

Maze

Official Maze Solving Rules

Maze solving race is a contest where a LEGO robot contestant attempts to be the fastest to travel from entrance to exit of a maze. The robot with the shortest count in seconds will win.

Robot Control: All robots are to act autonomously, that is to say, robots must operate without human control during a match. Robots must be completely self-contained and must receive no outside assistance.

Maze Details: The maze is an 8 x 8 matrix of 1x6 boards (see enclosed diagram)

Prohibited Items and Actions: A robot may not spray any material on the maze and nothing may be deposited in the maze. The method of wall sensing is at the discretion of the builder; however, the robot must not exert a force on any wall likely to cause damage. The robot must negotiate the maze; it must not jump over, climb, scratch, damage or destroy the walls of the maze.

Allowed Items and Actions: A robot may use various forms of detection to locate walls (IR, sonar, touch...) or for navigational purposes.

Programs: There are no restrictions as to what firmware or programming tools you use for your robot.

Maze Race Rules: The total time allowed per robot is called the performance time. The performance time cannot exceed 10 minutes. A robot may choose to not use its entire performance time. The time taken to travel from the entrance to the exit is called the 'run' time.

Performance Start: A performance starts when the builder indicates readiness to the judges.

Run Start: The robot may be placed anywhere on the start end of the playfield by the builder. The run time will start when the robot crosses the boundary between the start area and the first cell of the maze (that is to say, the run time clock will start when the robot actually enters the maze). The start area is about 12 inches long and as wide as the maze. The boundary wall of the maze is continuous except for the single entrance way. This will allow for wall-following style robots to be placed and activated along the wall outside of the maze. Remember, the clock does not start until the robot enters the maze.

Run End: A run ends when the robot crosses the boundary between the last cell of the maze and the end area(that is to say, the clock will stop when the robot actually exits the maze). The run may be canceled prematurely for any of the following reasons:

- The robot has become immobile.
- The robot is trapped in a loop.
- The robot has exited via the entrance of the maze.
- The performance time has expired.

Once a run has been cancelled, the run time clock is reset and the builder can take up to 1 minute to prepare for the next run. Cancelling a run does not stop the performance time clock.

Run Scoring: The run time is recorded for each successful run the robot makes. When the performance time for the robot expires its shortest run time achieved is used for scoring purposes.

Performance End: The performance is over when either 10 minutes has elapsed or the builder has chosen to retire and save the remainder of the performance time.

Maze Race Win: When all robots have used up their performance time, the race winner will be the robot with the fastest run time.

Repairs and Reprogramming: There is no restriction on repairs or reprogramming (aside from the limitation imposed by the performance time).

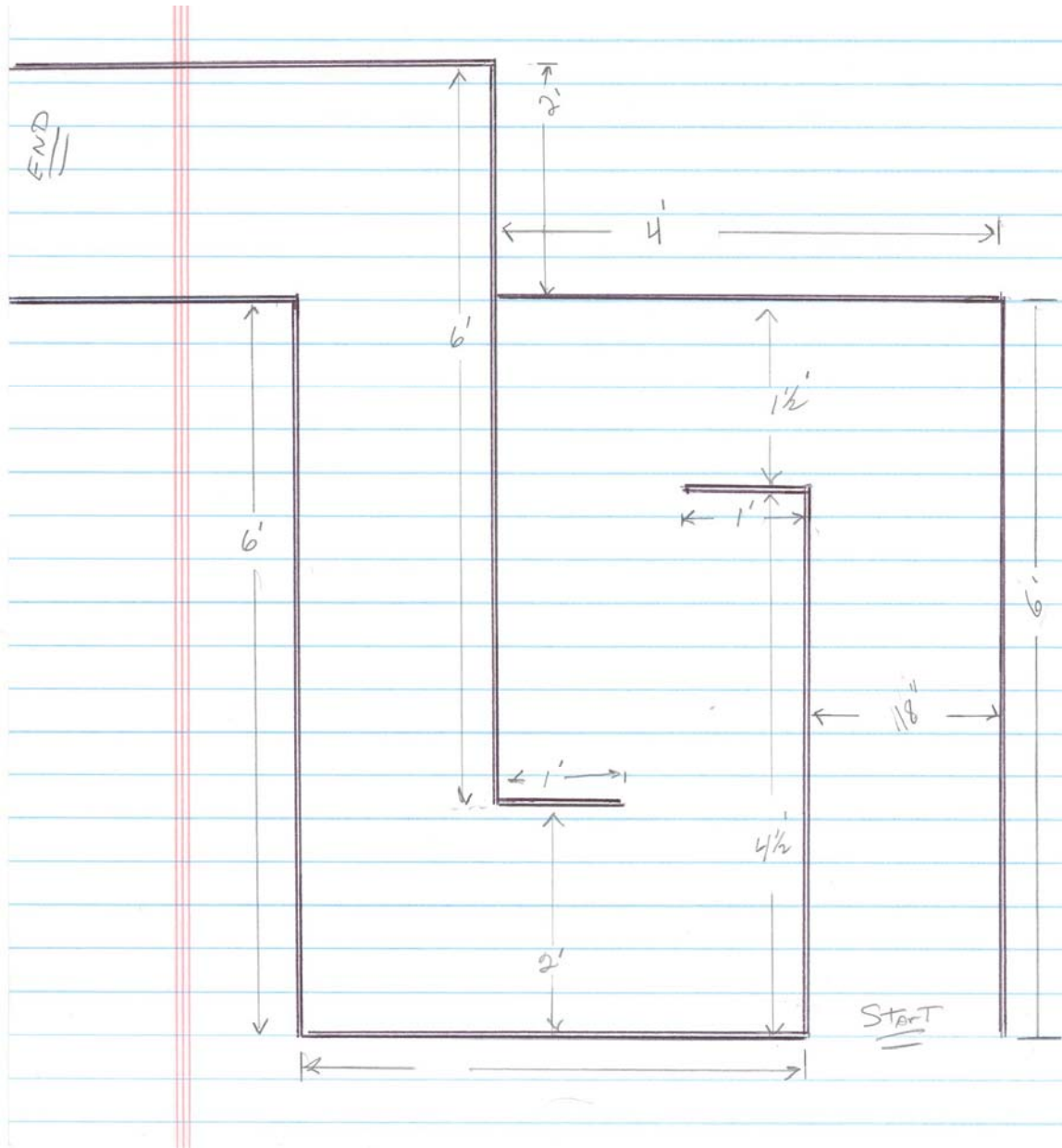
More Rules

1. Robots must be made with LEGO Mindstorms Robotics Invention System and be programmed with the Robolab programming environment.
2. Robots can only be touched prior to starting a run
3. The Robot must not damage the maze.
4. The robot cannot drop pieces of itself in the maze.
5. The robot must travel on the floor of the maze. Robots are not allowed to jump over walls.
6. Robots cannot touch the top of the maze walls

Scoring:

Score will be measured in seconds to complete the maze. Shortest run time wins the match.

A further bonus of -5 seconds of the final time will be granted to those robots that recognize (by flashing lights, playing a song, doing a dance) the black line at the finish



- 1) use 1x6 boards
- 2) all joints should be flush where robot travels