

Máquina de Estados e Árvore de Comportamentos

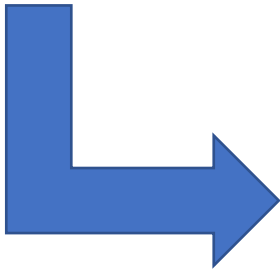
Daniel Nogueira

dnogueira@ipca.pt

Introdução

Máquinas de Estado Finitos

Finite State Machine (FSM)



- É um modelo matemático utilizado para representar os diversos comportamentos e as respetivas transições entre estes em um programa. Ou seja, é **composta por estados e transições**.
- Cada estado representa o sistema em um determinado momento no tempo e não é possível uma máquina de estados estar em dois estados ao mesmo tempo.
- É um conjunto de estados finitos que funcionam como intermediários entre uma relação de entrada e saídas. Desta forma, a saída dependerá do estado das entradas naquele momento
- Quando uma máquina está em um estado, ela aguarda que as condições para uma transição sejam atingidas
- Cada máquina possui um estado inicial, onde a máquina começa, e pode ter um ou mais estados finais, indicando que a máquina terminou a tarefa computacional.

Introdução

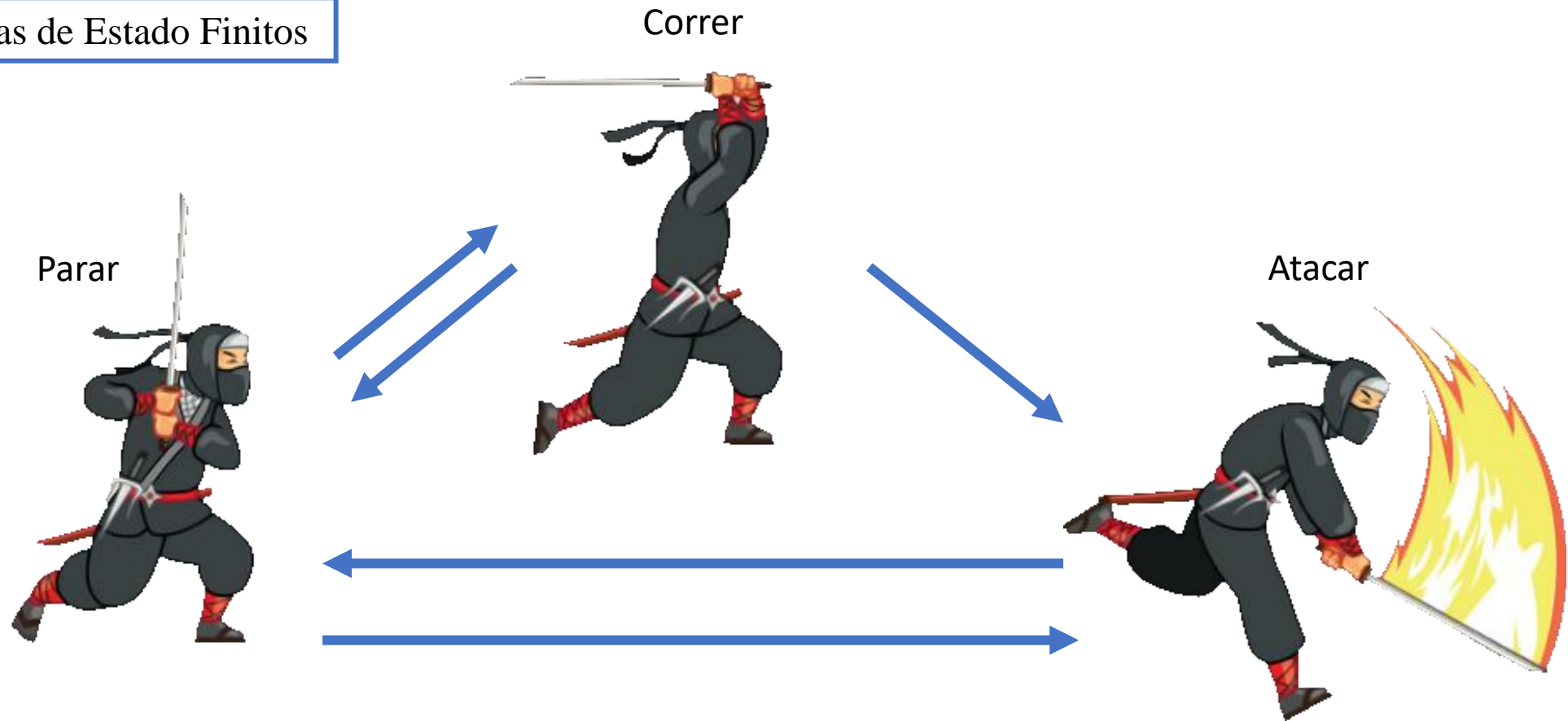
Máquinas de Estado Finitos

Finite State Machine (FSM)



