**Midterm Project Report**

**Skyler Atchison**

**2. Project Description**

2.1 Functional Specification

2.1.1 Functions Performed

NinetyNineAI will play the card game Ninety-Nine against two opponents. It will carry out this task as one of the players in a Ninety-Nine game interface. The two opponents may be human-controlled or simulated by a simple algorithm. If there is at least one human player, the human player(s) will be able to use an ASCII interface to select cards to play and see the current state of the board, such as the cards played by the opponents and the cards still held in hand.

In the case of a game using only simulated opponents, the user will play the role of a tester and will be able to instruct the program to run any number of games against the opponents with various settings. For example, the user will be able to select the specific search algorithms used by each computer player to compare their performance and merits.

2.1.2 Limitations and Restrictions

This project will not be concerned with the User Interface of the game, as its primary purpose is to test Artificial Intelligence algorithms and explore their strategies. However, as human testing is an important part of this process, a simple UI will be implemented.

Neural Networks and Machine Learning are also outside of the scope of this project. Instead, the decisions of the AI will be purely based on algorithmic search.

2.1.3 User Inputs

The program will take user input from the command line. When the program is launched, the user (a tester) will be prompted to select whether each opponent will be controlled by human input or one of a few algorithms. If there are no human players, The user will also be prompted to select how many games will be played. If there are human players, the user will be prompted to select a card to play on each of their turns.

2.1.4 Outputs

In a game involving human players, the system will print the user’s current hand and opponent’s cards played to the command line after each turn to allow the human player to make a move. In a simulated game, the computer will print the entire state of the game after every turn for analysis. If the computer has been instructed to print several games, it will only print summary statistics to the command line such as the percentage of games won by each opponent.

2.1.5 System Data Files

The program will have no need for a database or input files, although if the analysis setting is enabled it will write data to text or JSON files to allow human testers to review its performance against various opponents after the games have concluded.

2.2 Design Specification

2.2.1 System Data Flow Diagrams

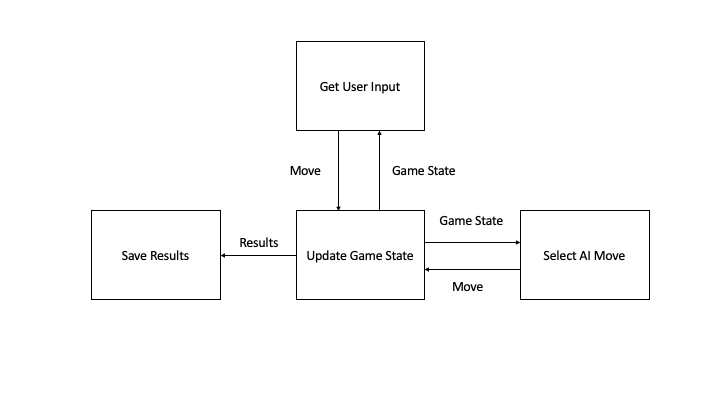


Figure 1: System Data Flow

The above data flow diagram shows the flow of data between the four key segments of the software.

2.2.2 System Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Data Item** | **Type** | **Description** |
| Game State | Python Object | An object representing the current state of the game, such as the cards in the player’s hands and the cards played so far. |
| Move | String | Parameters representing the chosen card to play |
| Results | JSON, TXT | Files with the results of the game for further analysis. |

2.2.3 Implementation Languages

The program will predominantly use Python as it is a standard language for AI research and comes with useful object-oriented functionality. If I decide to implement visuals, I will use the Pygame library as it is built for Python and is good for simple interfaces.