Lifelines:

Game System: The overarching system that manages the game flow, including player turns and game state.

Game: Represents the current game's logic, including checks for game over conditions and player actions.

Player: Represents the individual player interacting with the game system.

Sequence of Events

Game Loop:

The game begins with the startGame() function, which initializes the game state.

The system enters a loop that continues as long as the game is not over ([!gameOver]).

Game State Checks:

The checkGame() method is called to determine if the game should continue or end.

If the game is over, the flow diverges into two branches:

If gameOver is false, the game continues.

If gameOver is true, the GameOver method is invoked, prompting the end of the game.

Player Turn:

The playerTurn() method is called to manage the actions of the current player.

If the player is in jail (isInJail), the JailTurn() method is executed, handling the specific rules for players who cannot move freely.

If the player is not in jail, the user menu is printed for the player to make a choice.

The turnOver variable is used to control the flow, determining whether the player's turn has ended or continues.

If turnOver is false, the loop continues, allowing the player to make further choices until they decide to end their turn.

User Menu Interaction:

The player is presented with options (choices) through the printUserMenu() method within a loop.

Depending on the player's choice:

Choice 1: Calls the rollDiceWithPrint() method, which simulates rolling the dice and prints the result. After, triggers the positionMove(Dice, Player) method, moving the player based on the dice result. Ends the player's turn by setting turnOver=true.

Choice 2: Calls the printBoard() method and prints the current game Board. Continues the player's turn by continue setting turnOver=false and let user make further decisions on player menu.

Choice 3: Displays information about all players using the allPlayer(Info) method. Continues the player's turn by continue setting turnOver=false and let user make further decisions on player menu.

Choice 4: Allows the player to check information on a specific player via specificPlayer(Info). Continues the player's turn by continue setting turnOver=false and let user make further decisions on player menu.

Choice 5: Queries the next player’s turn with queryNextPlayer(). Continues the player's turn by continue setting turnOver=false and let user make further decisions on player menu.

Choice 6: Saves the game state using saveGame(txt.file). Continues the player's turn by continue setting turnOver=false and let user make further decisions on player menu.

Choice 7: User chooses to return back to main menu, the returnBacktoMain() function is called, the game continues, and the turnover is set to true as the turn is over. A new game is waited to be started for the user at the main menu.

Choice 8: User chooses to exit the game and sets the turnOver=true and gameOver=true so the two loops ends, and the game is exited.

If none of the choices are entered like a 9 or letters, the game system will generate an ”invalid choice, please try again” message and the call printUserMenu() method again for the player to choose.

Game Over Handling:

If gameOver is set to true, the game reaches a conclusion, exiting the game loop and concluding the game session.