Introdução

Máquinas de Estado Finitos

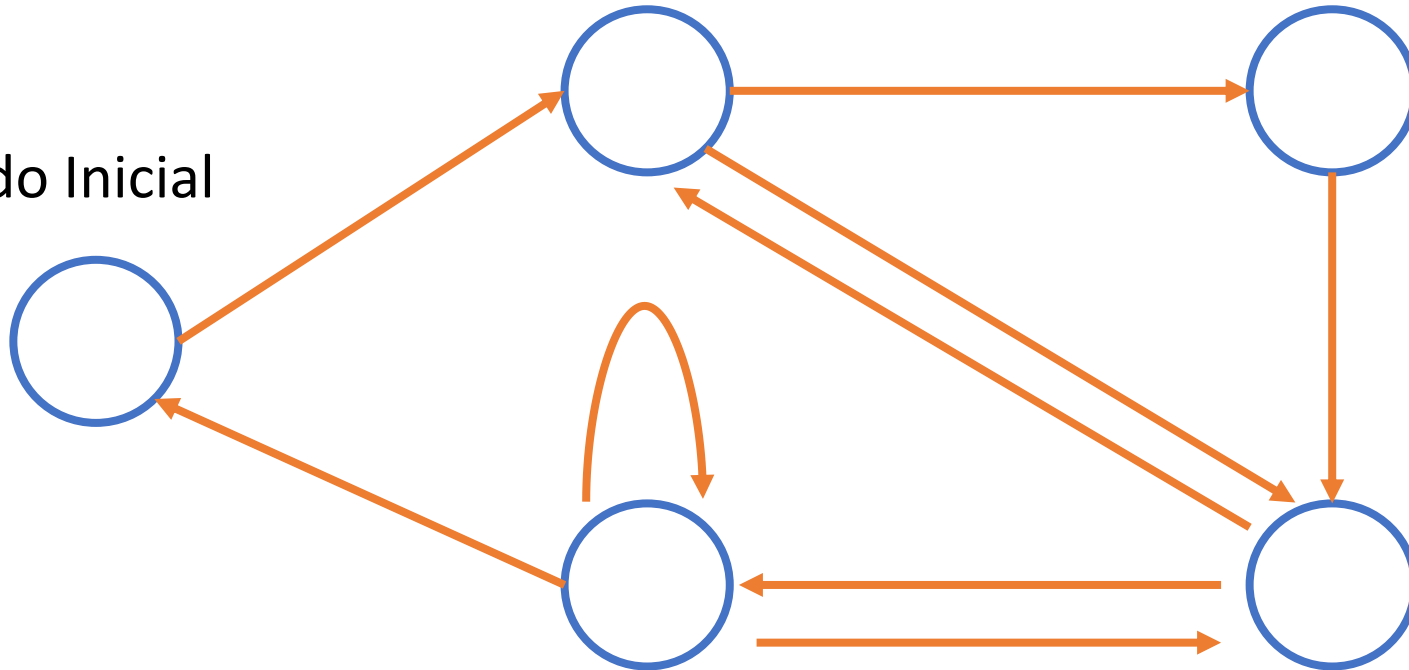


Representação

Máquinas de Estado Finitos

Regras de Transição

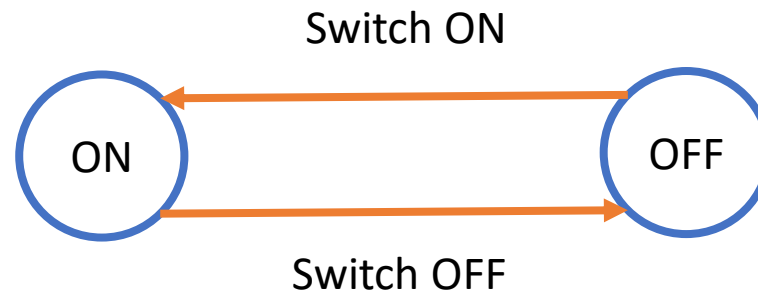
Estado Inicial



Exemplo

Máquinas de Estado Finitos

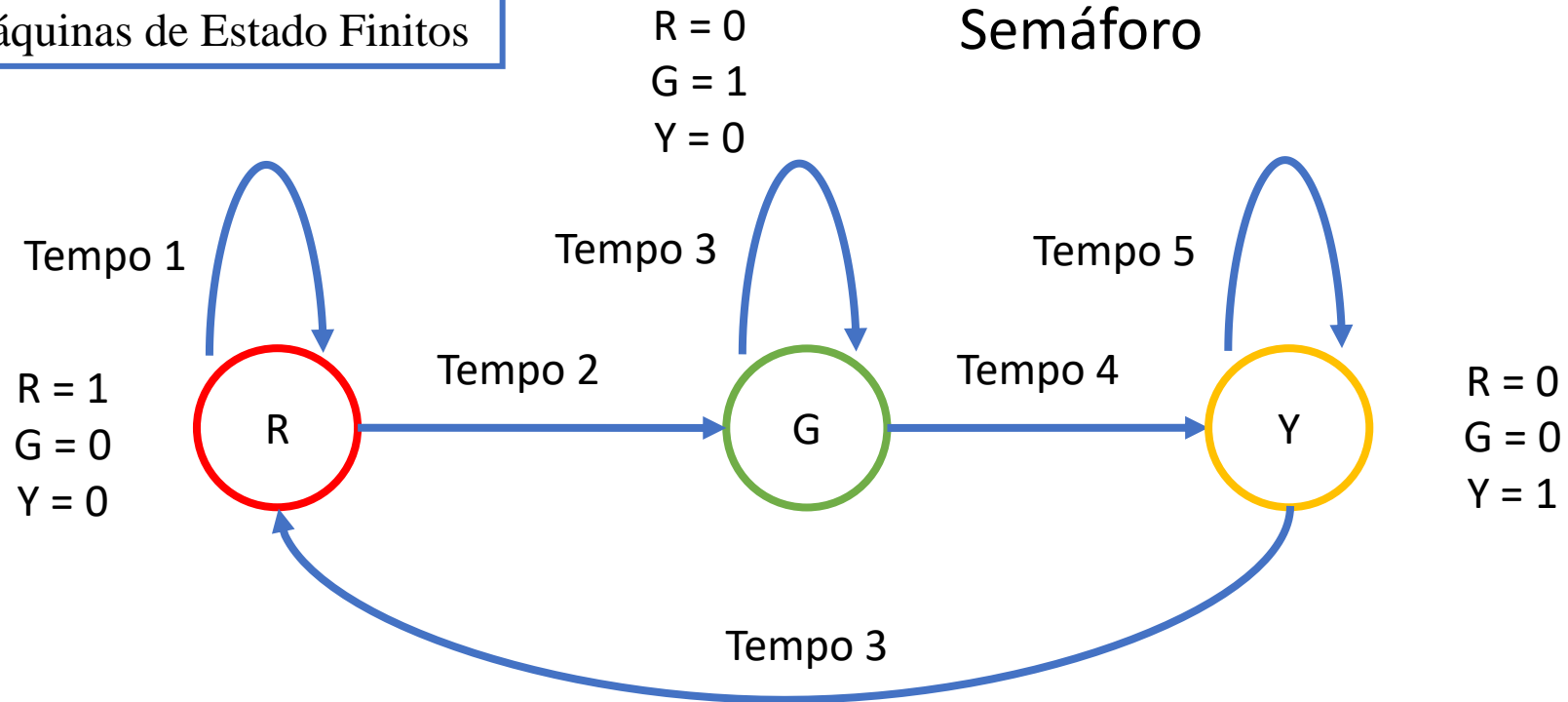
Interruptor de Lâmpada



Exemplo

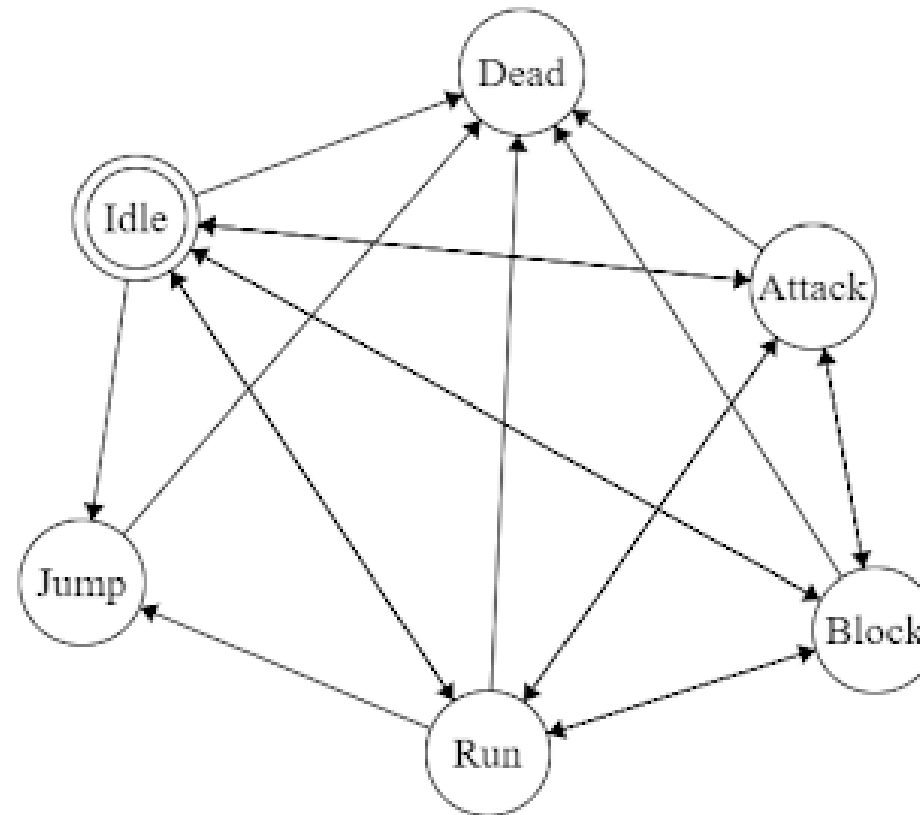
Máquinas de Estado Finitos

Semáforo



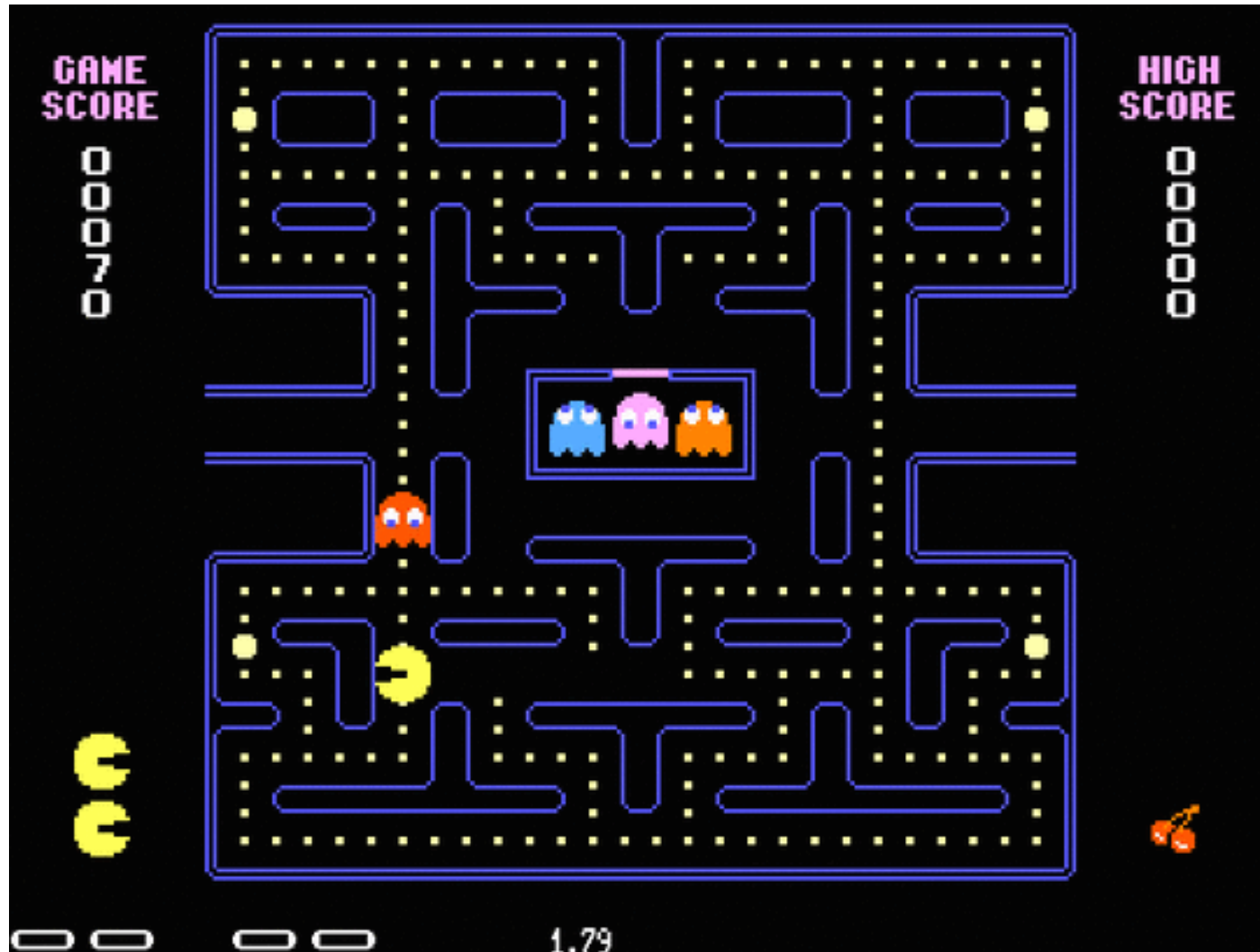
