curses GUI

Chris Xiong

Goal

This semester, my goal was to make a normalized framework for a more user-friendly text-based interface using the Python `curses` library.

player.py

`player.py` serves as the main file for the GUI. It consists of an argument parser (currently used to input game/variant) and a game loop. The game loop works as follows:

- wait for next user input (keypress/mouse action)
- if the user's click corresponds to a valid move, make that move
- get position string generated by UI function
- draw position string to the screen
- get list of next moves from UWAPI
- for each move, get move string generated by UI function
- draw each move string

ui_{game}.py

The UI functions to generate position and move strings are contained within `ui_{game}.py` files, where `game` corresponds to the UWAPI game id (e.g. `ttt` for tic-tac-toe). The two functions that must be implemented are:

`position_string(pos)`
Takes in a UWAPI position string `pos` and returns a string to draw and an offset.

`move_string(move, pos, [index])`
Takes in a UWAPI move string `move`, a UWAPI position string `pos`, and an optional `index` indicating the move's index within the move list. Returns a string to draw and an offset.

If no UI file is found for a specific game, it will default to `ui_default.py`, which will try to create a UI for UWAPI-like strings or simply list out the strings as given if not compatible.