### What is the problem you are attempting to solve?

I am intending to incorporate reinforced learning within an agent-based model.

### How is your solution valuable?

Reinforced learning within an agent-based model can be used to help develop models that can explain and predict human-like responses to given situations through a collective learning process.

### What is your data source and how will you access it?

My data will be self-generated but will be based on data from the John Jay & ARTIS Transnational Terrorism Database, and the Economics of National Security: Data Set Index, that was collected on terrorist networks.

The link for these databases are as follows:

1. [John Jay & ARTIS Transnational Terrorism Database](http://doitapps.jjay.cuny.edu/jjatt/attributes.php)

2. [Economics of National Security: Data Set Index](http://www.nber.org/ens/feldstein/ENSA_Dataset_BlueTOC.pdf)

### What techniques from the course do you anticipate using?

I will be using deep learning convolution neural networks built with TensorFlow to perform reinforced q-learning within an agent-based model.

### What do you anticipate to be the biggest challenge you'll face?

Determining what actions an agent will be able to take, and how to define the reward/penalty in the q-learning structure for each action.

## Explanation of the model:

### 1. What is an agent-based model?

An agent-based model is simply a model that is populated with agents, to study how agents interact with each other and their environment. But how does it actually work? An environment is computationally defined and created within the system. The environment can then move forward in time. Within the environment, an agent or agents can be defined and created as well. The agents can be of the exact same type, or different types of agents can be constructed and placed within the environment. Each agent will also step forward through time with the environment. At each step, a predefined action can take place for both the agents and the environment.

[https://youtu.be/bjjoHji8KUQ](https://www.youtube.com/watch?v=bjjoHji8KUQ&t=350s) (5:55)

### 2. What is q-learning?

Q-learning is a reinforcement learning system where a table of q-values is generated for an agent. The columns of the table are the actions available to the agent, and the rows are all the possible states the agent can be in. The agent will use the highest q-value for it's given state to determine which action to take, in order to accomplish a goal. If the goal is achieved, a reward is given; if the goal is not achieved, a penalty is given. Either given a reward or penalty, the action/state pair (and it's corresponding q-value) are then updated to reflect the consequence. However, in order to combat the model becoming stuck within a local maxima instead of achieving a global maxima, the system is built using epsilon-greedy exploration versus exploitation of the system. Epsilon is initially set at a high value, so the agent prioritizes exploring the system, by choosing a random action instead of using the q-value. Epsilon is then deteriorated over time until the agent begins using the q-values. This allows for the q-table to truly evolve and learn the system. However, epsilon is never reduced to 0, so that the agent still has a possibility of exploring the system to achieve a global maxima.

<https://youtu.be/qhRNvCVVJaA>

### 3. How can q-learning in an agent-based system work?

Agents within an agent-based model can not communicate and coordinate amongst themselves. However, if each agent is allowed to share one global q-table, then as each agent performs an action, the agents as a whole will begin to learn. This is useful within an agent-based model in that it more accurately demonstrates humanity's ability to learn from history and from others' actions.

<https://youtu.be/CvL-KV3IBcM>

## Explanation of my specific model:

My model is titled "Prediction of a Terrorist Network Actions and Response to Military Presence." The goal of the model is to successfully produce an agent-based model that uses 2 global reinforced learning networks, that will share the action and reward knowledge among same-type agents that cannot communicate directly with each other. I will be using an agent-based modeling system called Mesa. Mesa was developed to be used within Python code. My model will include 3 basic types of agents: a civilian, a terrorist, and a military agent. The environment in which they will be generated will be based around a geographical location. The user of the model will be able to choose several different aspects of the model, such as how many cities to use, the density of terrorists within a region, and how large of a military to deploy to the region. The terrorist agent's goal is to maximize the control over a given region, which may be determined by the equation: # of Terrorist Agents - # of Civilian Agents - # of Military Agents. The military agent's goal will be the opposite, maximize the civilian control over the region, which may be determined by: # of Civilian Agents + (# of Military Agents / 2) - # of Terrorist Agents. Each terrorist agent will have approximately 4-6 actions they are able to take, while each military agent will have approximately 3-5 actions. Civilian agents will not have any direct action, but will have the possibility of converting into a terrorist agent. Each agent will have certain characteristics that will be combined with global characteristics to determine what the agent’s current state is, which will be fed to the neural networks. TensorFlow will be employed to create 2 CNN which will determine the current q-value for the given state, and the next q-value. This will help the system to prioritize future rewards versus immediate rewards.

The current variables that my model will use vary from one type of agent to another.

Such as, the terrorist agents have are:

* **Wounded:**
  + This is a program variable that tell the model whether the agent is wounded or not. If the agent is wounded, do not use agent’s step() until wounded is false. Binary: True or False
* **Age:**
  + Current age of the agent. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Binary: 0-Male or 1-Female.
* **Gender:**
  + The gender of the agent. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Continuous: Range 16 – 55.
* **Religion:**
  + The religion of the agent. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Binary: 0 – Islam, 1-Other.
* **Agressive Behavior (agr\_bhv):**
  + The agent’s aggressive behavior. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Continuous: 0-Complete Pacifist – 1-Complete Agression
* **Religious Fanaticism (rel\_fnt):**
  + The agent’s fanatical adherence to their religion. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Continuous: 0-No fanaticism– 1-Complete fanaticism
* **Religious Conversion Ability (rel\_conv):**
  + The agent’s ability to convert other non-terrorist agents to become terrorist agents. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Continuous: 0-Unable to convert – 1-Master converter
* **Hatred Toward Foreigners (hst\_twd\_for):**
  + The agent’s attitude toward foreigners being within the region. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Continuous: 0-Love of foreigners – 1-Complete hatred
* **Level of Recent Activity (lvl\_rct\_act):**
  + How active the agent has been recently within the terrorist network. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Continuous: 0-No activity or long period of inactivity – 1-Constant activity or very recent activity
* **Current Aggression Level (crt\_agr\_lvl):**
  + The agent’s current desire to take a violent action. This is used in a Random Forest Regression model to determine the agent’s probability of threat. Continuous: 0-Unwilling to take action – 1-Complete willingness to take action
* **Probability of Threat (prob\_threat):**
  + The agent’s current probability of being a threat and taking a violent action. Continuous: 0-No threat – 1-Guaranteed threat
* **Type:**
  + This is a program variable to differentiate the types of agents.
* **State:**
  + This is a program variable of the current snapshot of the agent and the environment. This is used fed to 2 CNN to determine the current q-value and the next q-value. This is a list containing the agent’s other variables, minus the wounded, plus the current terror score, the current civilian score, the x position of the agent, the y position of the agent, the current count of terrorist agents, the current count of civilian agents, and the current count of military agents.

The military agents have considerable less variables, as the current model only looks at agression and response from terrorist agents. The variables are:

* **Wounded:**
  + This is a program variable that tell the model whether the agent is wounded or not. If the agent is wounded, do not use agent’s step() until wounded is false. Binary: True or False
* **State:**
  + This is a program variable of the current snapshot of the agent and the environment. This is used fed to 2 CNN to determine the current q-value and the next q-value. This is a list containing the current terror score, the x position of the agent, the y position of the agent the current civilian score, the current count of terrorist agents, the current count of civilian agents, and the current count of military agents.