Gameplay

Labyrinth is a game for two to four players. The game is set inside a tile maze that shifts with every turn, creating an unpredictable experience every game. Three objective tokens are placed next to the board, which represent items that can be found within the maze. The object of the game is to be the first to collect a set amount of tokens and escape to your home tile.

Each turn consists of two halves: maze movement and player movement. The first half, maze movement, is mandatory. The player takes the one extra piece, rotates it to a desired alignment, and inserts it into one of the rows or columns in the maze (numbered 0 to 6 top-to-bottom/left-to-right), sliding the row and kicking another tile out the other end. That other tile becomes the extra piece for the next player's turn.

The second half, player movement, is optional (though recommended). The player can move to any maze tile their piece has a path to follow to. Since the object of the game is token collection, it's in the player's best interests to use this half of the turn to get to, or near, the next token.

Setup

The server requires only one command to start:

- node index.js

The client is more complicated to set up, as it requires that the extension be added to Chrome.

- Navigate to chrome://flags
- Find and enable "Experimental Extension APIs"
- Restart chrome
- Navigate to chrome://extensions
- Check the Developer Mode checkbox at the top of the page
- Click "Load unpacked extension..." button, and navigate to and select the client directory. Chrome will see the manifest.json inside and do the rest.
- Click "Launch" to launch the client
- Once the client is launched, enter the server IP at the top of the client window, and click "Connect".

Networking

The game uses a single socket connection for each client in the game; the number of clients is determined when starting up the server. Both chat and game updates are sent over this connection. The game updates, which are sent after each turn, are encoded with a password to mark them as separate from the chat messages. Whenever a message is received, the server broadcasts it to all clients. When a client receives a message from the server, it determines whether the message is a game update; if not, it adds the message to the chat log. If it is a game update, the client calls an update function to show the new board.