

Useful for Baxter:

If something isn't working on Baxter try running the following code in a terminal
`source /opt/ros/groovy/setup.bash`

Baxter moveit:

To launch move it use the following:

Run this code in primary shell first:

```
roslaunch baxter_interface joint_trajectory_action_server.py
```

IN a fresh terminal with the correct shell use:

```
roslaunch baxter_moveit_config demo_baxter.launch
```

Connect 4

Start the vision from an [RSDK terminal session](#), making sure to define which hand camera we will use, ex:

```
$ roslaunch connect_four connect_vision.py -l right
```

Resulting in the connect four vision gui being launched

then

Running a New Game

After starting the vision as shown above, from another [RSDK terminal session](#), start a new Connect Four game, defining which limb you will be using, as well as the color of Baxter's game pieces, ex:

```
# We will use the right arm, where Baxter is playing with the red pieces
$ roslaunch connect_four connect_game.py -l right -c red
```

There are four limb configurations the player will need to specify, the camera location (where the game board can be observed through the hand camera), the game piece pick location (from the feeder), the left most board location (from Baxter's perspective), and the right most board location.

These locations will be saved to a file for convenience for future games.

You will be prompted to use these previously specified locations:

```
Would you like to use the previously found pick and place locations (y/n)?
```

Specifying 'y' will jump you to the start of the game.

Specifying 'n' will allow you to record the required locations for your setup.

Move right arm into a location allowing the camera to see the board - press Circle button to confirm

The circle button here, refers to the circle button on the cuff on the chosen limb.

Likewise, you will specify the game piece pick location with the circle button. The gripper will grasp your game piece.

You will then be prompted to:

Move Gripper into left most drop slot - press Dash button to record

Move the limb so that the held game piece is just inside the left most (from Baxter's perspective) drop slot, where if released would result in a successful placement. Press the dash button.

Do the same for the right most drop slot.

The manipulation portion will then be able to create/segment a vector across the top of the game board identifying the other place locations. It will also use the IK solver service to find an appropriate approach for the piece pick location, and each of the place locations 5cm above in the z-axis.

The hand camera will then go back to the camera observation location, prompting you to validate the ROI. This simply allows you to make any final adjustments so that the 'Board State' window is appropriately populated.

You will then enter your name and a difficulty level (1-4). This difficulty level describes the depth of recursion in which Baxter will explore the search tree in the minimax implementation. 1 being the easiest, 4 being the hardest. Please feel free to make this demo better by submitting pulls with improvements in Baxter's gameplay!!

Lastly, you will choose who will go first.

That's it! Game on!

Baxter example programmes::

<https://github.com/RethinkRobotics/sdk-docs/wiki/Example-Programs>

Baxter calibration:

Usage

Remove the grippers from both of Baxter's arms.

Start calibrate from an RSDK terminal session for the left limb using:

```
$ rosrun baxter_tools calibrate_arm.py -l left
```

Start calibrate from an RSDK terminal session for the right limb using:

```
$ rosrun baxter_tools calibrate_arm.py -l right
```