`cs2` Board File Specification

KISS Institute for Practical Robotics

December 30, 2013

1 Introduction

The simulator portion of `cs2` supports a simple "board" file format that allows customization of the simulation environment.

All commands follow a similar whitespace-delimited format:

⟨*command*⟩ ::= ⟨*function*⟩ ⟨*param1*⟩ ⟨*param2*⟩ ...

Some commands draw *objects* in the environment, while others mutate the board file's *state*.

For the remainder of this document, commands and their typed parameters will be bolded. For example, the statement **a b:string c:real** means that a function a takes two parameters – b of type string and c of type real. If a command is ≡ (equivalent) to another command, this means that the parameters and drawing function of the commands are identical.

Any line that begins with '#' in a board file is ignored.

2 Drawing Commands

Drawing commands create objects (lines, rectangles, etc.) in the environment. Different syntax is used for objects with different physical properties but identical drawing.

2.1 Walls

Walls are the simplest type of drawn object. Walls will effect simulated range sensors and touch sensors. Currently, there is no robot collision detection with walls.

- **line** x_1**:real** y_1**:real** x_2**:real** y_2**:real** — The line command draws a line in the environment from point (x_1, y_1) to (x_2, y_2).

2.2 Decorative

Decorative objects do not effect any sensor on a robot. They only serve as visual hints to the user.

- **dec-line** ≡ **line**
- **dec-rect** x**:real** y**:real** w**:real** h**:real** — Draws a rectangle starting at (x, y) with the width w and the height h

2.3 Tape

Tape is a special type of object that can be detected by analog reflectance sensors.

- **tape** ≡ **line**

3 State Commands

State commands effect the subsequent drawing commands that appear in a board file. For example, one could set the color of all subsequent lines to purple, draw several lines, then change the color again.

- **pen** c**:color** *width***:real** — Sets the current drawing "pen", which is roughly equivalent to the outline of an object. The color c can be a color name, color hex code, or even a CSS-style rgb tuple. The *width* is the drawn outline width.
- **brush** c**:color** — Sets the current drawing "brush", which is roughly equivalent to the color fill of an object. As with "pen", the color can be a multitude of different values.
- **set-units** u**:string** — Sets the units of any subsequent real number. The string u can be either "in" or "cm" (without quotes).
- **set-z** z**:real** — Sets the current drawing depth. For example, objects drawn at a z of -1.0 will appear above objects drawn at a z of 1.0. This allows fine-grained control of object layering.