

# Trabalho I

## Inteligência Artificial

### 1 Descrição

- O trabalho consiste em desenvolver um controlador para o pêndulo invertido usando Aprendizado por Reforço.

### 2 Regras

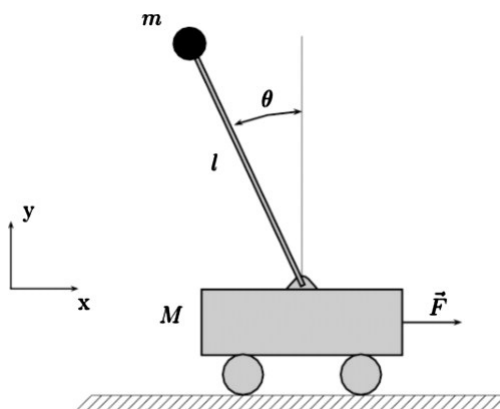
- O trabalho deve ser feito em grupos de até 2 alunos
- O trabalho deve ser entregue no Moodle até 01/05
- O trabalho deve ser apresentado em aula no dia 02/05
- Trabalhos podem ser entregues atrasados até 04/05, mas ficam sem nota de apresentação.
- Trabalhos muito semelhantes, onde fique evidente que houve cópia, estão sujeitos a serem considerados cola. Se isto acontecer ambos os trabalhos terão nota Zero.
- Todos os materiais entregues devem ter o nome dos componentes do grupo.
- O trabalho deve ser feito em python
- Devem ser entregue:
  - Uma descrição do trabalho de vocês, descrevendo o que foi feito e os resultados encontrados.
  - Um arquivo .py que execute o sistema
  - Caso algum pacote extra tenha sido usado, isto deve estar indicado num README, com instruções de instalação. Além disto este arquivo deve conter qualquer instrução de execução pertinente.
  - Os códigos devem ser documentados

### 3 Critérios de Correção

- Trabalhos que não seguirem as regras anteriores tiram Zero.
- A nota do trabalho levará em conta:
  - \* Execução: O trabalho tem que compilar e executar
  - \* Resultado: Se o trabalho atingiu os objetivos
  - \* Descrição: A descrição do que vocês fizeram e dos resultados alcançados. **"O que não foi dito não foi realizado!"**

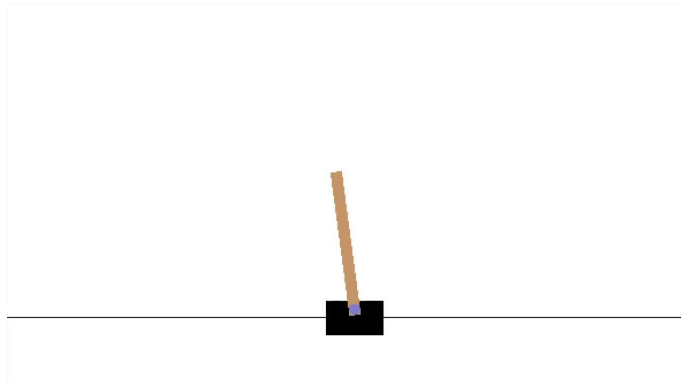
### 4 Problema

- O problema do pêndulo invertido consiste de um pino ligado a um carro por um eixo. Ver imagem abaixo.



O pino pode se mover em duas direções (esquerda/direita) e não pode ser diretamente controlado pelo usuário. O carro também se move em duas direções (esquerda/direita) e pode ser controlado pelo usuário. O problema está em mover o carro de forma a balancear o pino.

- Para fazer isto vocês vão usar o simulador para o pêndulo invertido do OpenIA Gym em python. Este simulador implementa o pêndulo e permite a visualização da execução do seu programa.



- Junto com esta descrição é disponibilizado um código python (`cart-pole.py`) que executa o simulador usando ações aleatórias.
- Vocês terão que modificar este código incluindo:
  - Código para ler e salvar os valores dos estados em um arquivo
  - Código para armazenar os estados durante a execução
  - Código para escolher as ações a partir do estado atual
  - Código para atualizar os valores dos estados
- Cuidado: Os valores que representam os estados são números reais, isto pode gerar infinitos estados. É importante que vocês discretizem os valores para gerar menos estados, mesmo assim podem ser muitos estados.

## 5 Pacotes necessários

Para rodar o OpenAI Gym você deve instalar os seguintes pacotes:

- python 2.7
- numpy
- scipy
- matplotlib
- tensorflow
- gym

## 6 OpenIA Gym

- O OpenIA Gym implementa uma série de ambientes para testar algoritmos de Inteligência Artificial. O Open IA Gym é feito em python e possui uma interface simples para iteração com os ambientes.
- O ambiente que vocês vão usar é o 'CartPole-v0'.

Este ambiente é criado pelo comando `env = gym.make('CartPole-v0')`, (ver `cartpolev0.py`)

- Para efetuar ações deve-se usar o comando `observation, reward, done, info = env.step(action)`, onde `action` é um valor inteiro dependendo do ambiente.

Este mesmo comando devolve as informações necessárias para atualizar os estados (`observation` e `reward`) e se a simulação acabou ou não.

- Veja o site <https://github.com/openai/gym/wiki/CartPole-v0> para uma descrição dos valores da observação e das ações possíveis.

Qualquer dúvida entre em contato comigo. Não deixe para a última hora.

Bom Trabalho