Exemplo

Máquinas de Estado Finitos



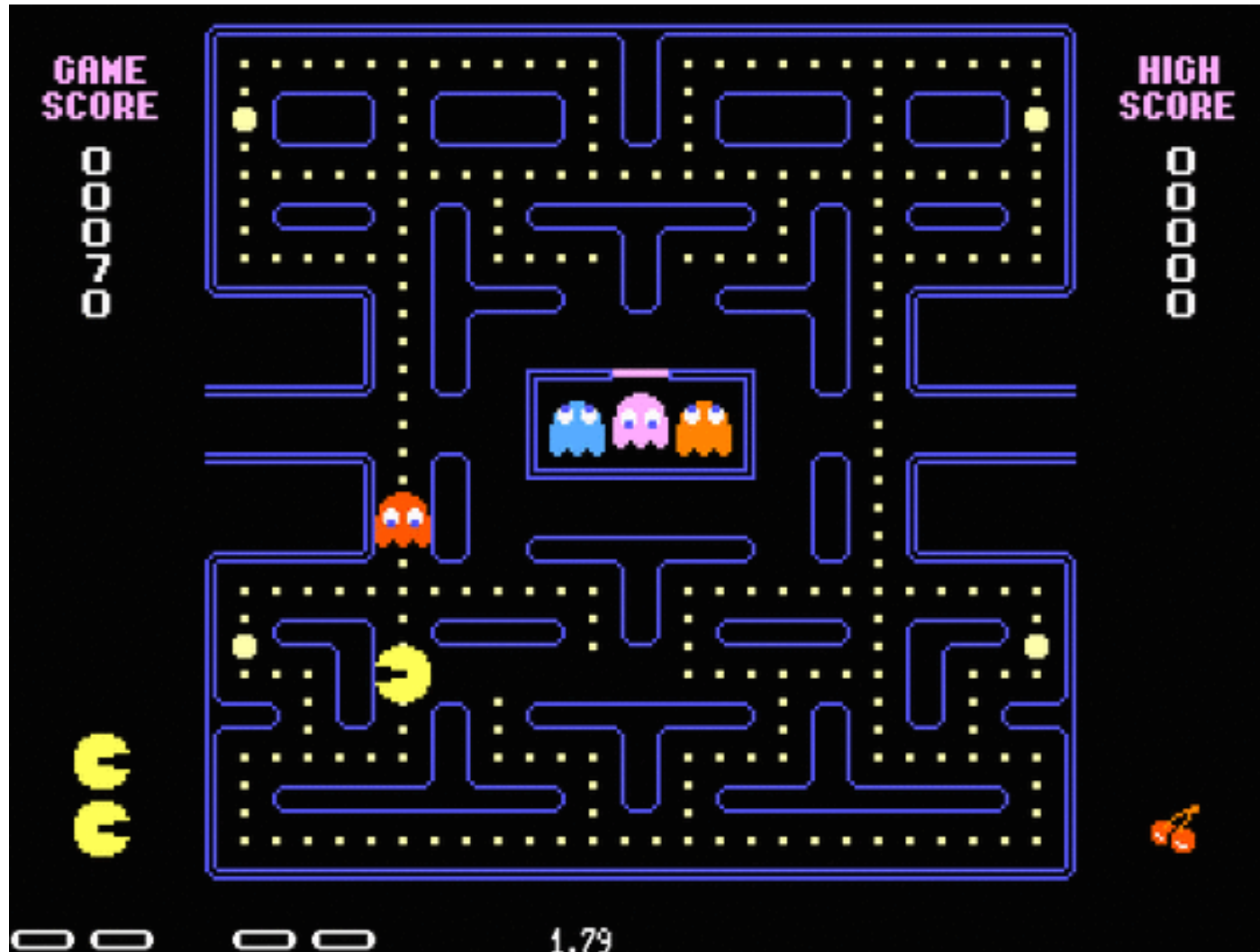
Exemplo

NAMCO™ Pac-Man - 1979



Exemplo

NAMCO™ Pac-Man - 1979

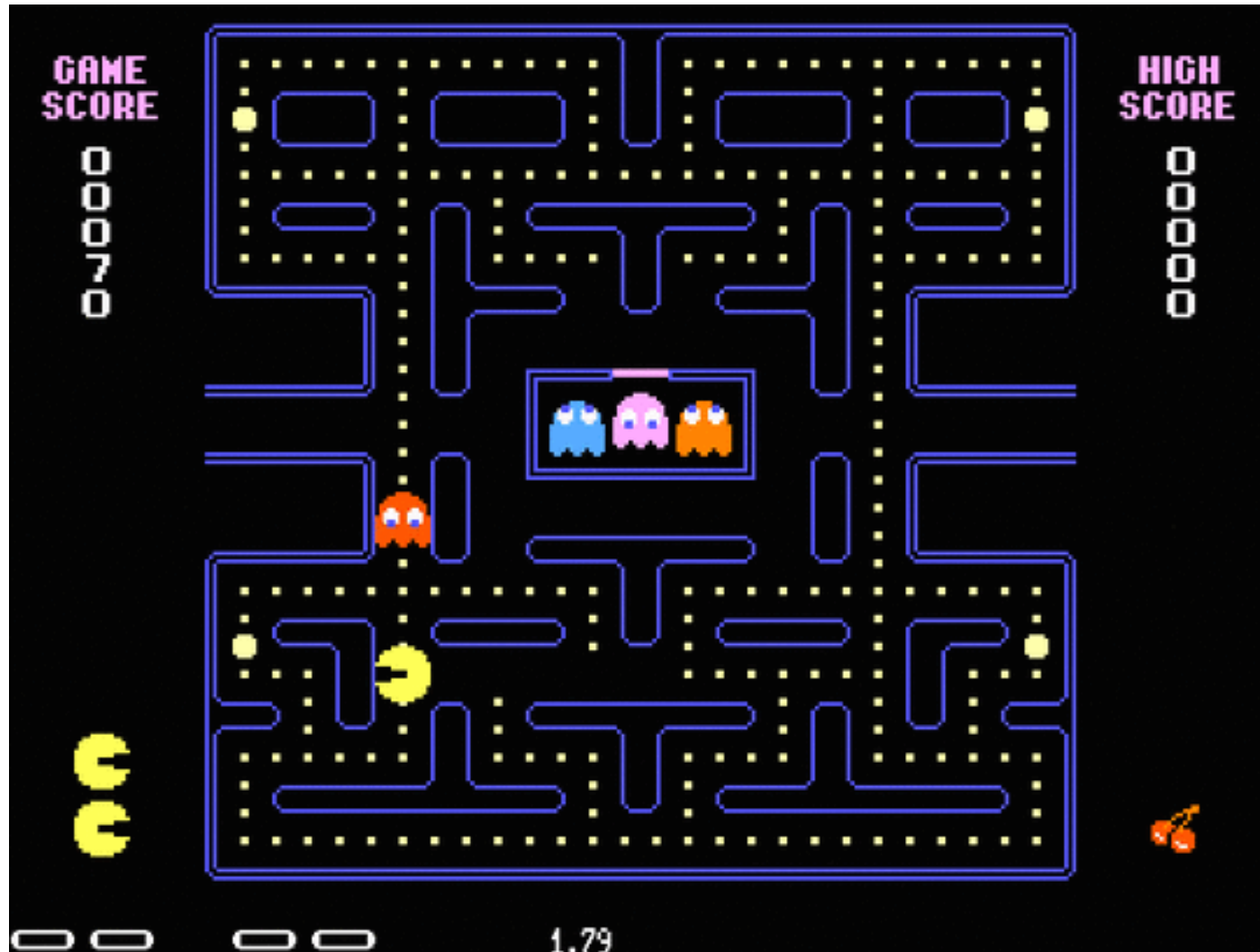


Comportamentos:

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

Exemplo

NAMCO™ Pac-Man - 1979



- A transição de estados ocorre em várias situações do jogo
- A ação de caçar é diferente para cada fantasma

