Inteligência Artificial

Paulo Vinícius Moreira Dutra[1]

[1]IF Sudeste MG – Campus Muriaé

Sistemas Inteligentes, 2022

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Sumário

- **Parte II - Algoritmos para movimentação**
  - Movimentos randômicos
  - Máquina de estados finitos (Finite State Machine)

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

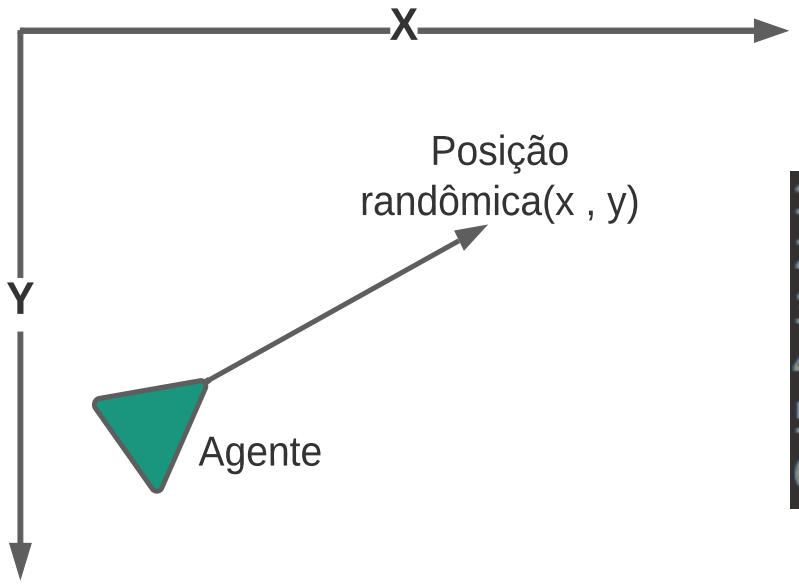
Parte II

- Movimentos randômicos
- Máquina de estados finitos (Finite State Machine)

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Movimentos randômicos

- Definir comportamentos randômicos em jogos é uma das formas de tentar imitar um agente **“inteligente”.** Neste caso, o agente se move em um ambiente até uma determinada posição gerada de forma aleatória.




```
1  alvo.x, alvo.y = gerarProximaPosicao()
2
3  def update():
4      moveagente(alvo.x, alvo.y);
5      if agente chegou em alvo.x, alvo.y:
6          alvo.x, alvo.y = gerarProximaPosicao()
```

Figura: Pseudocódigo

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Movimentos randômicos

- No pseudocódigo temos uma função, **gerarProximaPosicao**, que retorna a posição X, Y gerada de forma aleatória com base na dimensão do cenário.

- A função update é apenas uma representação do looping do jogo. Indicando que o algoritmo é executado enquanto o jogo está em execução.

```
1  alvo.x, alvo.y = gerarProximaPosicao()
2
3  def update():
4      moveagente(alvo.x, alvo.y);
5      if agente chegou em alvo.x, alvo.y:
6          alvo.x, alvo.y = gerarProximaPosicao()
```

Figura: Pseudocódigo

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

```
1  alvo.x, alvo.y = gerarProximaPosicao()
2
3  def update():
4      moveagente(alvo.x, alvo.y);
5      if agente chegou em alvo.x, alvo.y:
6          alvo.x, alvo.y = gerarProximaPosicao()
```

LEMBRE-SE: Pseudocódigo é uma forma genérica de escrever um algoritmo, utilizando uma linguagem simples, de forma a ser entendida por qualquer pessoa sem a necessidade de conhecer a sintaxe de nenhuma linguagem de programação.

