**Overview of procedural programming paradigm in GO**

In this code, I have used procedural programming, a paradigm that emphasises the use of sequential procedures or functions to perform tasks.

Each function is designed to carry out a specific task, and the overall logic of the program flows in a linear and orderly manner from one part to another. The state of the game is maintained using global variables, in this case, cells, which represent the game board. This style of programming is characteristic of a structured approach, focusing on the clear organisation of code into reusable blocks or functions.

**Gameplay flow:**

The process begins with initialising the game board. This setup is essential to prepare the game environment before actual gameplay commences. Once the board is set up, the main game loop is triggered, which is the heart of the gameplay. In this loop the game alternates between the human player and the AI opponent, ensuring each gets a turn, during each turn, the program validates the moves make by the players. This step it crucial to ensure that the move is legal and within the bounds of the game rules. After every move, the game checks to determine if there is a winner. this check involves looking at the board to see if any player has met the winning criteria. If a winner is found, or if the board is full leading to a tie, the game concludes.

After the conclusion of a game, the player is presented with an option to replay. This cycle continues based on the player’s desire to play again or to stop.

In essence, this code is a classic example of utilising fundamental features of the Go language, including the use of arrays for data storage, loops for iterating over data and controlling the flow of the game, conditional logic for decision making, handling user inputs.

All these elements are woven together in a procedural programming style, which is marked by its structured and systematic approach to problem solving.