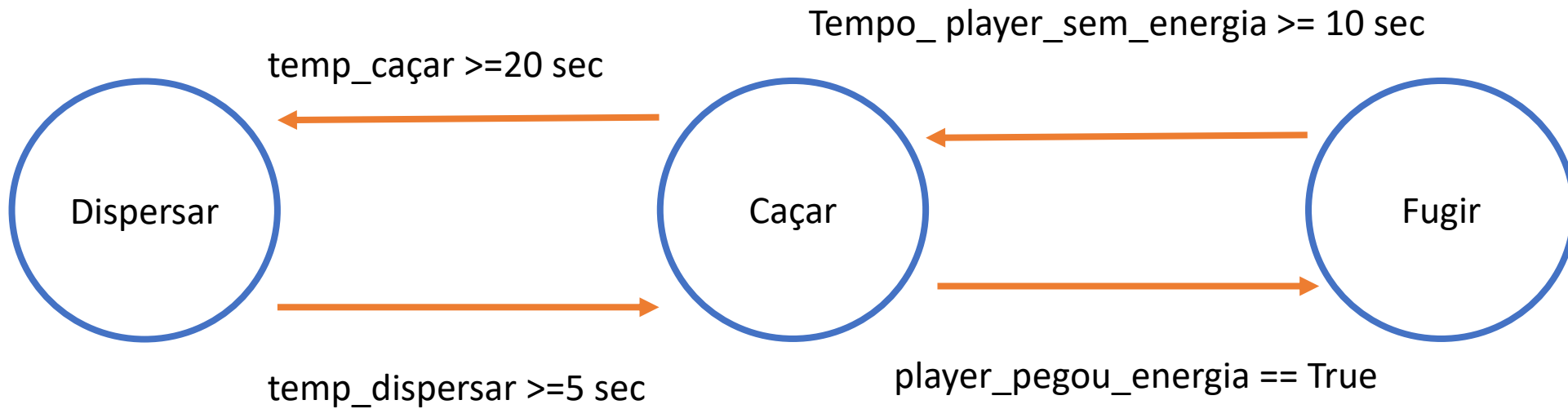
Exemplo



Exemplo

Máquinas de Estado Finitos

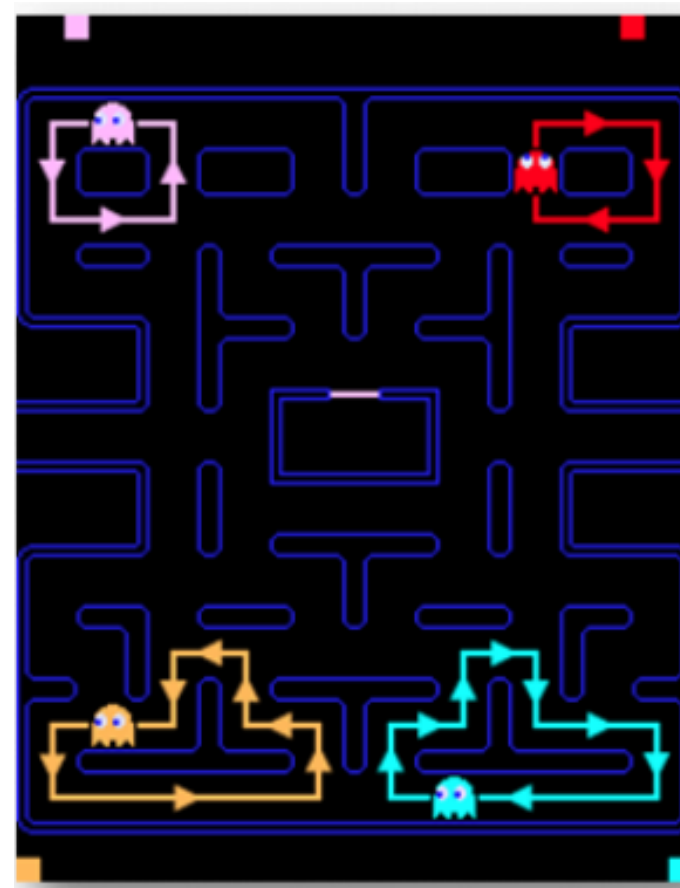
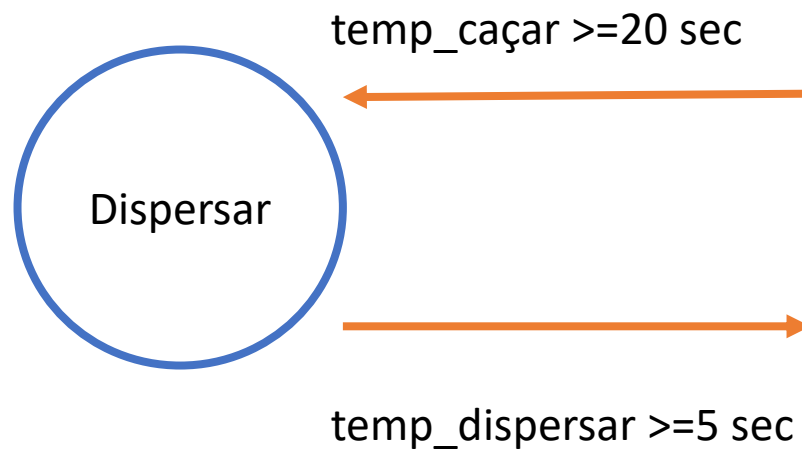
NAMCO™ Pac-Man - 1979



Exemplo

Máquinas de Estado Finitos

NAMCO™ Pac-Man - 1979

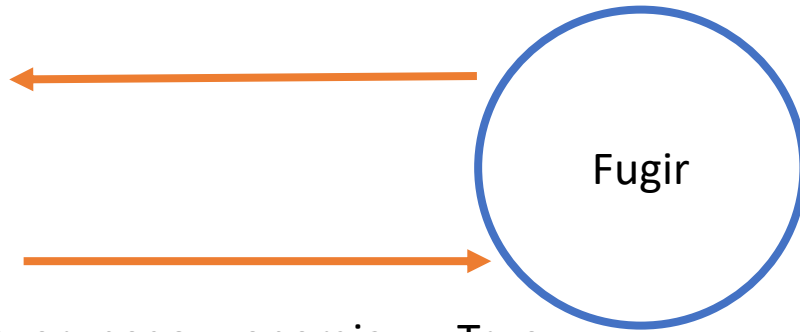


Exemplo

Máquinas de Estado Finitos

NAMCO™ Pac-Man - 1979

$\text{tempo_player_sem_energia} \geq 10 \text{ sec}$



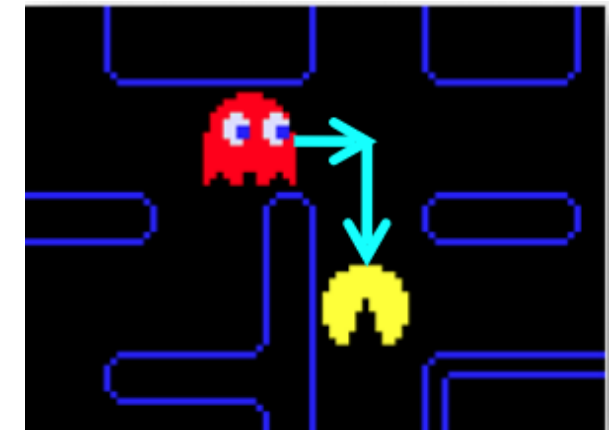
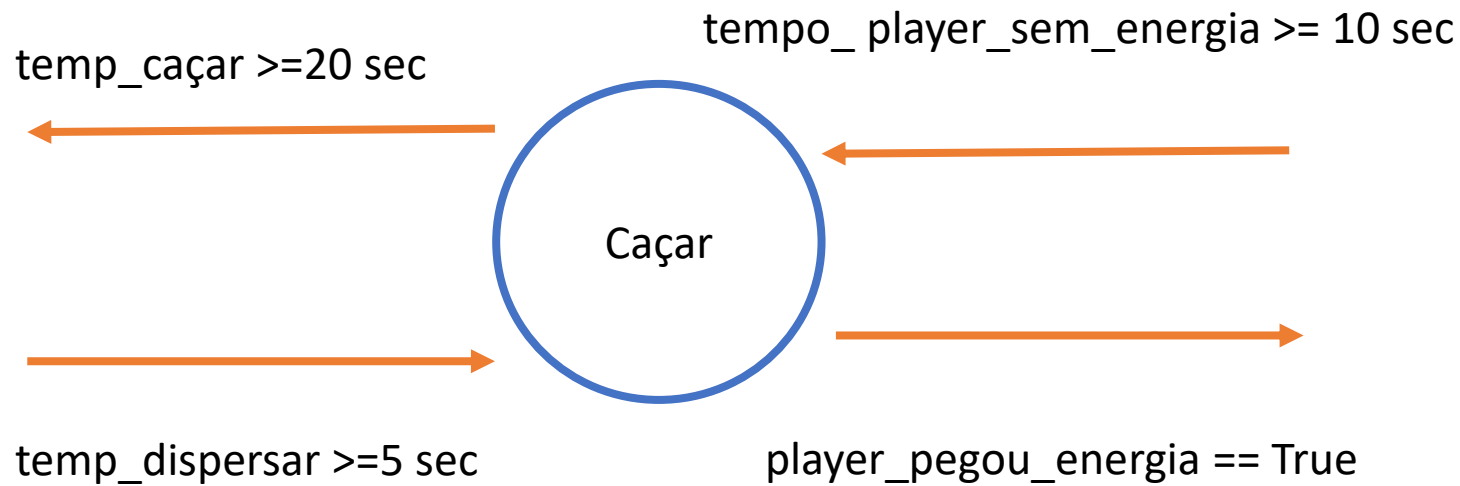
$\text{player_pegou_energia} == \text{True}$