Movimentos randômicos – Valor de Tolerância

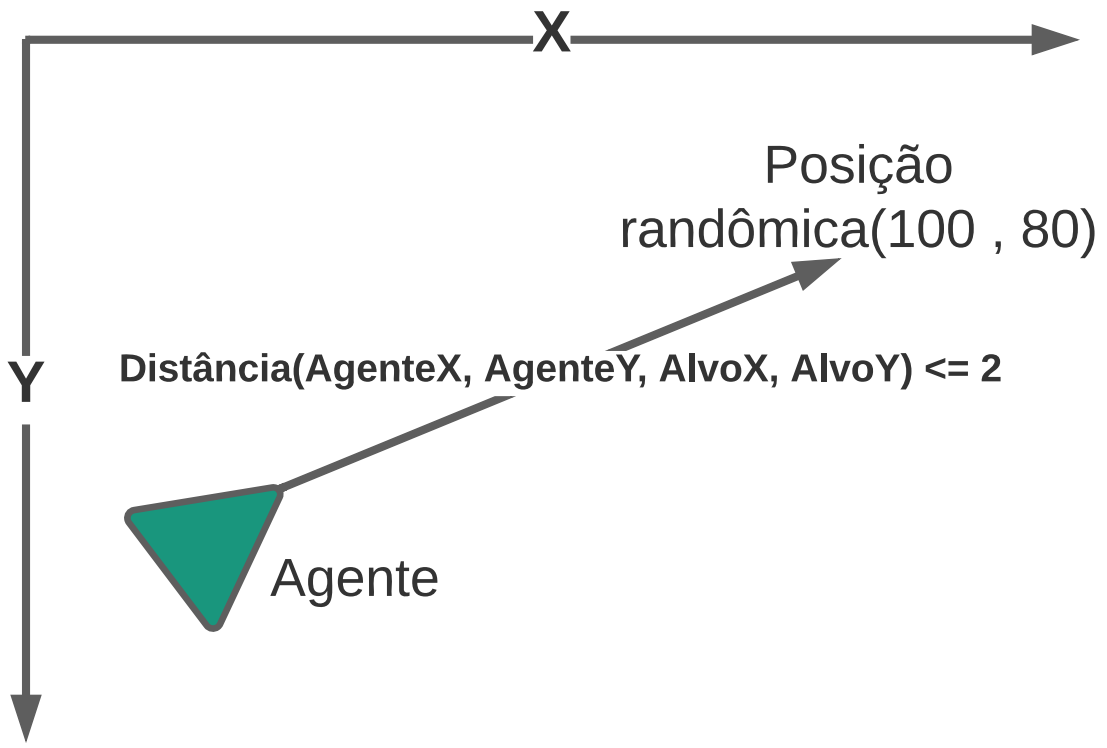
- Em certos algoritmos de movimentação pode ser necessário definir um valor tolerância para chegar na posição correta. Pois certos algoritmos movem os agentes em valores constantes, por exemplo, 55.55, 100.9888 e 245.547

- Se o algoritmo gerou uma posição alvo 100 e 80, respectivamente X, Y, então, você pode utilizar um valor de tolerância entre 0 e 2, verificando se o agente está se aproximando da posição alvo (100 e 80).

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Movimentos randômicos – Valor de Tolerância

- Neste caso, é calculado a distância entre o agente é o alvo enquanto o agente está se movendo.
- O valor de tolerância é necessário em certos casos para evitar que o agente ultrapasse o ponto de destino.



INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Movimentos randômicos – Valor de Tolerância

- Nosso pseudocódigo utilizando o valor de tolerância:

```
1  alvo.x, alvo.y = gerarProximaPosicao()
2
3  def update():
4      moveagente(alvo.x, alvo.y);
5      if calcularDistancia(agente.x, agente.y, alvo.x, alvo.y) <= 2:
6          alvo.x, alvo.y = gerarProximaPosicao()
```

Figura: Pseudocódigo

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Movimentos randômicos – Temporizador

- Uma terceira alternativa é utilizar um temporizador. Neste caso, mudamos levemente o algoritmo. No pseudocódigo abaixo, o agente muda de posição a cada 2 segundos.

```
1  alvo.x, alvo.y = gerarProximaPosicao()
2  inicia um temporizador como 0
3
4  def update():
5      moveagente(alvo.x, alvo.y)
6      if temporizador é >= 2 segundos:
7          alvo.x, alvo.y = gerarProximaPosicao()
8          reinicia o temporizador para 0
```

Figura: Pseudocódigo

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Movimentos randômicos

- Com movimentos randômicos podemos simular diversos comportamentos, e há diferentes formas de se implementar.

- Até esse ponto, implementamos uma forma simples de um agente. Podemos adicionar um pouco mais de **“inteligência”** ao nosso agente, por exemplo, permitindo que ele desvie de obstáculos caso encontre algum ou gerar uma nova posição alvo.

Exercício - Movimentos randômicos

Implemente um agente para realizar o comportamento do movimento randômico visto nos slides anteriormente.

