Project Structure

general

gamesprog.c

- the main program, initializes all the games and releases them at the end

gui.c

- manages all the gui related functions, main abstraction over SDL
- renders the gui control tree
- polls and dispatches user input events – clicks, keyboard
- create/release the gui_controls – panels, buttons and windows

game.c

- manages game structure and flow – players, game state (board instance), game ui panels.
- Interacts with the current game instance to generically control win/lose states, ai and human moves, and the side panel ui.

board.c

- represents a simple matrix abstraction
- mainly used to manage the logical game board

linked_list.c

- linked list abstraction used by gui control child list, minimax move list etc.

stack.c

- stack abstraction used to simplify the navigation model between the different panels

minimax.c

- the implementation of the Minimax algorithm, required for the AI player

6 different menus:

menu_main.c - creates the main menu 'the games program', new game, load game, quit

menu_newgame.c – the menu for choosing the specific game – tictactoe / connect4 / reversi

menu_player.c – choose the player option (AI VS AI, PLAYER vs PLAYER, AI vs PLAYER, PLAYER vs AI)

menu_game_panel.c – menu for the game, creates the game panel with all the buttons (save, quit, difficulty)

menu_difficulties.c – choose the difficulty level from one of the possible difficulties for each game

menu_saveload.c – generic ui to choose the slot to save or load your game

3 games:

Tictactoe.c, connect4.c, reversi.c – contain the specific implementation of the scoring function, possible moves and etc for each game