Exemplo

Máquinas de Estado Finitos

NAMCO™ Pac-Man - 1979

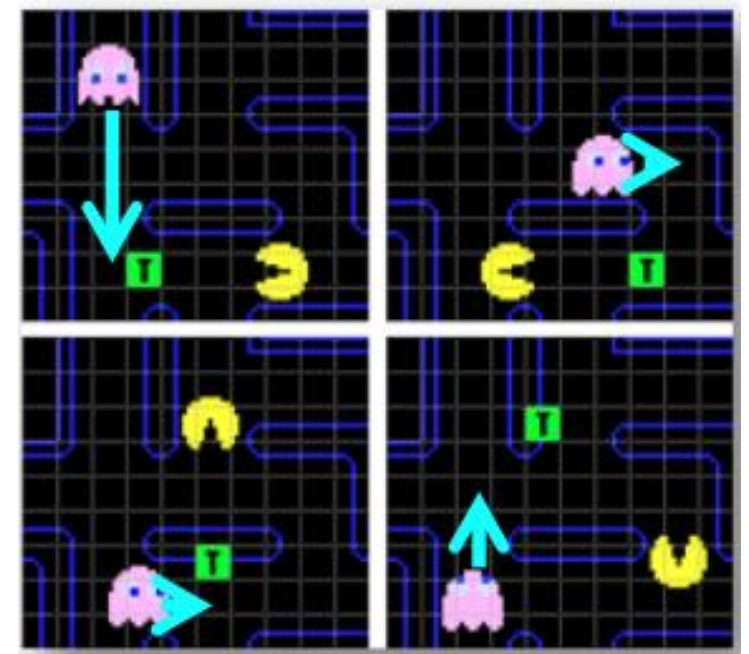
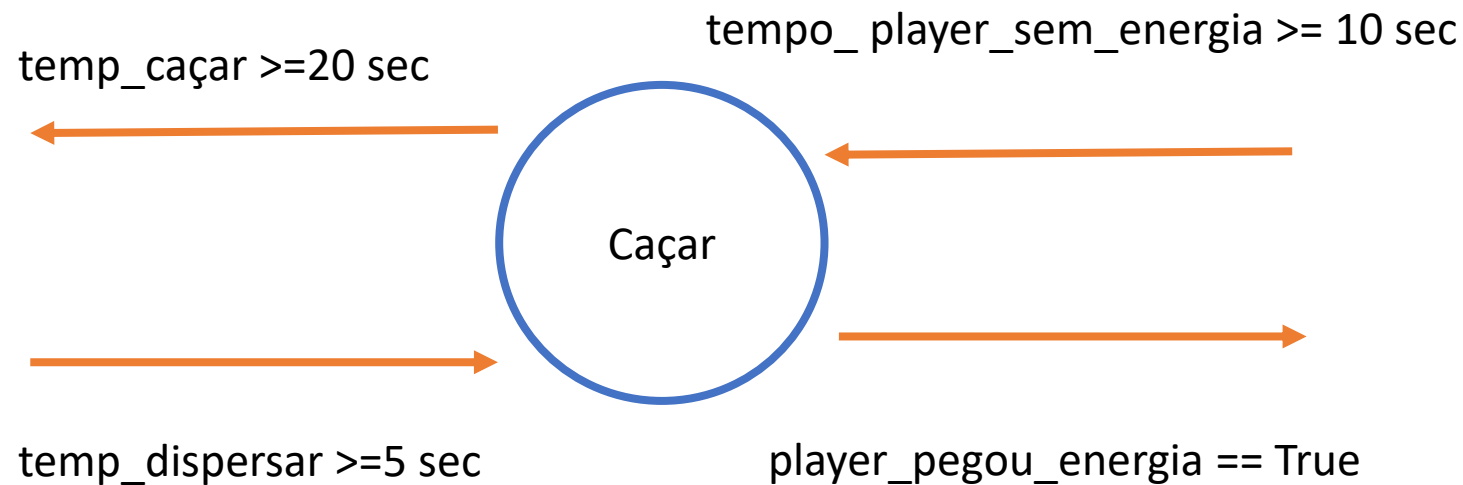


Movimenta em busca da posição do player

Exemplo

Máquinas de Estado Finitos

NAMCO™ Pac-Man - 1979

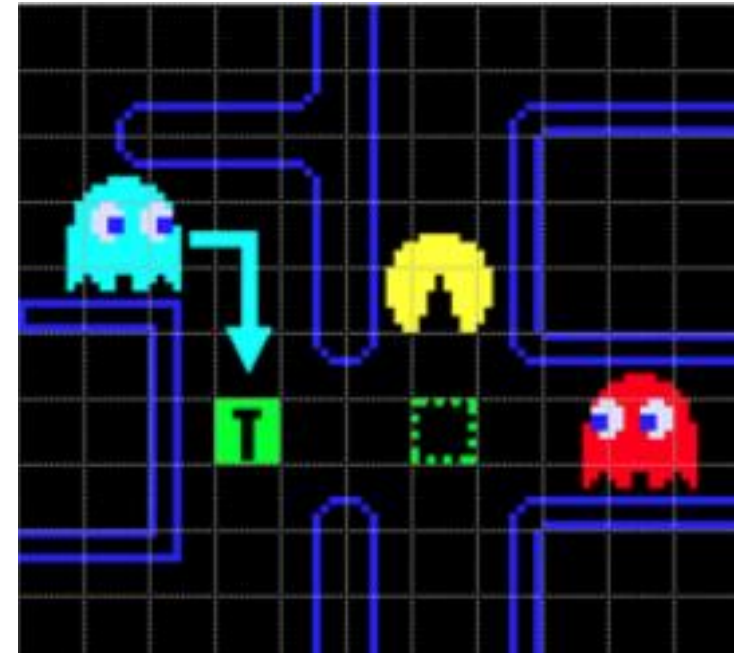
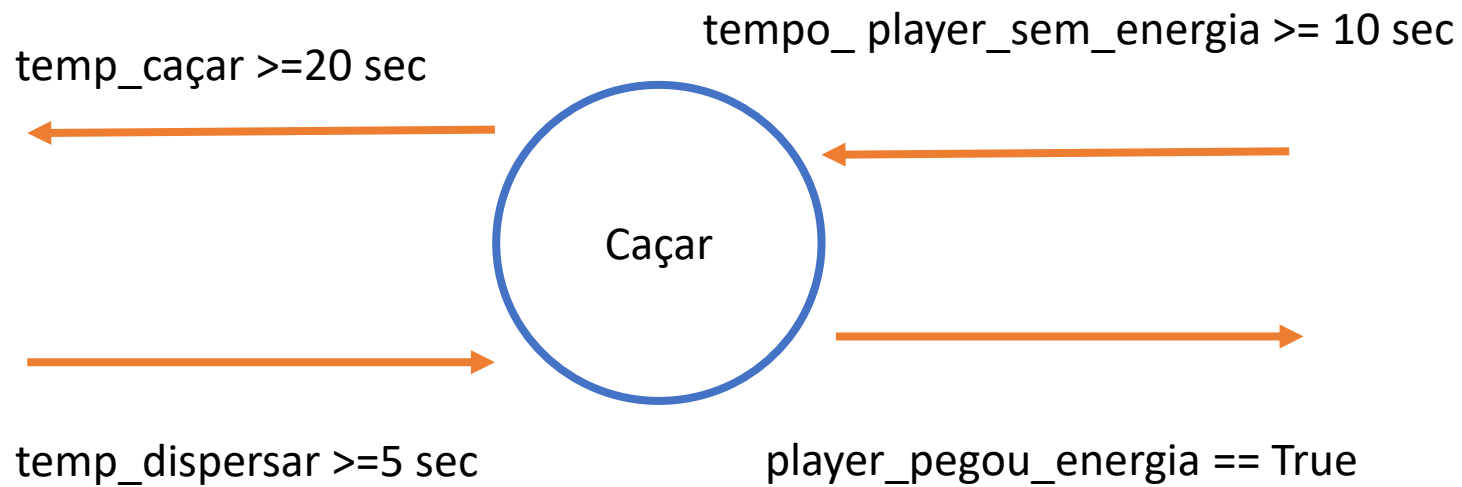