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Finite State Machine - FSM

- A máquina de estado finito é uma das formas mais básicas de implementação de inteligência artificial em jogos.

- Os personagens do jogo basicamente assumem um estado. Normalmente, as ações e comportamento estão associadas a cada estado do personagem. Por exemplo, atirar, andar, perseguir e entre outros.

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Finite State Machine - FSM

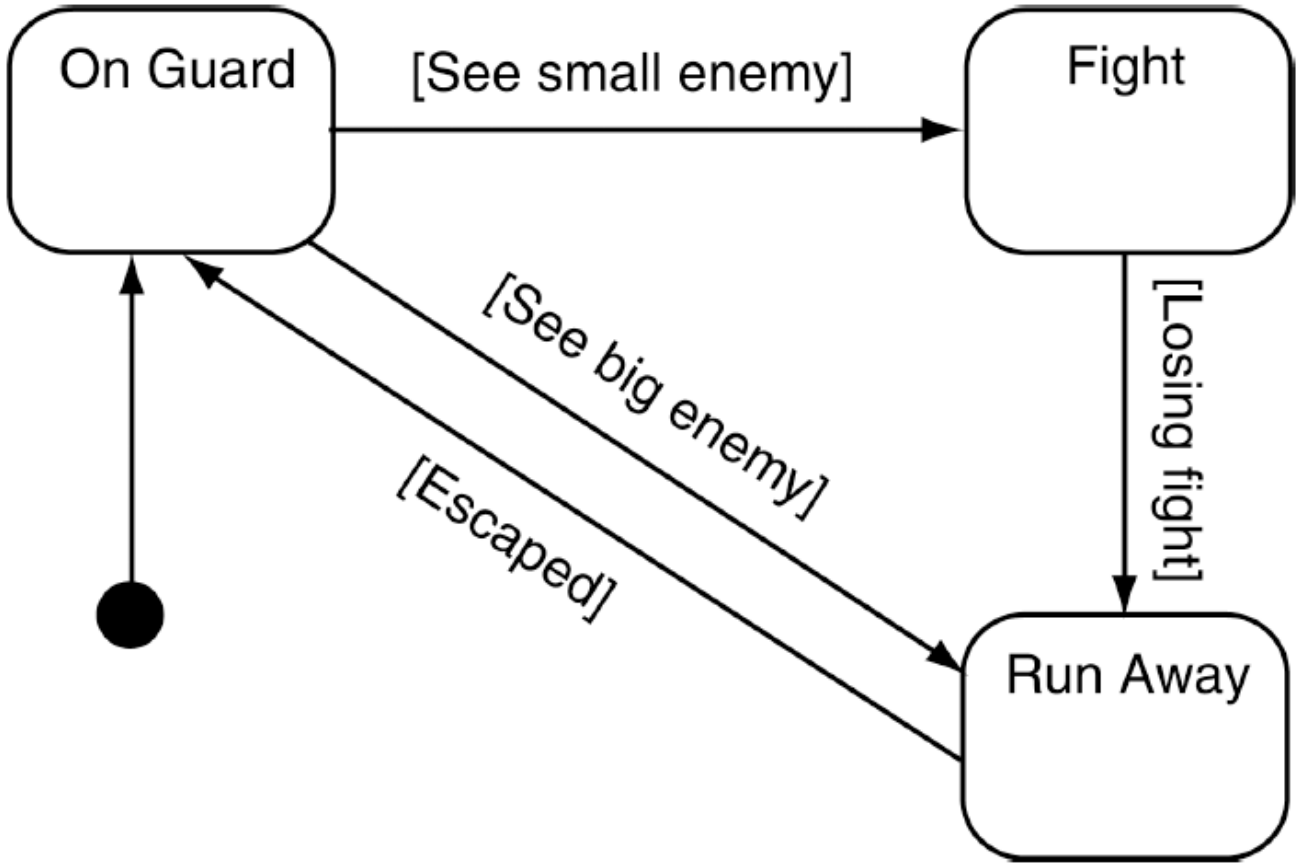


Figura: Uma simples máquina de estados (MILLINGTON. Ian, 2019)

