

**AAI – Atividade Auto-Instrucional**  
**Assunto: Criando Agentes Inteligentes para Jogos**

**I - Observações:**

1. **O trabalho é em Trio:** É permitido discutir com outros trios os problemas e as estratégias para solucioná-los, mas as implementações das soluções, bem como as resoluções dos exercícios propostos devem ser feitas somente pelos componentes do trio. Trabalhos plagiados da Internet ou de outros trios não serão corrigidos e receberão nota 0 (zero).
2. **Forma de Entrega:** Entrega Eletrônica via **SINEF**. O relatório deve ser entregue em formato PDF (outros formatos não serão corrigidos). Todos os arquivos fonte e o relatório devem ser enviados em um único arquivo ZIP (*não é RAR, nem TAR, etc!*) cujo nome deve ser: ***PrimeiroNome\_UltimoNome\_LAB1.zip*** onde ***PrimeiroNome*** e ***UltimoNome*** são o primeiro e o último nome de algum dos membros da dupla. **Qualquer entrega fora deste padrão será penalizada em 20% da nota.**
3. **Data Limite para Entrega: 09/06/2010 (Segunda-feira) até as 23:59 pelo SINEF**

**II - O que deve ser entregue:**

1. **Um relatório contendo:**
  - a) os nomes dos componentes do trio;
  - b) descrição das estratégias e algoritmos utilizados para o Agente.
  - c) análise dos resultados obtidos.
2. **O código fonte do Agente criado em java (somente o arquivo .java) e todos os outros arquivos que o trio tenha usado para compilar o agente.**

**III – Descrição da Atividade**

Nesta atividade você deverá desenvolver um agente capaz de agir racionalmente como um jogador do jogo Isola. Para isto, deverá usar as técnicas apresentadas na apresentação sobre “Busca com Adversários”, em especial, o algoritmo Minimax. Um servidor do jogo foi desenvolvido pelo professor da disciplina para facilitar o desenvolvimento dos trabalhos, uma vez que os alunos terão apenas que implementar o código dos agentes. Como uma parte do programa já está desenvolvida, você deve seguir **COMPLETAMENTE** as instruções dadas neste relatório, em sala de aula pelo professor e a documentação da interface fornecida juntamente com os programas de suporte.

**IV – Descrição do Jogo Isola**

Isola é um jogo de tabuleiro de estratégia para dois jogadores. É jogado em um tabuleiro de, geralmente, 7x7 posições, preenchidas por quadrados, exceto nas posições iniciais das peças dos jogadores. Ambos os jogadores possuem uma peça que é posicionada inicialmente no meio da linha mais próxima a ele, e no lado oposto da peça do oponente, como mostrado na Figura 1.

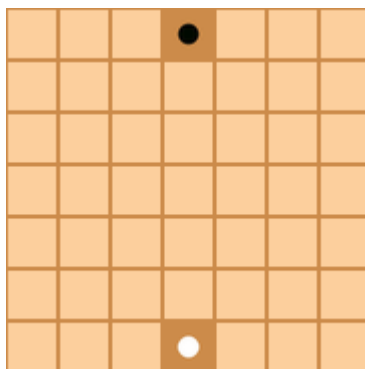


Figura 1 – Posição inicial do jogo Isola

Os jogadores se alternam as jogadas. Cada jogada consiste de duas ações, na seguinte sequência:

1. O jogador da vez move a sua própria peça para uma posição vizinha (na horizontal, vertical ou diagonal). A posição destino deve conter um quadrado e não pode conter a peça do oponente;
2. Depois da movimentação, o jogador da vez remove do jogo um quadrado que não tenha peças sobre ele.

Se no início de sua jogada, o jogador não puder se movimentar para nenhum quadrado, este perde o jogo.

#### **IV – Descrição do Servidor**

Um servidor simples foi desenvolvido para padronizar a apresentação dos agentes, permitindo com que agentes desenvolvidos por diferentes alunos possam competir entre si. Além disso, o servidor faz com que não seja necessário que você se preocupe em mostrar os resultados do tabuleiro, nem verifique a consistência das jogadas. Os métodos para estas tarefas já foram implementados.

Quando executado, o servidor pergunta o nome da classe dos agentes que jogarão a partida. Para se familiarizar com o jogo e com o servidor, execute o servidor e indique as classe do agente como AgentKeyboard, que permite que um ser humano jogue o jogo isola no lugar de um agente racional. Para executar o jogo, na linha de comando, entre no diretório onde você descompactou o arquivo zip fornecido pelo professor (um diretório acima do diretório ia) e digite a seguinte linha de comando:

```
java ia.isola.server.IsolaServer
```

O console terá uma aparência semelhante à da figura 2. Assim como nesta figura, você deve entrar o nome da classe do agente que irá jogar o jogo. Para este primeiro contato, use o AgentKeyboard. Quando o seu agente estiver pronto, ou para testá-lo, você deve entrar o nome da classe do seu agente neste campo. Uma vez que os agentes para os dois jogadores estiverem definidos, o servidor pergunta qual o tamanho do tabuleiro. Para tabuleiros maiores que 4x4, é preciso utilizar algumas técnicas para acelerar o processo de teste das jogadas. Por este motivo, neste trabalho os agentes devem executar, pelo menos, em tabuleiros de 4x4. Depois de entrado o tamanho do lado do tabuleiro, o jogo começa. O jogador 1 é sempre o primeiro a jogar e os jogadores alternam jogadas até que um perca o jogo.

