

Becca version 0.4.0 // User's Manual

Brandon Rohrer

May 15, 2012

Contents

1	Get and run Becca	2
2	Write and run your first world	4
3	Share your world with other Becca users	5
4	Modify your agent code	6
5	Share your agent with other Becca users	7
6	Future contributions	8

Chapter 1

Get and run Becca

Each chapter in this guide is designed to help you do something specific with Becca. This chapter helps you to get a copy of Becca on your local machine and run it on some generic worlds.

Becca can be downloaded from www.openbecca.org or www.sandia.gov/rohrer. The latest bleeding edge version (for which any of this documentation may already be outdated) can be downloaded from the github repository at

<https://github.com/matt2000/becca>

Becca is intended to be runnable on any hardware platform. It relies on Python, NumPy, and Matplotlib. Becca 0.4.0 was developed on Python 2.7, NumPy 1.6.1, and matplotlib 1.2. The code makes use of at least one NumPy call, `count_nonzero()`, that is not supported in NumPy 1.5.x. Becca has been run several different platforms (Mac OS 10.6.8, Ubuntu 12.04 ¹) and IDEs (Eclipse, IDLE)².

¹From developer SeH: BECCA's dependency on NumPy 1.6 is provided by the latest Ubuntu 12.04 package (but not Ubuntu 11.10 which provides NumPy 1.5). A note on the website indicating this might be helpful to some users. Matplotlib is also provided, so everything seems to work fine on Ubuntu 12.04.

²Any notes on successes or incompatibilities would be very welcome at openbecca.org.

The `benchmark.py` module automatically runs Becca on a collections of worlds that are included with the download. Run it in a Python interpreter to get a report of Becca's performance in the worlds. `Benchmark.py` can be used both to compare Becca's speed on different computing platforms and, more importantly, to compare different variants of Becca against each other.

The worlds in `benchmark.py` are intended to be simple, but to test Becca's fundamental learning capabilities. Becca is an agent, in the sense that it makes decisions in order to achieve a goal, but it is intended for use in many different settings, each of which is referred to as a world. The worlds tested in `benchmark.py` include one and two dimensional grid worlds and one and two dimensional visual worlds. The reward provided by each world gives motivation to Becca to behave in certain ways. When it behaves correctly, it maximizes its reward. This is Becca's one and only goal.

Chapter 2

Write and run your first world

world code overview

Chapter 3

Share your world with other Becca users

Chapter 4

Modify your agent code

Chapter 5

Share your agent with other Becca users

Chapter 6

Future contributions

There is a list of changes that will potentially improve Becca. Anyone hoping to contribute to the Becca core can use this as a suggested list of “to do” items. Here are some of the items as of this version.

- Create an extendable version of numpy arrays. There are many variables and structures in Becca that grow in size over time. A data structure that allows one- and two-dimensional numpy arrays to grow inexpensively may speed up its operation. This may be done by creating a class that uses a zero-padded numpy array that is larger than the array it is representing. As needed, it could grow in jumps of a constant factor (say, of 2) and track which portions of it are actually being used to represent the growing array. It’s not space efficient, but may be much faster.