Pontifícia Universidade Católica
do Rio de Janeiro

INF 1771 – Inteligência Artificial

Aula 21 – Inteligência Artificial em Jogos

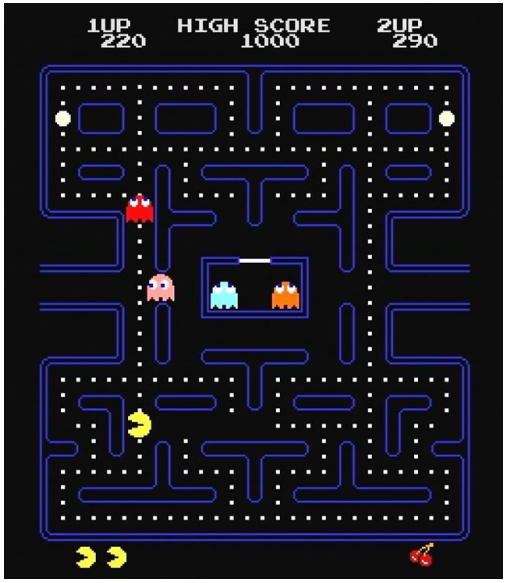
Edirlei Soares de Lima
<elima@inf.puc-rio.br>

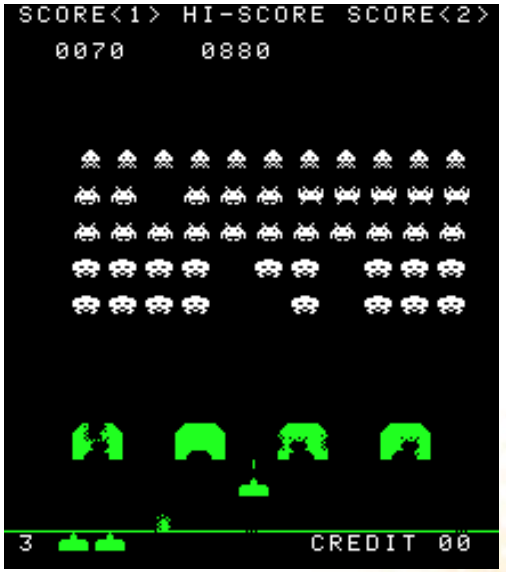
visionlab
visualization laboratory

Introdução

- **Surgiu com a criação dos primeiros jogos (Pac-Man, Space Invaders...).**

- **No inicio:**
  - Regras simples, sequencias pré-definidas de ações, tomada de decisão aleatória.





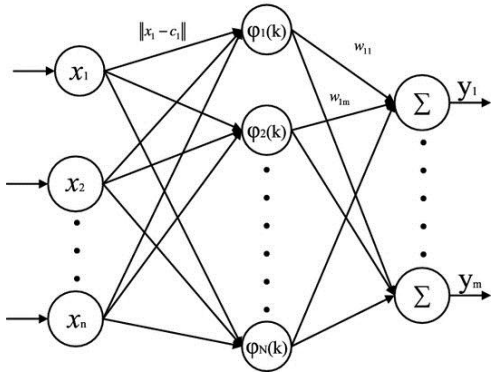
Introdução

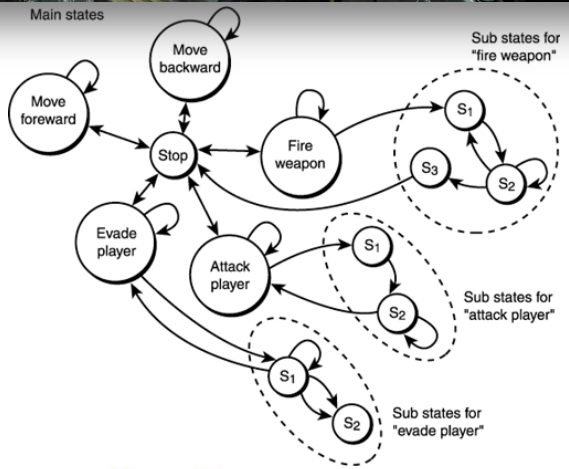
- **Atualmente:**
  - Melhoras em **gráficos e som** são pouco notadas.
  - Ambiente visual já está **suficientemente complexo**.
  - Foco agora está no **gameplay**, na **jogabilidade** e na **inteligência artificial**.
  - Personagens devem ser tão bons quanto **oponentes humanos**.