Movimenta em busca da posição 4 tiles a frente do player

Exemplo

Máquinas de Estado Finitos

NAMCO™ Pac-Man - 1979

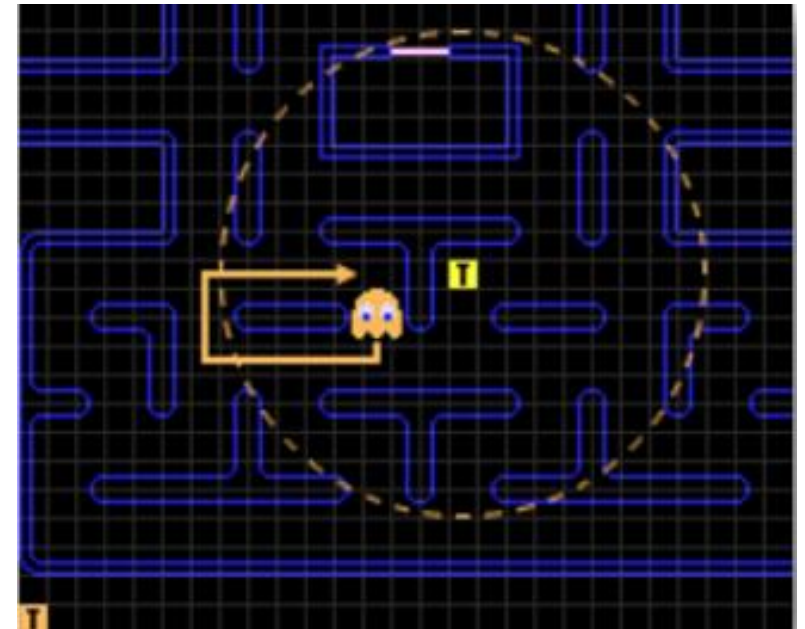
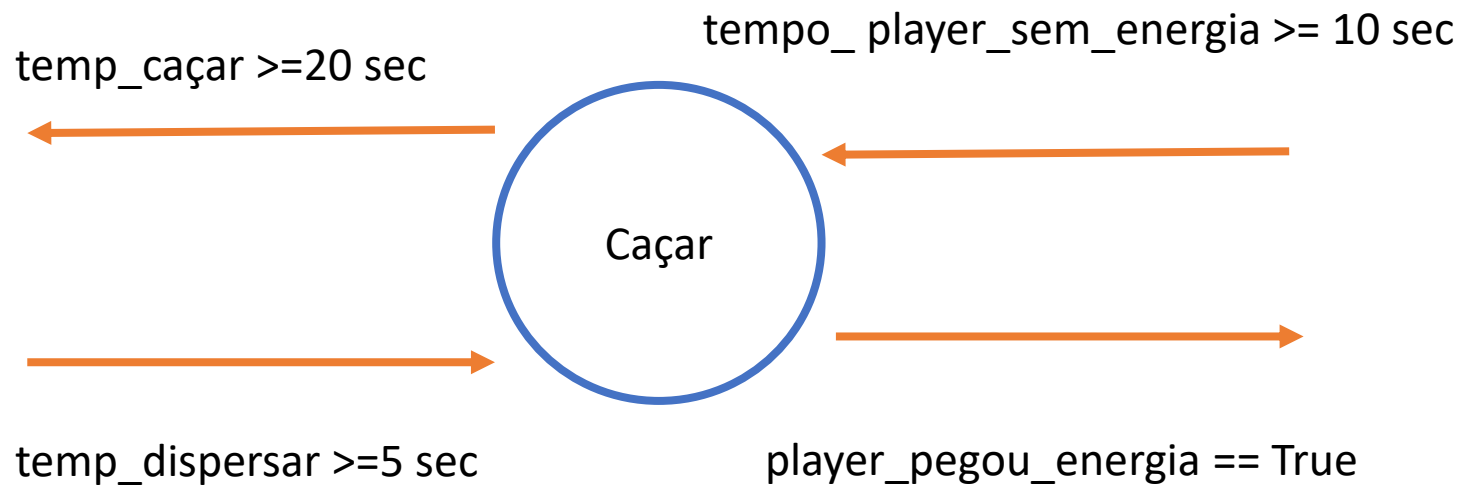


Movimenta combinando a posição e/ou direção do player e do Blink

Exemplo

Máquinas de Estado Finitos

NAMCO™ Pac-Man - 1979



Movimenta em direção do player (se estiver longe). Movimenta em direção ao canto (se estiver perto)

Exemplo

Id SOFTWARE™ Quake - 1996



Comportamentos NPCs:

- Procurar armadura (Find Armor)
- Procurar cura (Find Health)
- Perseguir/Caçar (Chase)
- Atacar (Attack)
- Correr (Run Away)
-

Exemplo

Id SOFTWARE™ Quake - 1996



Comportamentos Player:

- Mover (Move)
- Tocar Objeto (Touch Object)
- Morrer/Explodir (Die)

Exemplo

EA SPORTS™ FIFA 23



Comportamentos Jogadores:

- Driblar (Dribble)
- Correr atrás da bola (Chase Ball)
- Marcar adversário (Hold Player)
-

Exemplo

EA SPORTS™ FIFA 23



Comportamentos Grupo:

- Defender (Defend)
- Atacar (Attack)
-

Características

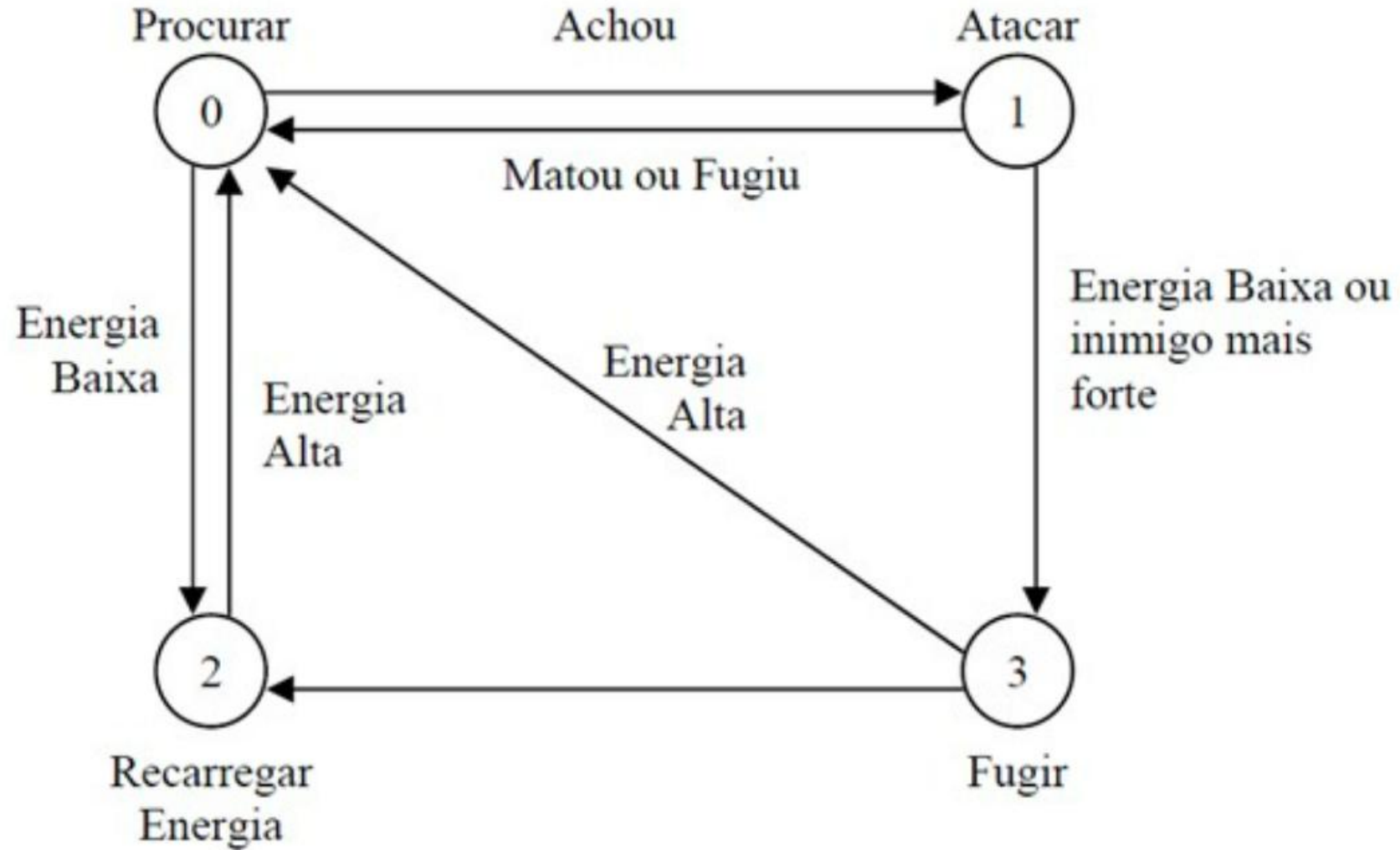
Vantagem

- Baixa necessidade de processamento
- São intuitivas: facilmente identificável ao observar os diagramas (facilita o trabalho dos Gamers Designers)
- São flexíveis: são facilmente ajustáveis pelo programador para prover comportamentos
- São simples de implementar

