CS5300 - Final Project Report

For our final project for Artificial Neural Systems, we decided to try to tackle the Monty Hall Paradox. The Monty Hall Paradox is a game involving three doors, one of which is the "correct" door to choose. After picking a first door, you must decide to keep your door, or switch to another one for the second round. In order to ensure a better chance of winning the game, one should switch doors for the second round. We have implemented this paradox in PyBrain, and have experimented with a variety of parameters in an attempt to achieve a higher success rate. This report will document our findings.

Initially, we implemented the Monty Hall Paradox in Python, with no PyBrain machine learning. The program chose doors at random, but we experimented with several different methods of making the decision of whether or not to switch. Here are our results over 100,000 iterations:

	Wins	Losses	Percentage
Always switch	66,671	33,329	66.6%
Never switch	33,385	66,615	33.3%
Randomly switch	50,172	49,828	50.1%

The results of this experiment are surprisingly straightforward. It is clear that switching your second choice guarantees a higher rate of success than keeping your current choice. Can we teach a machine to learn to achieve the same results?

For our second phase, we implemented the Monty Hall Paradox in a PyBrain environment, using similar parameters. In order to set up a PyBrain environment for learning, we were required to create the following:

1. **Environment:** used to set up the Monty Hall Paradox game, and retrieve whether the user won or lost a given round
2. **Task:** used to run the Monty Hall Paradox game, and determine a positive reward for correct choices, or a negative reward for incorrect choices
3. **Agent:** used to learn from the results of the Monty Hall task, and to adjust the program's choice preference in future runs

After implementing the aforementioned environment, we began to experiment using our PyBrain environment. The learning agent required the most experimentation, because it requires the following configuration:

1. **ActionValueTable:** a table of available states (in our case, 3, for 3 doors), and actions which can be performed (2, stay or switch door)
2. **Q-Learner:** an algorithm used in machine learning
3. **Explorer:** we use the EpsilonGreedyExplorer, which will "lock on" to a correct choice very quickly once it has been found

Finally, we were able to run the program for 100,000 iterations using both learning (via agent.learn()) and no learning (by removing agent.learn()). Here are our results:

	Wins	Losses	Percentage
No machine learning	50,063	49,937	50.0%
Machine learning	66,691	33,309	66.6%

As expected, without machine learning, the program is essentially randomly guessing which door to pick. By implementing machine learning, we were able to teach the machine to switch its second choice in order to improve its chances of winning. The end result is nearly identical to the "always switch" approach in the simple Python program!

Though PyBrain was initially frustrating to set up, the benefits of its use are quite clear in this program. We were able to teach our program to attain a high rate of success through use of reinforcement learning. Perhaps it would be possible to tweak the experiment parameters to achieve an even higher success rate, but we were pleased that we were able to match the results of the "always switch" approach. This experiment has only touched the surface of what reinforcement learning is capable of doing, but the knowledge we have gained from it will surely be of benefit in our future careers as computer scientists.

```
[zsh|matt@servnerr-2]:~/git/montyhall (master) python montyhall.py
Monty Hall Paradox - CS5300 Final Project - Andy Ladd, Matt Layher
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[main] running 100000 iterations...
[main] iterations complete!
[main] results:
       wins:   50079
       losses: 49921
[zsh|matt@servnerr-2]:~/git/montyhall (master) python pybrain-monty.py
Monty Hall Paradox - CS5300 Final Project - Andy Ladd, Matt Layher
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
[main] running 100000 iterations using PyBrain...
[main] iterations complete!
[main] results:
       wins:    66675
       losses:  33325
[zsh|matt@servnerr-2]:~/git/montyhall (master)
```

Demonstration of Python program and Python+PyBrain program