Python Software Project Planing Document

Julian Hoffmann, Christian Spreng

1 Game

- Sudoku

2 Role Distribution

- Julian: UI & Interaction & AI
- Christian: Gamelogic & AI

3 Implementation

Because of space constraints we replaced the keyword `self` *with* `s`.

3.1 Game Class

The core class representing a Sudoku game instance. Handles the game board, rules, and overall game flow. A gamesession contains the initial boardvalues, the placed values and the difficulty.

Method	Input	Output
`__init__(s, difficulty)`	Difficulty level.	None
`load_board(s, filename)`	Filename of the board.	None
`generate_board(s, difficulty)`	Difficulty level.	None
`is_valid(s, row, col, value)`	Row, column, and value.	`True` if move is valid.
`is_game_over(s)`	None	`True` if puzzle is solved.
`save_game(s, filename)`	Filename to save the game.	None
`get_board(s)`	None	The current game board.
`update_val(s, row, col, value)`	Row, column, and value.	None
`delete_all_values(s)`	None	None

3.2 Player Class

Represents a player of the game. Stores player information and high scores.

Method	Input	Output
`__init__(s, name, pwd)`	Name and password.	0 (Error), 1(Success), 2(New)
`get_highscore(s, diffic)`	Difficulty level.	User's high score for difficulty.
`update_score(s, diffic, time)`	Difficulty and time.	None.

3.3 UserInterface Class

Handles user interaction, displaying the game board, getting input, and providing feedback.

Method	Input	Output
`__init__(s, game)`	Reference to the `Game` object.	None
`display_board(s)`	None	None
`get_move(s)`	None	A tuple of row, column, value.
`display_message(s, msg)`	Message string.	None
`display_highscores(s)`	None	None
`choose_difficulty(s)`	None	The selected difficulty level.

3.4 HighscoreManager Class

Manages the overall high score list for all users.

Method	Input	Output
`load_scores(s, file)`	Filename of the highscore file.	None
`save_scores(s, file)`	Filename to save high scores.	None
`add_score(s, p, diffic, time)`	`Player` obj, difficulty level, and time.	None
`get_top_scores(s, diffic, n)`	Difficulty & n of top scores to retrieve.	High scores list.

3.5 AIPlayer Class

Implements an AI player that interacts with the game like a human.

Method	Input	Output
`__init__(s, game, diffic)`	Ref to `Game` obj and difficulty.	None
`make_move(s)`	None	Tuple of row, column, value of AI's move.

3.6 Additional Functions

Method	Input	Output
`main()`	None	None
`load_game(filename)`	Filename of the game file.	A `Game` object.
`instructions()`	None	None

3.7 Difficulty Levels

Difficulty	Description
Easy	Many given numbers, easy to solve.
Medium	Fewer given numbers, slightly more challenging.
Hard	Only a few given numbers, challenging.
Expert	Extremely few given numbers, very difficult.

3.8 UI Enhancements

- Usage of the `curses` for improved console display (colored cells, highlighting, . . .)