrabble is a game for 2-4 players played on a 15 x 15 grid. In the English version of the game, players draw and hold 7 tiles at a time from a set of 100. Each tile (except for 2 wild-card tiles) has a letter and a corresponding point value printed on them. The 7 tiles held by each player is known as that players "rack." Players take turns arranging tiles on the board from their rack to form words. Tiles must be placed such that new words must use at least one letter from an existing word on the board and are to be placed either horizontally or vertically, but not diagonally. Points are awarded by summing the value from each letter of the word, and the sum is added to a running score kept for each player. The player with the highest score after all tiles from the available set have been placed wins. In addition to points earned from the face-value of the tiles, players can earn extra points by placing tiles on top of specially-marked "premium" squares. These premium squares can double or triple the value the letter placed on it, or the sum for the entire word, depending on the square.

Before diving into the details about what I plan to accomplish within the scope this project, I would like to note that I have goals for it that extend beyond the current semester. I hope to parlay this project as the basis for another project in Computer Vision/Optical Character Recognition, perhaps as a project for CS545 to be taken in another semester, if not just in my spare time. The ultimate goal is to create an Android app to made available on the Google Play Store. The idea is to use the camera on an Android device to capture an image of a Scrabble board in a state-of-play. The software would then use CV/OCR to identify words and their placement on the board. The identified board state, along with the player's current rack would then be fed as an input into the scrabble agent I hope to produce this semester. The software would then produce a board (or perhaps several board options) showing the placement of words that could be played, along with suggesting/highlighting the words that would produce highest score. With that out of the way, onto what I plan to accomplish this semester...

As already stated, I plan on creating an agent to play Scrabble. As I hope to ultimately create an Android app, my implementation will be written in Java. There is a wealth of information about Scrabble playing agents available from a simple internet search, with the earliest programs appearing in the early 1980s. In 1982, Stuart Shapiro presented a program implemented in SIMULA 67 that introduced many of the data structures and fundamental search techniques that are still used today as the basis for modern implementations[1]. It is claimed (on Wikipedia) that the current best Scrabble agent is Maven, introduced by Brian Shepard in 2002[2]. The focus of my project, however, isn't necessarily to produce "the best" agent, in terms of optimality, but rather the focus will be on producing a fast agent that can provide best solution possible under the constraint of time. Scrabble tournament rules state a player has a full 25 minutes to decide on a word to play. In the context of an application where a user would expect a fast response, it would be unacceptable to wait that length of time, so my implementation would have to produce acceptable results on the order of seconds, rather than minutes. However, this could be a tunable setting if the user desired better results at the expense of time. For my project, I will attempt to replicate the results from Appel and Jacobson's implementation reported in

volume 31 of the Communications of the ACM, where they were able to attain an average computation time of 1.4 seconds per move[3]. This may include incorporating existing Scrabble dictionaries into my code, well as examining existing implementations (Appel and Jacobson or otherwise) to leverage ideas for my own implementation.

The goals I hope to accomplish for this project during the course of this semester are outlined below:

- User Interface
  - For this semester, the UI stall be text only. The focus will be on creating the agent, not visual design. Future work would include creating a user interface suitable for an Android device.
- Number of players
  - At a minimum, two players will be supported. In addition to the scrabble agent, the program shall support human players. Any combination of human and agent players shall be possible. If time permits, the full four players and a mixture of agents may be supported.
- Speed
  - The program shall be able to find as best as possible moves on the order of seconds. Of course, evaluation of my implementation using longer search times will be conducted in order to gauge it efficiency. Time/optimality trade-offs will be reported.
- Optimality
  - The focus of this project will be on speed, but any opportunity to maximize winning move selections will be taken and reported on.
- Correctness
  - The agent shall never choose or suggest invalid moves. The agent shall not be given any information to select moves that may lead to an unfair advantage (such as knowing what tiles are unplaced, or what's on its opponent's rack.)
- Future Work
  - Lay the foundation for future a project in Compute Vision/Optical Character Recognition and an Android application.

Given the large amount of literature available regarding creating a Scrabble playing agent, I realize that this project may be considered old hat for an AI topic. However, I don't believe it precludes this from being sufficiently challenging for this course and will be an interesting project to work on. “Agents” is a topic covered later in the semester, and this will be an opportunity for me to explore that topic of Artificial Intelligence in greater detail. The unique properties of Scrabble (i.e. partially observable, imperfect information) and how the state is represented will make this an interesting problem to work on. Exhaustive search is impractical, and it has been shown that typical local search techniques, such as simulated annealing, are inefficient for this particular problem[2]. Domain-specific knowledge that make clever use of search algorithms and data structures seem to be the path forward, and tuning the agent for speed will be a unique challenge all its own.

[1] S.C. Shapiro, Scrabble crossword game-playing programs, SIGART Newsletter 80 (1982). http://www.cse.buffalo.edu/~shapiro/Papers/ubtr119.pdf

[2] Sheppard, Brian (2002). "World-championship-caliber Scrabble.". *Artificial Intelligence* **134** (1-2): 241–275. https://smp.uq.edu.au/sites/smp.uq.edu.au/files/WorldChampionshipScrabble.pdf

[3] A.W. Appel, G.J. Jacobson, The world's fastest Scrabble program, Comm. ACM 31 (5) (1988) 572–578, 585. http://www.gtoal.com/wordgames/jacobson+appel/aj.pdf

*Instructor's response:
This is very well written (modulo typos, please proofread!) and indicates considerable thought. It is not necessary to produce the world's best Scrabble player. I am most interested in what AI techniques you will be using. I have a pretty good Scrabble App on my iPhone that plays at the level you are suggesting and it is fast. I would rather this did not turn out to be about optimizing data structures and string search. A more AI spin on this might be to incorporate lookahead search so that you trade-off maximizing points now with leaving opportunities for later or look at opponent modeling (see http://www.aaai.org/Papers/IJCAI/2007/IJCAI07-239). I appreciate that you are downplaying the UI for the AI. There must be public domain Scrabble implementations to start with, no? I would be ok with your re-implementing the algorithm in [2].