NOVEMBER 2024

YAHTZEE GAME REPORT

SE1012

PROGRAMMING METHODOLOGY ASSIGNMENT

BSC (HONS) IN COMPUTER SCIENCE
YEAR 1 SEMESTER 1

PREPARED BY: K.S.R PERERA
IT24101845

01

INTRODUCTION

This Yahtzee game is a C-based, CLI (Command Line Interface) implementation of the classic dice game. This text-based version recreates the same experience as the classic dice game where the goal is to achieve the highest score by rolling dice to match specific combinations. The game follows the traditional rules of Yahtzee and feature a human player and a computer AI player that follows a programmed strategy for optimal decision making. With a user-friendly text-based interface, this version of Yahtzee can be played directly in the terminal allowing players to have a fun and straightforward experience.

STRATEGY USED FOR COMPUTER PLAYER

The computer player utilizes a strategic decision-making approach designed to score the highest possible score by prioritizing high-value scoring opportunities. The strategy involves evaluating the outcome of the end of the each roll and making the decision to either keep dice or reroll for better opportunities.

Initial roll Evaluation

After the first roll in each round, the computer sequentially examines whether there are any high-value scoring category available, such as a Yahtzee , Large Straight, Four of a kind (with high values like 5s or 6s) or Full House that could be immediately achieved. When such a high-score option is available, the AI prioritizes scoring it without further rolls to secure these rare opportunities.

Reroll Strategy

If no immediate scoring category is selected after the initial roll, the computer will reroll specific dice to enhance its chances in high-scoring categories. The AI prioritizes going for high value categories Yahtzee, Four of a Kind, and Large Straight as well as aiming for higher numbers in the upper section by keeping any dice that contribute toward those patterns. For example, if multiple dice show high values like 6's, the computer will keep these to aim for Yahtzee or Four of a Kind. Additionally, if there's a small straight, the AI keeps the dice that are part of the sequence to complete it, aiming for a large straight.

End of Turn Decision

Following the last dice roll, the computer assesses its combination of dice to determine the best scoring category available. It examines the scoring categories in a specific order based on a hierarchy that prioritizes higher scoring potential. This approach guarantees that the AI optimizes point totals with each roll result, even in situations where luck does not support high-value categories.

Fallback Strategy

The AI has multiple fallback strategies determine the best scoring category when a high-value scoring option isn't available. If the computer hasn't chosen a category yet, it attempts to score in any remaining upper section category with the lowest default score preventing leaving leaving high-value options unutilized and preserving them for later rounds. If all upper section categories are exhausted or low-scoring, the computer checks the remaining lower section categories, selecting one with the highest possible score. This ensures that if any of these categories can yield a high score, they will be prioritized over a smaller upper section score.

Adaptability for Bonus

The computer also strategizes to fill the upper section by prioritizing high-value dice to secure the 35-point bonus. This adaptability helps the computer achieve consistent scoring progress even when optimal combinations are rare.

CHALLENGES FACED DURING IMPLEMENTATION

Coming up with a scoring strategy

Developing an optimized scoring strategy for the computer player was a significant challenge. It involved extensive research and time, with some assistance from generative AI as well. Ultimately, the challenge was overcome by implementing a modular strategy for rerolling and selecting scoring options. The choices are evaluated using a hierarchical logic, based on their potential scores, ensuring the computer prioritizes the most advantageous moves.

Implementing the multiple Yahtzee logic & Joker Rule

Creating the logic for multiple Yahtzees was challenging. Especially validating its functionality was difficult because obtaining a Yahtzee is quite uncommon. I had to conduct numerous test runs before arriving at the current version. I implemented the Joker logic by breaking it down into specific sub-functions and organizing the code to manage the complex logic as smaller, more manageable tasks.

UNIMPLEMENTED FEATURES

Graphical User Interface (GUI)

The game is currently text-based with all user interactions and results managed through the console. Implementing a graphical user interface would enhance the game's interactivity and visual attractiveness, allowing players to click on dice and view score tracking visually.

Advanced Computer AI Strategy with Machine Learning or Probability

Currently, the computer player relies on a score based hierarchical decision making method to determine the optimal moves, but adopting a machine-learning model to improve its strategy could enhance its gameplay. This would require significantly more data and computational resources, which were beyond the scope of the current implementation.
Using probability will also will significantly enhance decision-making of the computer AI by allowing it to evaluate the likelihood of achieving certain scoring outcomes based on the current dice roll. But It will require more complex mathematical calculations in order to calculate these possible outcomes.

Leaderboard (High Score Tracking)

An additional potential feature could be a leaderboard for high scores that records players' scores throughout various sessions. To implement this feature, it would require reading from and writing to a file, enabling players to monitor their progress over time.