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Finite State Machine - FSM

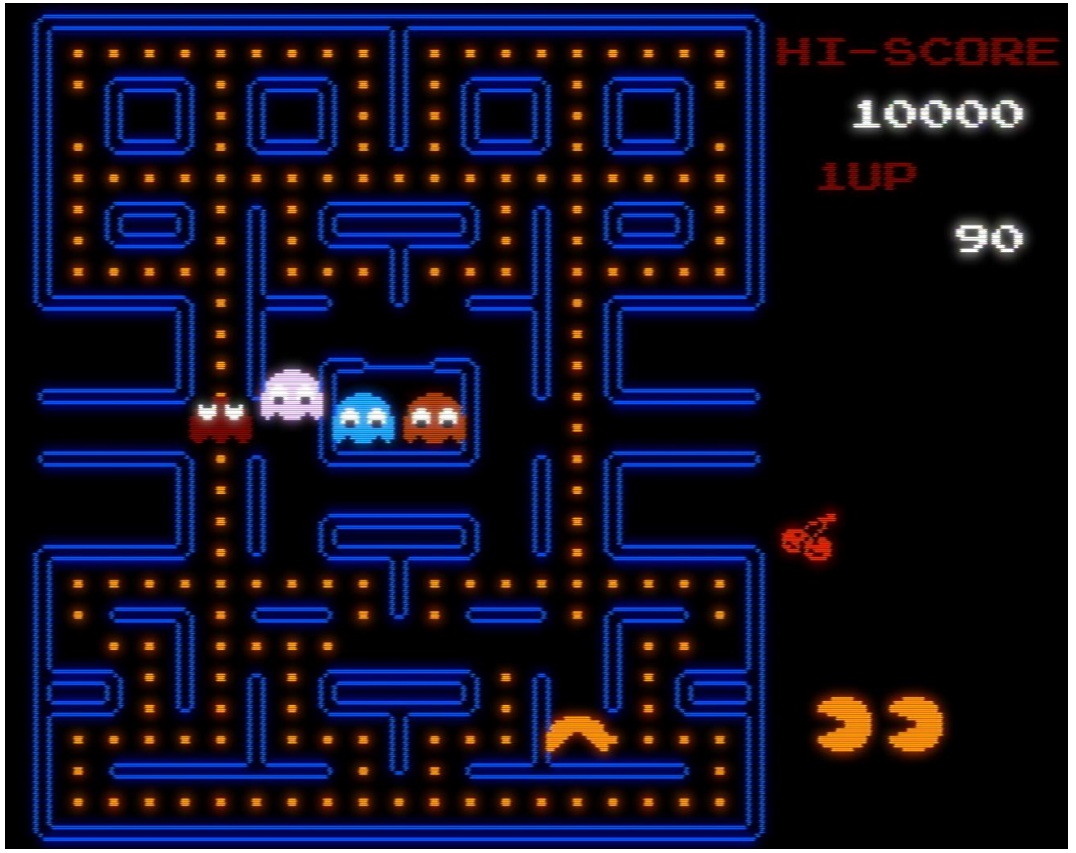
- **Estado:** Define o conjunto de estados destintos que cada personagem do jogo pode escolher (Patrulhar, perseguir ou atirar).
- **Transições:** Define a relação entre os estados.
- **Regras:** Utilizado para ativar a transição entre os estados (Player está visível, perto para atacar ou player morreu)
- **Eventos:** Define o que será disparado após a verificação das regras.

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Exemplo de FSM – PAC-Man

- Os fantasmas Inky, Pinky, Blinky e Clyde do jogo Pac-Man são implementados via FSM utilizando três estados distintos:

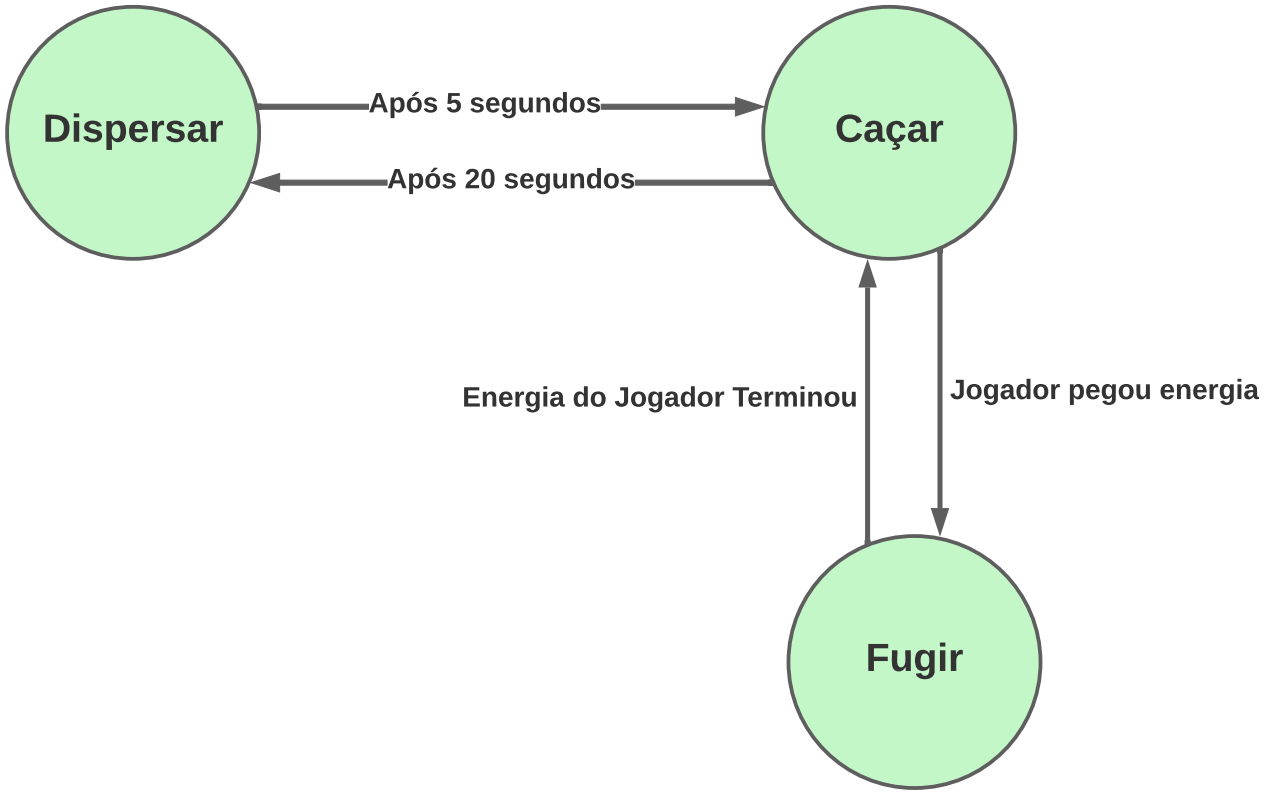
- Comportamentos:
  - Caçar (Chase)
  - Fugir (Evade)
  - Dispersar (Scatter)



INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Exemplo de FSM – PAC-Man

- Máquina de Estados para os fantasma:

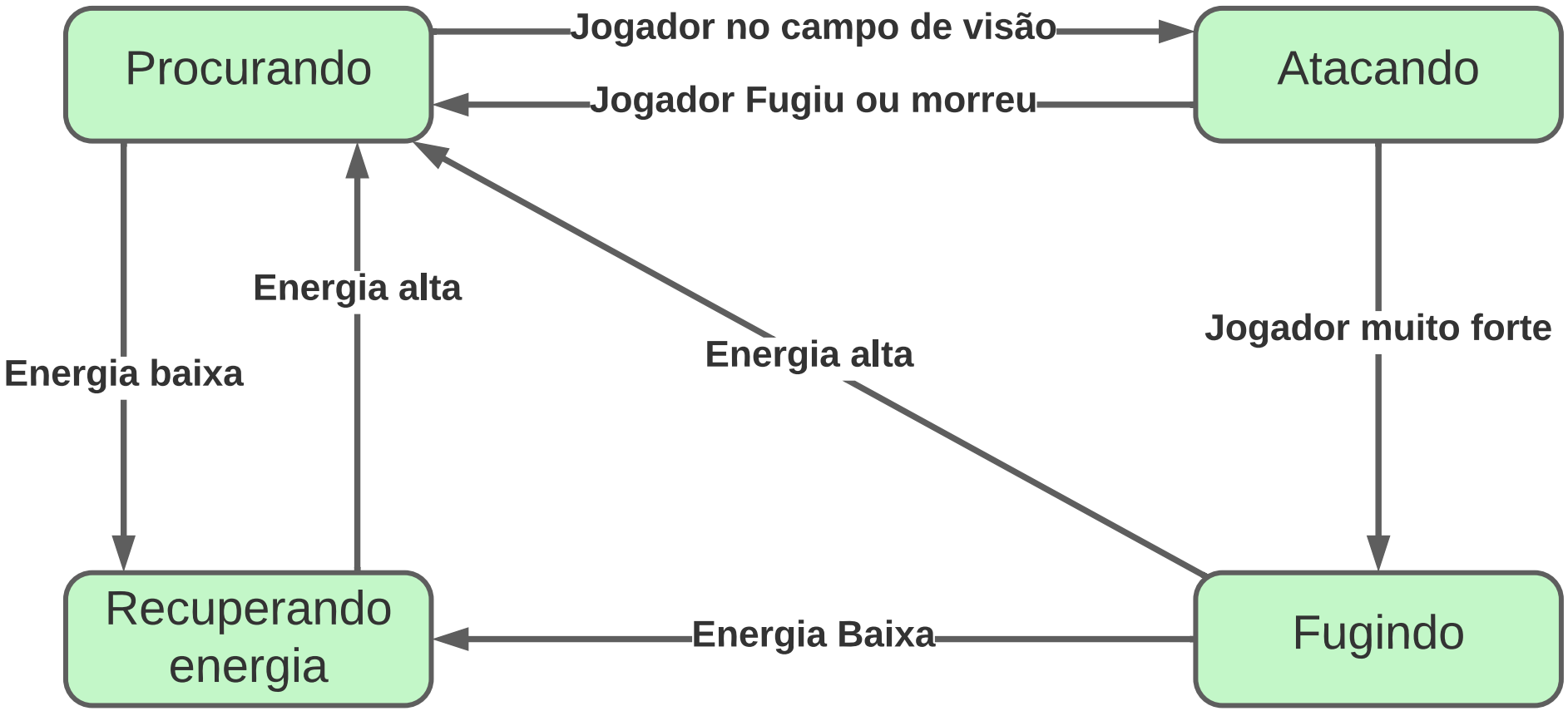



```
1  # Nome de cada estado.
2  CACAR      = 1
3  FUGIR      = 2
4  DISPERSAR = 3
5
6  # Estado Atual
7  estadoAtual = DISPERSAR
8
9  def update():
10     if estadoAtual == DISPERSAR:
11         if aguardar_tempo() >= 5:
12             estadoAtual == CACAR
13     else if estadoAtual == CACAR:
14         if playerPegouEnergia():
15             estadoAtual == FUGIR:
16         if aguardar_tempo() >= 20:
17             estadoAtual == DISPERSAR:
18     else if estadoAtual == FUGIR:
19         if terminouEnergia():
20             estadoAtual == CACAR:
```

Figura: Pseudocódigo

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Máquina de estados finitos



INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Vantagens Máquina de estados finitos

- São fácies de implementar e podemos implementar de várias formas.
- Não requerem muito processamento devido a sua simplicidade.
- São extremamente intuitivas e com isso são fáceis de depurar.
- Podemos facilmente adicionar novos comportamentos.

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Desvantagens Máquina de estados finitos

- A complexidade das FSM tendem aumentar quando o número de comportamentos do agente aumenta.
- Caso precisamos realizar uma representação gráfica de uma FSM com muitos estados, a sua visualização se torna inviável.
- Muitas das vezes, movimentos muitos complexos não são interessantes de implementar utilizando somente FSM.

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Referências

RUSSEL et al. **Inteligência Artificial – Tradução da Terceira Edição**. Elsevier, 2013. ISBN 788535237016

SCHWAB, B. **AI Game Engine Programming**. Course Technology/Cengage Learning, 2009. (Game Programming). ISBN 9781584505723.

MILLINGTON, Ian. **AI for Games, Third Edition**. CRC Press, 2019. ISBN 9781351053297

SPIEGELEIRE, S. de; et al. **The role of Artificial Intelligence in future technology.** The Hague Centre for Strategic Studies. 2017

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé

Referências

BOURG M. David. **AI For Games Developers**. O'REILLY, 2004. ISBN 0596005555

BARRERA, R. **Unity 2017 Game AI Programming - Third Edition: Leverage the power of Artificial Intelligence to program smart entities for your games, 3rd Edition**. Packt Publishing, 2018. ISBN 9781788393294

INSTITUTO FEDERAL
Sudeste de Minas Gerais
Campus Muriaé