visionlab
visualization laboratory

Introdução

- **Industria vs Academic/Research**





visionlab
visualization laboratory

Ilusão de Inteligência

- Não se espera criar unidades inteligentes, mas sim criar uma "**ilusão de inteligência**".
- Em outras palavras, espera-se criar comportamentos que imitem comportamentos humanos.
- Roubar ou não roubar?
- Percepção semelhante a dos humanos?

visionlab
visualization laboratory

Princípios de Design

- NPCs devem gerar uma **experiência divertida para o jogador** e não para o programador.
- No meio acadêmico são criados programas para superar o usuário (derrotar o jogador).
- Meta da inteligência artificial para jogos **não é vencer o jogador**. O objetivo é dar ao jogador desafios e diversão!
- Todo jogador deve ser capaz de superar os desafios do jogo.

visionlab
visualization laboratory

Princípios de Design

- Humanos não gostam de jogar se estão perdendo.
- O jogo deve ser agradável para todos os níveis de habilidade.
- Deve-se **evitar excessos** nos graus de dificuldade (muito fácil ou muito difícil).
- O ideal é **ajustar dinamicamente** a dificuldade dos desafios dependendo do jogador.

visionlab
visualization laboratory

Princípios de Design

- Deve-se usar **métricas para medir o desempenho do jogador** para um ajuste dinâmico de dificuldade.
  - Tempo em cada nível, no. de vidas perdidas, grau de dano…
- Deve-se evitar que o jogador **descubra métrica** e tente engana-la.
- O jogador quer derrotar tudo e todos na sua primeira tentativa dando o melhor de si.

visionlab
visualization laboratory

Princípios de Design

- Todos os NPCs trapaceiam, mas o **jogador não pode perceber**.
- Não existe tecnologia para NPCs serem justos. Os NPCs devem ser simples (mais baratos e realistas).
- Jogador deve entender a o que os NPCs estão fazendo. **O importante é parecer inteligente.**
- NPC só ganha vida quando o Jogador o entende.

visionlab
visualization laboratory

Criando Erros Intencionalmente

- O que torna um **jogo divertido** não corresponde necessariamente à criação de NPCs mais espertos.

- Criar um personagem que possa vencer um humano é **fácil**. O **difícil** é fazer um que perca para um humano em uma batalha desafiadora.

- **Princípios:**

  - Mova antes de atirar
  - Seja visível
  - Tenha uma péssima mira
  - Erre o primeiro tiro
  - Ataques individuais
  - Adição de vulnerabilidades

Criando Erros Intencionalmente

- **Mova-se antes de atirar! Seja visível!**
  - O jogador deve ter a chance de ver os inimigos. O movimento dos inimigos é uma ótima forma de deixar claro a existência deles para o jogador.
- **Tenha uma péssima mira!**
  - Os NPCs poderiam acertar todos os tiros no jogador, mas isso poderia mata-lo em uma fração de segundos. Deve existir uma porcentagem de erro dos disparos. Jogadores gostam de ver balas passando próximas a sua cabeça ou batendo em paredes próximas.
- **Erre o primeiro tiro**
  - Nenhum jogador gosta de morrer sem pelo menos saber quem o acertou. Por isso, é sugerido que o primeiro tiro erre o jogador, ou acerte em algum lugar próximo a ele, de modo a alerta-lo.

Criando Erros Intencionalmente

- **Ataques individuais**
  - Em situações onde existem muitos oponentes simultâneos deve-se fazer com que poucos inimigos ataquem o jogador a cada momento. Ou mais especificamente, que ocorra um revezamento de quem ataca.
- **Adição de vulnerabilidades**
  - Todos os NPCs devem ter algum tipo de vulnerabilidade que possa ser explorada pelo jogador. Mas deve-se tomar cuidado para que vulnerabilidade não comprometa completamente os NPCs quando o jogador a descobrir.

Técnicas Mais Usadas

- **Técnicas mais comuns:**
  - Waypoints e Pathfinding (Busca de Caminho com A*);
  - Máquinas de Estados Finitos (FSM - Finite-State Machine);
  - Aprendizado de Máquina Simplificado;
  - Sistemas de Gatilhos (Trigger Systems);
  - Previsão de Trajetória (jogos de esporte);
  - Incerteza com N-Gram.