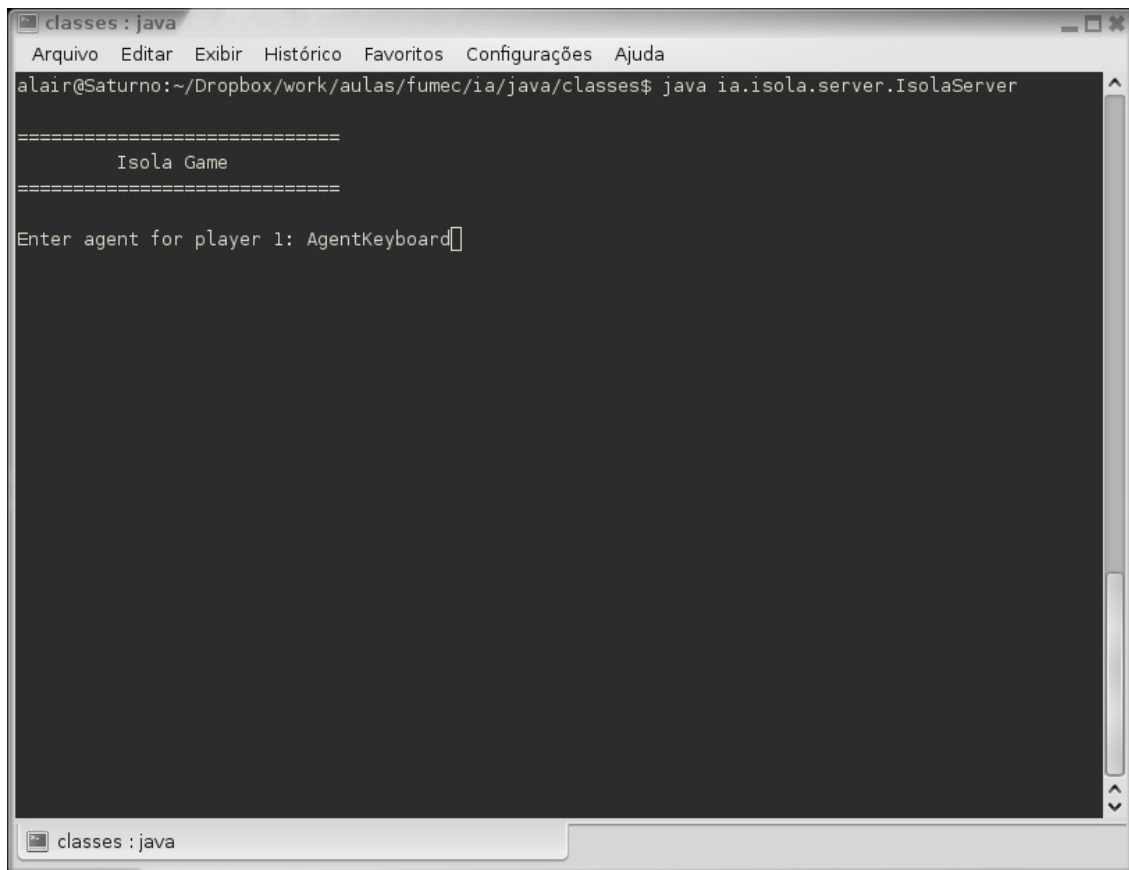


Figura 2 – Entrada da Classe do Agente

#### IV – Descrição dos Agentes

Você deverá criar apenas um arquivo que implementa uma classe de agente básica. Esta classe possui um método que é chamado pelo servidor do jogo, passando um objeto sobre o qual você fará suas jogadas. O código abaixo mostra a interface que você deve implementar.

```
package ia.isola.agents;  
import ia.isola.game.*;  
public interface BasicAgent  
{  
    public boolean move(final Isola game);  
}
```

Um programa de exemplo, de nome `AgentKeyboard`, foi fornecido e está no diretório `ia/isola/agents`. Trata-se de um agente que permite que as jogadas sejam entradas a partir do teclado. Além disso, um *template*, de nome `AgentTemplate.java`, também foi fornecido para facilitar o seu desenvolvimento. Crie o seu agente utilizando este *template* e salve-o com outro nome (seja criativo no nome do seu agente).

**IMPORTANTE:** Não modifique o nome do *package* do agente, e compile o seu programa no mesmo diretório em que se encontra o arquivo de *template*. O servidor busca pelo nome do agente neste diretório! Para compilar o programa, o seu *classpath* deve apontar um diretório acima do diretório "ia".

Uma explicação sobre a interface da classe Isola é dada no arquivo de documentação fornecido com os programas. Este arquivo foi gerado automaticamente pelo programa javadoc e é fornecido no formato HTML. Clique duas vezes sobre o arquivo index.html na pasta docs.

#### **IV – Requisitos para Construção do Agente**

O agente deve obedecer, além da interface fornecida, poucos requisitos. O primeiro é que o agente não pode gastar mais do que 30 segundos para fazer a sua jogada para tabuleiros de 4x4. Como o desempenho de diferentes computadores é diferente, o computador base será o do laboratório da FUMEC. Assim, seu agente deve fazer sua jogada em até 30 segundos para qualquer computador destes laboratórios. Se decorridos os 30 segundos e seu agente não tiver realizado a jogada, ele será desclassificado.

A outra regra que deverá ser obedecida: Não serão permitidos agentes que utilizem de má fé no tratamento da interface. O agente deve ser honesto em suas jogadas e não deve prejudicar o adversário (por exemplo, executando tarefas para reduzir o desempenho durante a jogada do adversário). O *fairplay* é um requisito para o seu agente.

Qualquer estratégia que não afete os princípios de *fairplay* pode ser usada em benefício do seu agente, mesmo que não tenha sido apresentada em sala de aula. Qualquer estratégia, no entanto, deve ser descrita, com referências, no relatório entregue juntamente com o código do agente.

BOM TRABALHO!