MAYKABUK CASINO MACHINES, INC.

Congratulations! You've just joined the software development team at Maykabuk Casino Machines, Inc., the world's leading provider of gambling machines for casinos!

Maykabuk has a proud history of making high quality, enjoyable, fair, and profitable mechanical slot machines. However, in response to the increasing demand for computerized gambling machines that allow increasingly complex play, the company has branched out.

Your first assignment

Maykabuk is planning to sell a new computerized gambling machine called Pathfinder. You are a developer on the project.

Pathfinder rules

In Pathfinder, the player is presented with an initial board, consisting of an 8x8 grid of squares. There are 5 special squares: three "key" squares and two "bonus" squares. No special square ever occurs along the outside border of squares (i.e., in the first or last row or column).

On each turn, the player draws a card. A card consists either of a "path" or a "strike". The objective of the game is to build a continuous path from the top left square to the bottom right square before drawing accumulating three "strike" cards. The path must run through each special key square. If a player plays a path card on a bonus square, one accumulated strike is removed from the strike count (if there are any strikes at all).

A path card has between two and four exits. The first path card drawn must be played on the top left square. Every path card drawn for the rest of the game must extend the currently existing path on the board. If the path can no longer be extended, the player loses immediately.

The player starts the game by placing an initial wager. At any time, the player can surrender and receive back 25% of the initial wager. If the player loses (accumulates three strikes before reaching the bottom right corer), the entire wager is lost. If the player wins (reaches the bottom right corner before accumulating three strikes), the player receives 4x the initial wager.

The interface to the game should always indicate: the initial wager; the number of accumulated strikes; the current path; the special squares (whether or not the path has crossed those squares); and, the current state of the game (lost, won, in progress). While the game is in progress, the next card available to the player should be shown face down. When the player taps the card, it should be shown face up. If it is a path card, the player should be able to rotate the card and place it on the board. The interface should indicate which squares are available for extending the path (i.e., which squares can have a path card played on them).

Odds

The game designers tell you that the odds of winning any random game of Pathfinder that is optimally played by the player are 22.5%. However, these odds depend on placement of the special squares. Roughly speaking, the odds of winning are proportional to how close the key squares are to the top left to bottom right diagonal and to the how close the bonus squares are to the middle of that same diagonal. (Special squares that are far away from the diagonal, for example, near the top right or bottom left corners, require a longer winning path. Bonus squares that occur too early in the path building process have a higher chance of occurring before any strikes have been accumulated. Bonus squares that occur too late in the path may not be reached before accumulating three strikes.) Also, although the logic of the game is not very complicated, not every player plays optimally. In early mock-up testing with a physical board and cards, players won about 20% of the time.

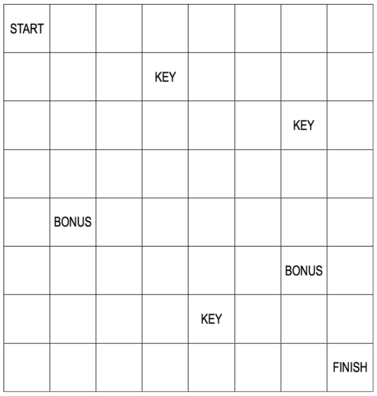
To be certified by the state gambling commission, a game company must prove that the published odds of winning are accurate. This is done by simulating 10,000 games, and showing that on any given simulation run, the number of wins is within 0.5% of the stated odds. (This is also important to the casinos buying the game; they want to be sure their calculated house advantage is correct!)

Your approach

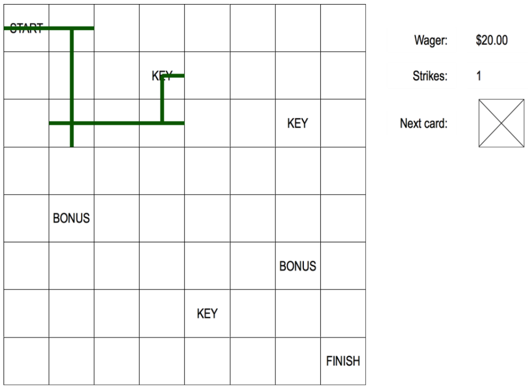
There are two important pieces of development that need to be done. Firstly, we need a “solver”. This should take a random initial empty board (i.e., randomly but legally placed key and bonus squares), and a deck of Pathfinder cards. The solver should iteratively draw and legally play a card until: 1) a third strike is accumulated (in which case, the game is marked lost); 2) no legal moves remain (in which case, the game is marked lost); or, 3) the end square is reached and all three key squares have been crossed (in which case, the game should be marked as won). Run trials with a deck that consists of 8 strike cards and 92 path cards (23 each of “straight line”, “corner”, “T-crossing” and “4-way crossing” cards) and verify if the odds are correct. Remember, odds verification depends on building a (nearly) optimal solver.

If you have time remaining, see if you can modify the construction of the deck to achieve a given winning percentage. You may have to constrain the types of decks that can be generated (e.g., by bounding the maximum number of cards in a deck or assuming something about the distribution of different types of path cards). This second piece of code should describe the deck construction necessary for your solver to achieve a given winning percentage.

Examples



An initial Pathfinder board.



Wager: $20.00

Strikes: 1

Next card:

A screen in the middle of a game.