Blockis

Chris Xiong

Goal

This semester, I worked on the game Blokus. However, the original version of Blokus is far too large to solve. My goal was then to create and then solve different variations to make Blockus solvable.

Parameters

To make Blockus solvable, there are multiple different parameters that I needed to either reduce or adjust. The following parameters were changed to reduce game size:

a) Board size: 20x20 → 5x5

b) Available pieces: 21 (1 monomino, 1 domino, 2 trominoes/triominoes, 5 tetrominoes, 12 pentominoes) → 5 (5 tetrominoes)

Because the set of pieces is now only tetrominoes, I have changed the game name to Blockis (a combination of Blokus and Tetris).

The game board has been reduced drastically, meaning that the original Blokus rules of placing pieces with only corners touching heavily restricts placement. Thus, Blockis requires touching edges rather than vertices.
Another variation that I used when testing the game involved using only the L triomino, and allowing for unlimited reuse of pieces.

In the original Blokus, the winner is decided by who has placed the most pieces. For Blockis, the winner is the player who makes the last move.

GUI

In addition to solving Blockis with varying parameters, I also coded up a graphical text-based interface using the Python `curses` library. The GUI is controlled using `getch()` to allow for user feedback that feels more "live," compared to the usual type-and-enter approach used by other text-based GUIs.

For example, the board is refreshed after every keypress to allow the user to see visually where their piece will be placed, and adjust placement using the arrow keys. From the player's perspective, this feels a lot more intuitive than entering (x, y) coordinates, and so on.