

To: Matthias^2
From: Matthias
Subject: Santorini Proposal

Our friends have recently become obsessed with the classic game Santorini, and have expressed interest in allowing autonomous bots they've created to play in tournaments. Santorini is a game played by two players, bots in this case. Each bot will connect to our server through TCP and send it its name, and our server will send them each a Json description of the current state of the game. Each bot will send a few integers back to the server when it is told it's their turn.

First we will describe the protocol for a single game. The server will keep track of the entire gamestate, and make changes to it as it gets information from the bots. The game consists of a 6x6 board, where each field represents a building. At the beginning of the game, all the buildings have 0 floors. Each player gets to place two workers, taking turns. Each bot will connect to the server and send a name for itself as a string. The first bot the server receives a name from will become player 1. The server will begin the game, send each player a Json representation of the gamestate (as it will do after every change to the gamestate henceforth), and ask the first bot where it would like to place its first worker. The bot will respond over TCP with two integers between 0 and 5, in the format "x y". The server will update the gamestate, resend it to each bot, and ask the second bot for its first worker placement. Bot 2 will return in kind, and the server will ask Bot 1 for its second worker placement, and again ask Bot 2 for its final placement. The next phase of the game will begin: The server sends each player the gamestate, and asks Bot 1 for its move. Bot 1 responds with three integers in the format "w x y", with w being 1 or 2 to represent its workers, and x and y as before. The server will update the gamestate, check if the game is over, then ask Bot 1 where it would like to build. Bot 1 will respond with two integers in the format "x y". The server will check if Bot 2 has legal moves, and then ask Bot 2 for its move, beginning the cycle again. If at any point a bot attempts to place a worker in an illegal spot or build in an illegal spot, the gamestate will not be updated and the bot will be told to try again. Tournaments will be held Round Robin style.

The gamestate is represented in Json as an Array:

```
[status:string, turn:string, [[N,N,...][N,N,...]...], [bot1:string, [worker11x:N, worker11y:N, worker12x:N, worker12y:N]], [bot2:string, [worker21x:N, worker21y:N, worker22x:N, worker22y:N]]]
```

Where: N is one of {0, 1, 2, 3, 4, 5}

boti is the name of player i

status is one of {"placeWorker", "moveWorker", "build", "gameOver"}

turn is one of {bot1, bot2, "Draw"} ; "Draw" is reserved for status "gameOver".

The server will represent the Game using a Board as a 6x6 double array of Fields and two Players. Each Player has the server's connection to the bot it represents, and also has two Workers. Each Field has an int representing its floor height, and a boolean representing if there is a worker on it. A Player can placeWorker, moveWorker and build using the x and y coordinates of the intended Field. placeWorker and moveWorker will place or move a Worker on the Board by flipping the boolean of the corresponding Fields. These methods check if the input is valid given the current state of the Board, and return a boolean representing the success or failure of the move. The Game will get this boolean and ask for new input upon failure. After every moveWorker and build the Board will checkStatus if a player has won, and then if the next player has legal moves. After every successful change to the game state, the Game will sendGameState in the form of a JSONArray to each bot through the Players. A Game will begin with the setup method, in which each Player's placeWorker method will be called twice. Once that's finished, the Game's turn method will call a single Player's moveWorker and build methods, switch players, and repeat until checkStatus indicates that the game is over.

A Tournament will contain a list of bot connections, and arrange a schedule for each to play each other, creating a new Game. For N players, the Tournament will hold N-1 matches and keep track of each bot's score. A win is worth 1 point, a draw is .5 and a loss is worth 0. At the end the Tournament will report every bot's score.