# Assignment 4: Probability Agents in Sokoban

Due on: Apr 4, 2021

## Introduction:

Welcome back to your last venture through Sokoban as this time we need your help to not solve puzzles but instead clean up the warehouse from mice! While you were away learning constraint satisfaction problems mice have infested the warehouse and its up to you to clean them up. There is a full proof method of getting all of them and its through the use of probability agents! You will work through creating two probability agents for this assignment, the MarkovAgent and the ParticleAgent.

How do we detect mice you ask? Well we do so through listening (sensing) via echoes the mice give off. When mice move they make noise and through these noises we will gather information (probabilities) and make actions accordly. We're relying on you to help so let's get started!

## Installation:

In order to solve the assignments you will need to install Python 3 along with the Tkinter library (GUI) and TkThread library (for TKinter multithreading). You can install Python 3 through your terminal if you're on unix or through an installer found here `https://www.python.org/downloads/release/python-374/`. To install the Tkinter and TkThread libraries we recommend downloading and using python pip:

- pip install tkinter
- pip install tkthread

## Files to Edit:

- particle_grid.py

- particle_agent.py
- markov_agent.py

The agent files can be found in the agents directory while particle_grid.py is in the probability directory.

## Question 1:

Let's start off simple with writing the *listen* method found in *markov_ agent.py*. The MarkovAgent focuses solely on utilizing the *EchoGrid* to help build a set of thoughts for the player agent. These thoughts are comprised of different probabilities which relate to positions on the grid and how likely it is for a mouse to be in a position. Use the distribution given by the EchoGrid combined with pre-existing thoughts to build an accurate understanding of where the mice are on the map at a **given moment in time**. You will find the concept of **Hidden Markov Models and Variable Elimination** helpful here.

## Question 2:

Now you will implement the *predict* method in *markov_ agent.py*. This method focuses on chasing down one mouse at a time by building up our set of thoughts to allow prediction of a single mouse's movement. We chase the same mouse until it is caught then move onto the next until all mice are caught. You will need to combine the movement distribution of the mouse and the distribution given by the *EchoGrid* to build an effective set of thoughts. The predict method can be considered a distribution **over time** or a **time elapse** approach.

## Question 3:

Before we can proceed to the *ParticleAgent* we need to first understand how particle filtering will affect our thoughts. So first you will be implementing the *reset* method in *particle_ grid.py*. For this method simply reinitialize the particle distribution such that the particles are evenly split among the positions on the grid (i.e. uniform distribution). Don't worry, the number of particles given will always be evenly divisible by the number of legal positions on the grid.

## Question 4:

The next step is to implement the *reweight_particles* method also found in *paricle_grid.py*. This is the essential method of how particle filtering works and can be tricky to understand. Here you will want to sample the given distribution and then update the probabilities of the positions based off the sampling. This will reweight the particles allowing us to further strengthen our thoughts distribution for the *ParticleAgent*. **Hint: Consider the case where all particles lie on the same square and how you should go about handling it.**

## Question 5:

Now we are ready to work on the ParticleAgent. First you will implement the *listen* method found in *particle_agent.py*. Unlike the *MarkovAgent*, the *ParticleAgent* utilizes both a *ParticleGrid* and an *EchoGrid* to help build up the agent's thoughts. Similar to the *MarkovAgent's* listen method you will focus on building thoughts for a **given moment in time**. Don't forget to reweight the particles as you build your thought distribution!

## Question 6:

For the final question you will write the *predict* method for the *ParticleAgent* found in *particle_agent.py*. This is done with a similar approach as in the *MarkovAgent* as we once again focus on chasing one mouse at a time, except now we use particles as well as echoes to track down the mouse. Remember to utilize the movement distribution to help build the thought distribution for the agent. Also recall the predict method is building a distribution **over time** or in a **time elapse** approach.

## Testing:

To test your solutions we provide a partial autograder. This autograder **should not** be treated as your final mark as there will be more tests involved during actual marking. To run the autograder simply type 'python autograder.py' or 'python3 autograder.py' (depending on how your environment variables are setup).

## Inspiration and References:

The structure of this assignment and some questions are inspired from those found in the assignments provided to students in CSC384 during 2018 (last year). This assignment follows the course material (specifically the lecture slides) of CSC384 during 2018. As the CSC384 course and assignments from 2018 reference concepts taught to students in a similar AI course found at Berkeley, I find it important to include Berkeley as a reference for these assignments as well. To see more information on Berkeley's AI course visit `http://ai.berkeley.edu`.