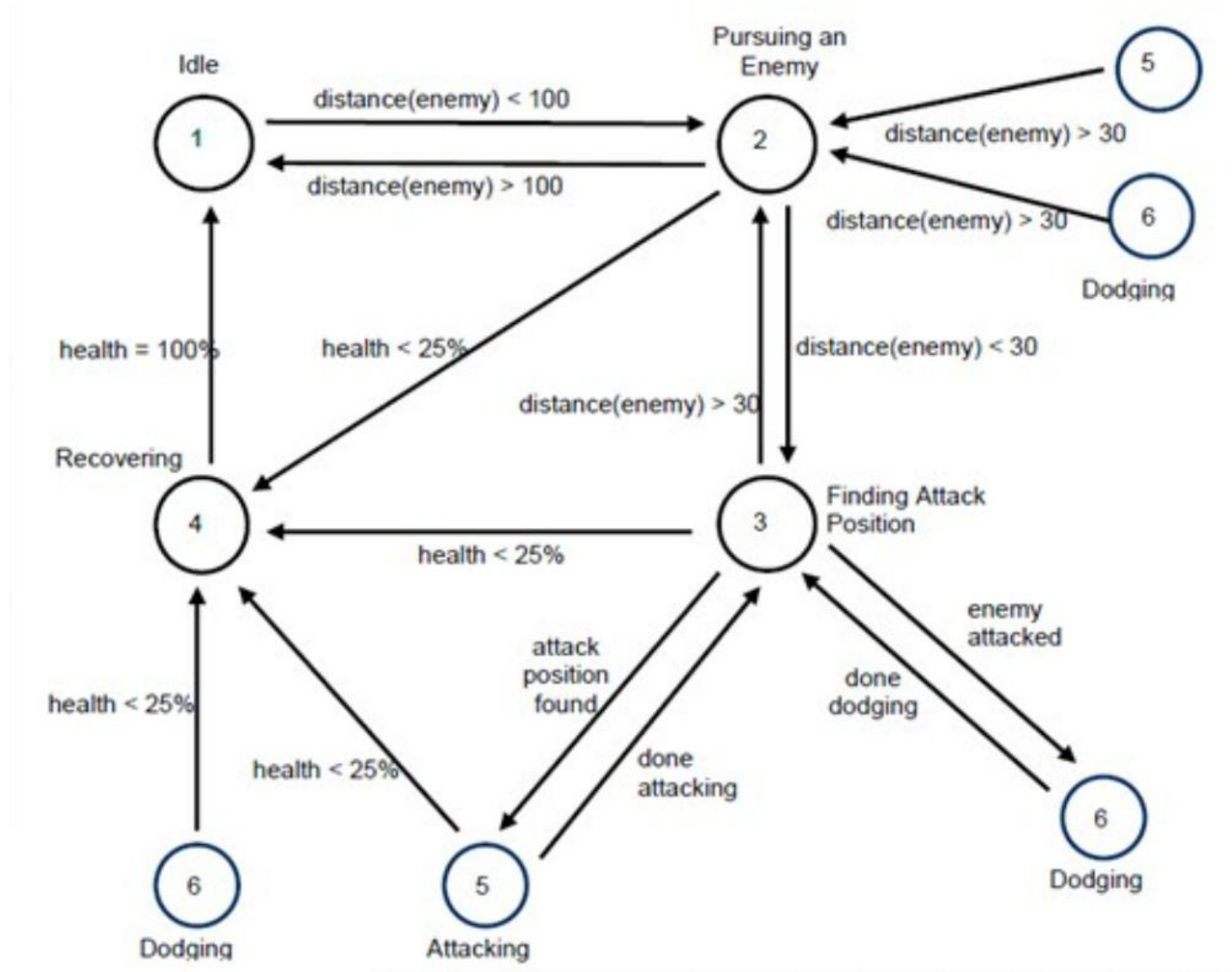
Desvantagem

- Dificuldade de representar esquemas muito complexos
- Possibilidade de falhas em esquemas mais complexos

Exemplos

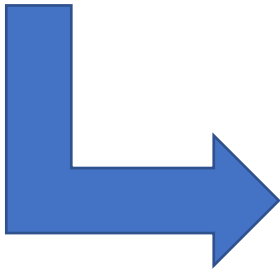


Exemplos



Introdução

Árvore de Comportamentos



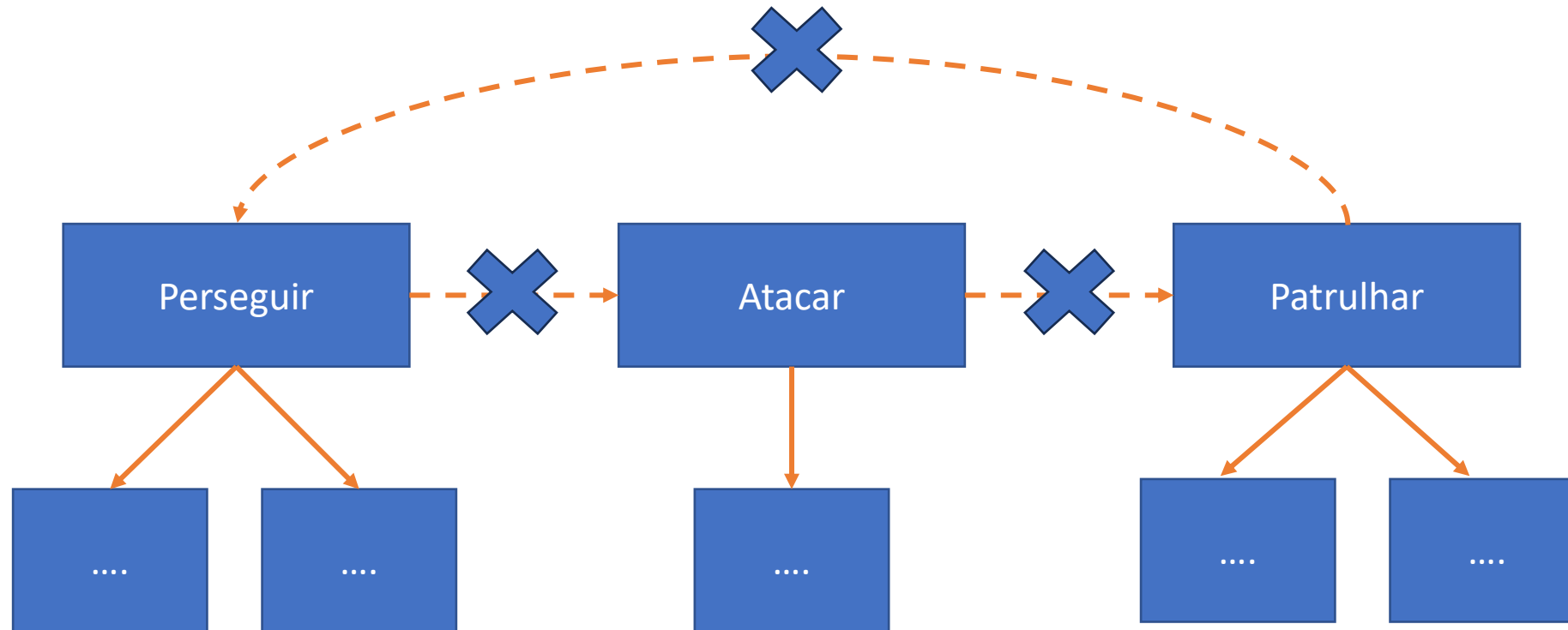
Behaviour Tree (BTs)

- A árvore de comportamento é uma árvore de nós hierárquicos que controlam o fluxo de tomada de decisão de uma entidade de Inteligência Artificial (IA)
- É uma arquitetura de IA que fornece aos **Non Player Characters (NPC)** do jogo a capacidade de selecionar comportamentos e executá-los, por meio de uma arquitetura semelhante a uma árvore que define operações lógicas simples.
- Tem sistemas semelhantes as de uma Máquina de Estado Finita
- Os estados das BTs são chamadas de **Tarefas**.
- Nas folhas estão os comandos reais que controlam a entidade da IA, e formando os ramos estão vários tipos de nós utilitários que controlam a caminhada da IA pelas árvores para alcançar as sequências de comandos mais adequadas à situação.

Definições

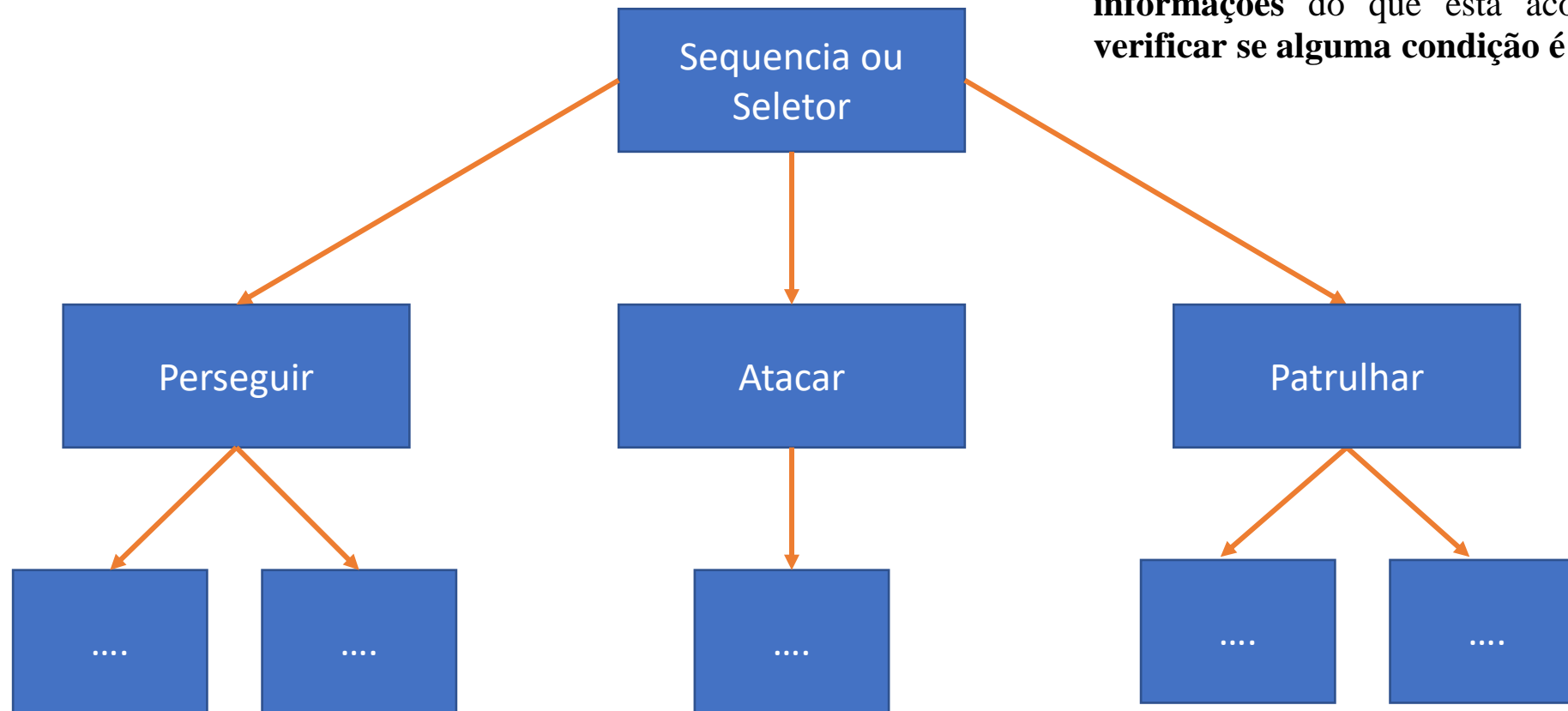
Árvore de Comportamentos

As folhas da árvore proporcionam a comunicação entre a **lógica da inteligência artificial** e a **game-engine** do jogo, sendo utilizadas para **buscar informações** do que está acontecendo no jogo e **verificar se alguma condição é verdadeira ou falsa**.



