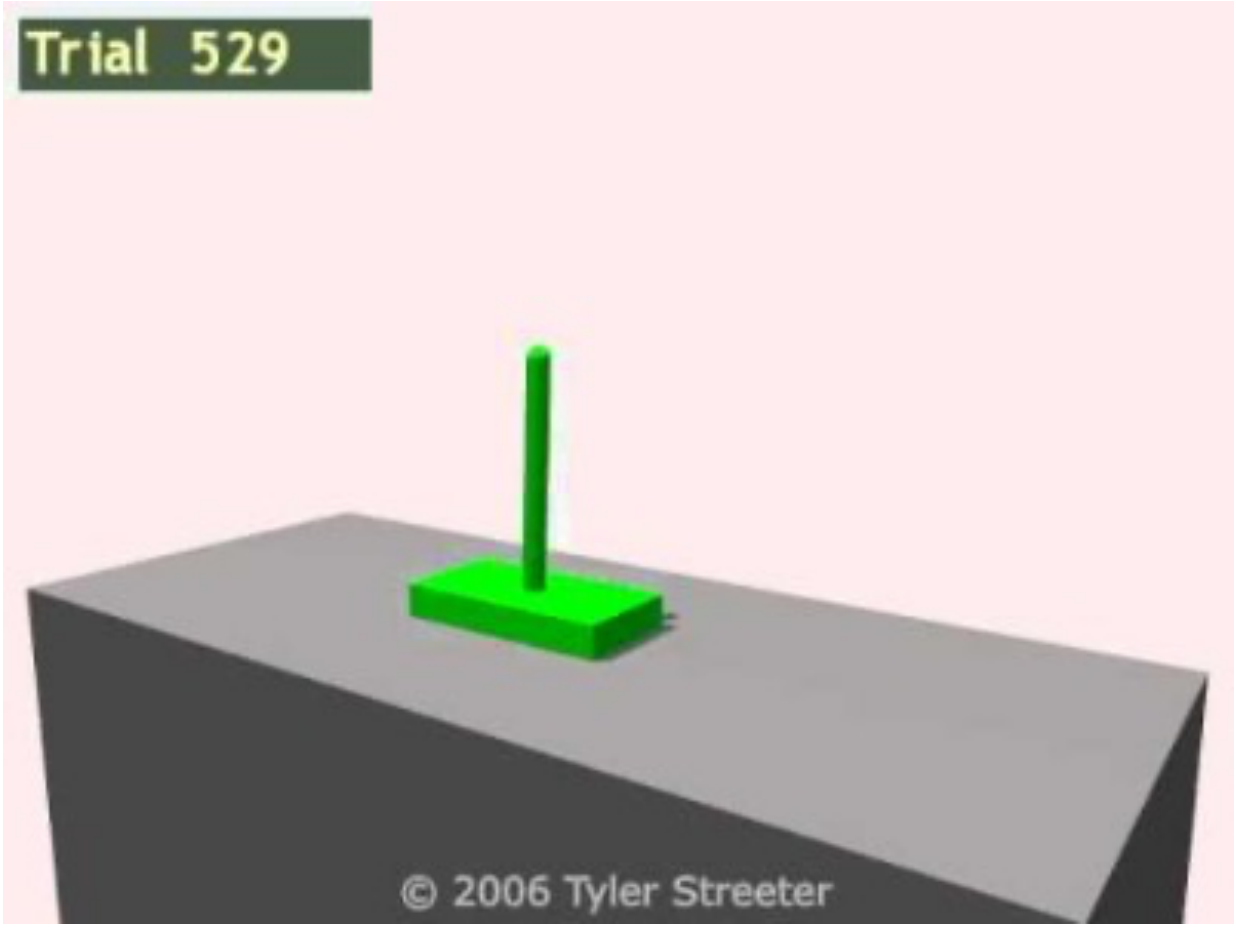
Module A - Reinforcement Learning

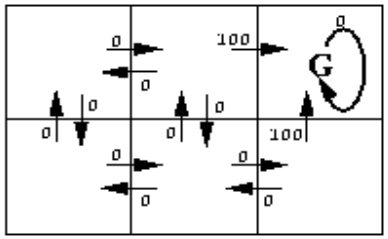
Inverted Pendulum Problem - states, actions, rewards?



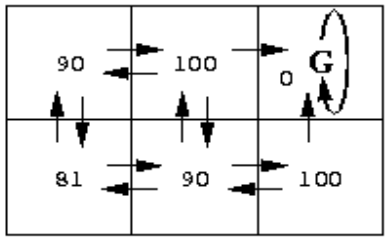
Plan

- Reinforcement learning setting
  - The need for RL
  - The RL setting and types of RL
  - What behavior should we try to learn
- Q-learning
  - V-learning to motivate the need for Q-learning
  - The Q-learning algorithm
    - Deterministic worlds
    - Non-deterministic worlds
  - Proof of convergence and any-time behavior
  - Limitations and how to overcome them
- The need for Deep Q-Learning
  - If I start talking about convolutions, backpropagation, mini-batch RELU units etc. are people familiar with those concepts?
- **For each class I need a scribe to put up notes on things missing from my notes!**

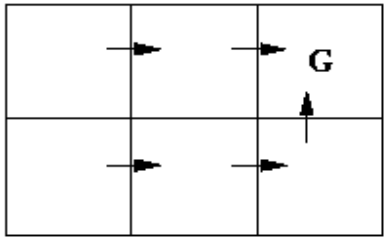
Grid World
γ=0.9



$r(s, a)$ (immediate reward) values



$V^*(s)$ values



One optimal policy

Two Steps:

Learn V values
Then the optimal policy

But this makes one big Assumption?
Need to know state transition operator to "look" ahead

$\hat{Q}$ Learning Algorithm – Lets Go Over Simple Example

For each s, a initialize table entry $\hat{Q}(s, a) \leftarrow 0$

Observe current state s

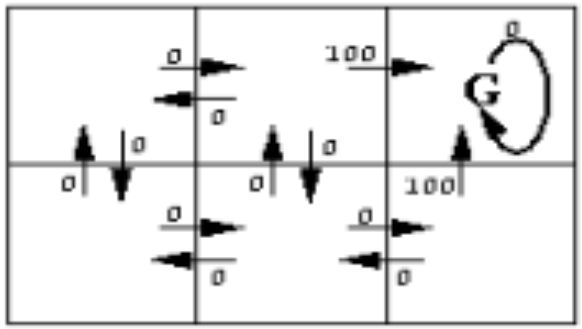
Do forever:

- Select an action a and execute it
- Receive immediate reward r
- Observe the new state s'
- Update the table entry for $\hat{Q}(s, a)$ as follows:

$$\hat{Q}(s, a) \leftarrow r + \gamma \max_{a'} \hat{Q}(s', a')$$

- $s \leftarrow s'$

Rewards: γ=0.9



Lets run this algorithm for this game

Think about limitations of algorithm