Definições

Árvore de Comportamentos



As folhas da árvore proporcionam a comunicação entre a **lógica da inteligência artificial** e a **game-engine** do jogo, sendo utilizadas para **buscar informações** do que está acontecendo no jogo e **verificar se alguma condição é verdadeira ou falsa**.

Definições

Árvore de Comportamentos

Dois tipos de TAREFAS: *Condições e Ações*

Definições

Árvore de Comportamentos

Dois tipos de TAREFAS: Condições e Ações

CONDIÇÕES:

- As condições são utilizadas para realizar verificações do que ocorre na cena do jogo.
- Não realizam nenhuma mudança na cena.

EX: se o jogador está próximo ao player, verificar a quantidade de vida, armamento, etc.

Definições

Árvore de Comportamentos

Dois tipos de TAREFAS: Condições e Ações

CONDIÇÕES:

- As condições são utilizadas para realizar verificações do que ocorre na cena do jogo.
- Não realizam nenhuma mudança na cena.

EX: se o jogador está próximo ao player, verificar a quantidade de vida, armamento, etc.

AÇÕES:

- As ações são utilizadas para realizar mudanças no sistema.
- Existem dois tipos básicos de tarefas compostas, conhecidos como sequências e seletores.

EX: atacar o player, patrulhar, diminuir a vida do jogador, etc.

Definições

Árvore de Comportamentos

SEQUENCIAS:

- Possuem a responsabilidade de executar as subtarefas.
- Quando uma subtarefa obtém sucesso, a sequência passa a executar a próxima subtarefa.
- Quando uma subtarefa falha a execução a sequência é interrompida.

Definições

Árvore de Comportamentos

SEQUENCIAS:

- Possuem a responsabilidade de executar as subtarefas.
- Quando uma subtarefa obtém sucesso, a sequência passa a executar a próxima subtarefa.
- Quando uma subtarefa falha a execução a sequência é interrompida.

SELETORES:

- Executa todas as subtarefas, mesmo se uma subtarefa retornar uma falha na execução.

Definições

Árvore de Comportamentos

DECORETOR:

- Tem como objetivo permitir que sejam adicionados novos comportamentos sem que haja a necessidade de modificar o código original.
- Permite adicionar um comportamento a um objeto já existente em tempo de execução, ou seja, agrega dinamicamente responsabilidades adicionais a um objeto

Definições

Árvore de Comportamentos

DECORETOR:

- Acrescentar ou remover responsabilidades a objetos individuais dinamicamente, de forma transparente
- Evitar a explosão de subclasses para prover todas as combinações de responsabilidades
- Acrescentar responsabilidades a um objeto dinamicamente
- Prover alternativa flexível ao uso de subclasses para se estender a funcionalidade de uma classe

Definições

Árvore de Comportamentos

DECORETOR:

EX:

Colocar acessórios em uma arma (miras, silenciadores, etc.).

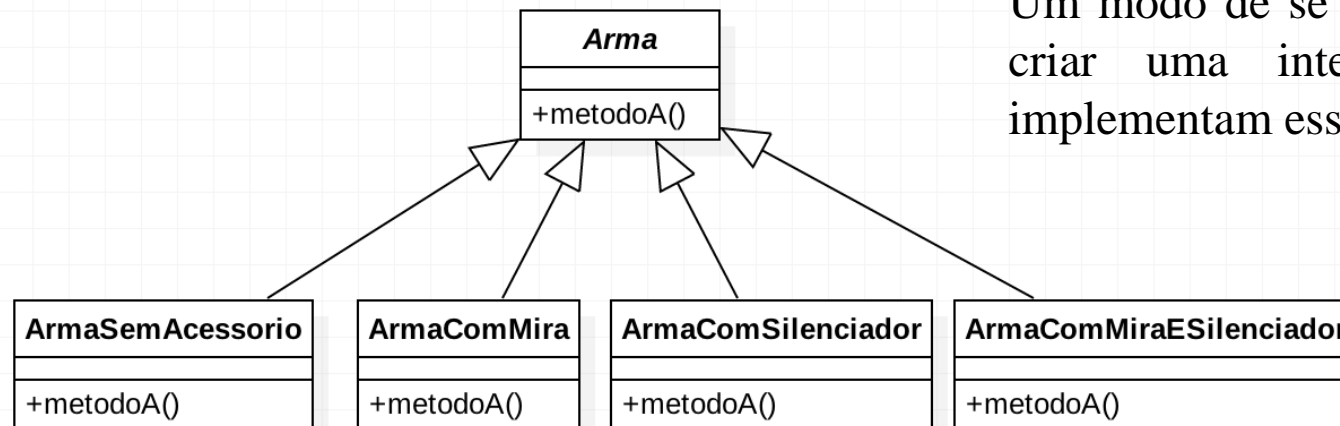
Definições

Árvore de Comportamentos

DECORETOR:

EX:

Colocar acessórios em uma arma (miras, silenciadores, etc.).



Um modo de se contornar esse problema seria criar uma interface e criar armas que implementam essa

Definições

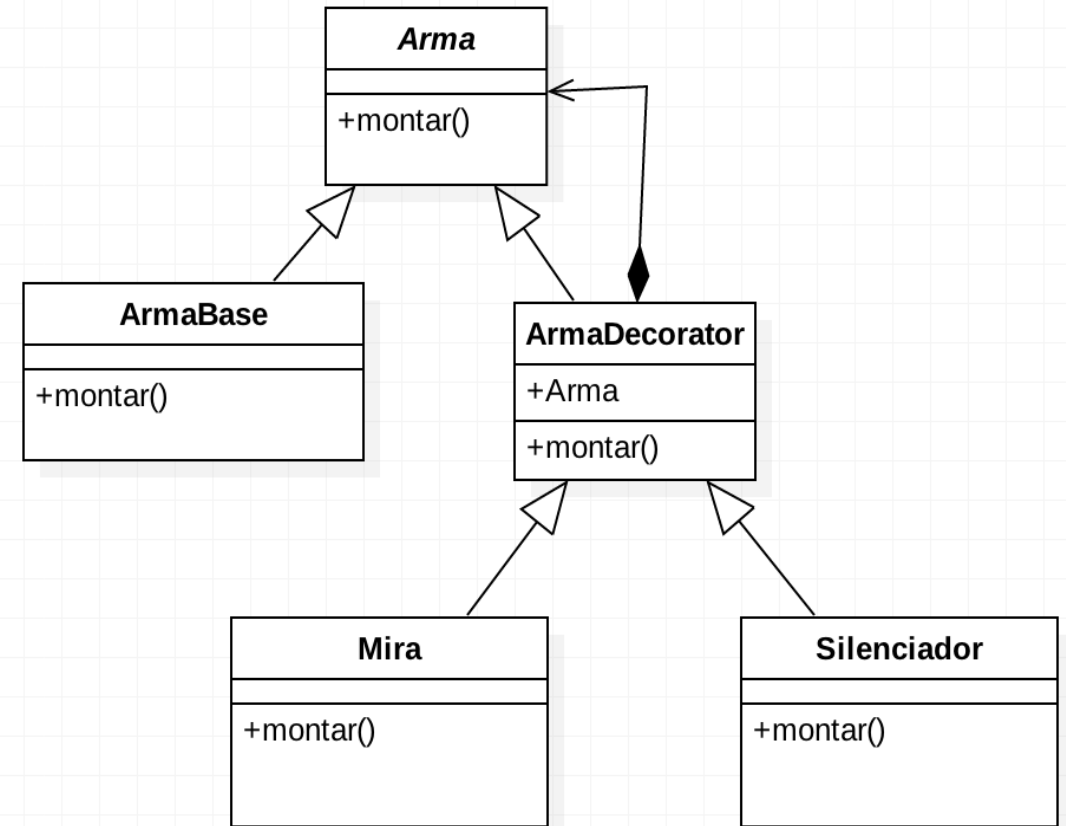
Árvore de Comportamentos

DECORETOR:

EX:

Colocar acessórios em uma arma (miras, silenciadores, etc.).

Design Pattern Decorator



Daniel Nogueira

dnogueira@ipca.pt