**Jogo Santiago**



**Mestrado Integrado em Engenharia Informática e Computação**

**Agentes e Inteligência Artificial Distribuída**

**João Manuel Ferreira Trindade ∙ 201109221 ∙** [ei11118@fe.up.pt](mailto:ei11118@fe.up.pt)

**Paulo Bordalo Marcos ∙ 201100759 ∙** [ei12131@fe.up.pt](mailto:ei12131@fe.up.pt)

**Rui Pedro Menezes da Rosa Neves ∙ 201109350 ∙** [ei11047@fe.up.pt](mailto:ei11047@fe.up.pt)

**Dezembro 2014**

# Índice

[Índice 2](#_Toc406341206)

[Objectivo 4](#_Toc406341207)

[Especificação 5](#_Toc406341208)

[Identificação e Caracterização dos Agentes 5](#_Toc406341209)

[Jogador Aleatório (Gambler) 5](#_Toc406341210)

[Jogador Gastador (High-Roller) 5](#_Toc406341211)

[Jogador Ganancioso (Greedy) 6](#_Toc406341212)

[Game Managing Agent 8](#_Toc406341213)

[Protocolos de Interacção 9](#_Toc406341214)

[Desenvolvimento 10](#_Toc406341215)

[Plataforma/Ferramentas Utilizadas 10](#_Toc406341216)

[Estrutura da Aplicação (módulos, diagrama de classes) 11](#_Toc406341217)

[Detalhes relevantes da implementação 13](#_Toc406341218)

[Experiencias 13](#_Toc406341219)

[Objectivo de Cada Experiencia 13](#_Toc406341220)

[Resultado de Cada Experiencia 13](#_Toc406341221)

[Conclusões 13](#_Toc406341222)

[Da analise dos resultados 13](#_Toc406341223)

[Do desenvolvimento do trabalho e aplicabilidade de SMA ao cenário proposto 13](#_Toc406341224)

[Melhoramentos 13](#_Toc406341225)

[Recursos 14](#_Toc406341226)

[Bibliografia 14](#_Toc406341227)

[Software 14](#_Toc406341228)

[Elementos do Grupo 14](#_Toc406341229)

[Apendice 14](#_Toc406341230)

[Manual do Utilizador 14](#_Toc406341231)

# Objectivo

Conceptualizado em 2003, Santiago é um jogo de tabuleiro para grupos de 3 a 5 jogadores de todas as idades. Neste jogo, os jogadores disputam entre si produções agrícolas e secções de terreno com o objective de no fim, ser o jogador com maior produção de alimentos.

Neste jogo, cada jogador assume o papel de um agricultor que pretende maximizar a sua produção agrícola, para isso, disputa leiloes com outros agricultores para conseguir as produções que pretende e para conseguir ter água nos seus terrenos.

O cenário deste projecto é o próprio jogo Santiago uma vez que é sobre ele que os nossos agentes vão interagir.

Este jogo tem como base um tabuleiro de 8 por 6 células com espaço para os canais de irrigação a cada duas células.

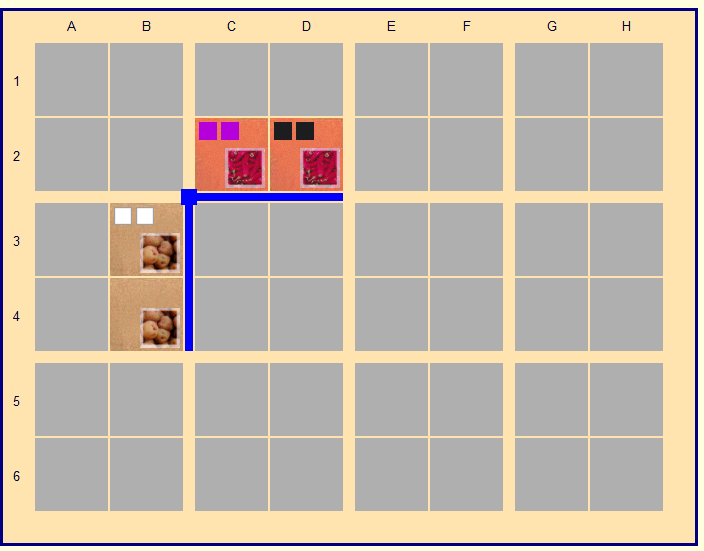


Figure 1 - Imagem do jogo Santiago Online de http://www.spielbyweb.com/

Neste exemplo temos dois canais já com água, representados a azul, e quatro plantações, os quadrados acima da imagem da produção representam as unidades de produção daquele terreno, e a sua cor indica o jogador que é dono do mesmo.

O jogo em si tem sete fases distintas:

1. Licitação sobre as plantações
2. Alterar o Administrador do Canal
3. Posicionar as plantações nos terrenos
4. Subornar o Administrador do Canal
5. Construção de Canais Extra
6. Secagem das plantações sem irrigação
7. Receber dinheiro

No nosso projecto todas estas fases serão desempenhadas por agentes no lugar dos jogadores, as regras de jogo estão pormenorizadamente descritas em (Hely & Pelek, 2003) apresentado na Bibliografia.

O objectivo deste projecto passa, em primeiro lugar, por implementar este jogo assente num sistema distribuído “multi-agente”. Desenvolveram-se com esse fim diferentes tipos de agente, com diferentes comportamentos, estratégias e processos de raciocínio. E estabeleceram-se regras de comunicação e negociação entre eles. Foi objectivo deste grupo criar um tipo de agente que tentasse recriar da forma mais fielmente possível o comportamento de um bom jogador de “Santiago”.  
Após a implementação ser dada por concluída, efectuamos um conjunto de experiencias para analisar o comportamento de cada um dos nossos agentes, a metodologia utilizada será descrita posteriormente neste relatório, mas os dados obtidos foram interessantes e permitiram retirar boas conclusões.

# Especificação

## Identificação e Caracterização dos Agentes

### Jogador Aleatório (Gambler)

Este agente tomará as suas decisões de acordo com uma função de aleatoriedade. Das decisões possíveis de tomar, escolherá aleatoriamente uma delas. Este agente serviu-nos como termo de comparação para a capacidade de tomar boas decisões dos outros restantes agentes. É teoricamente possível que este agente obtenha bons resultados mas estatisticamente improvável uma vez que cada jogo tem uma duração de 12 turnos, e em cada turno seria necessário que a “roleta das probabilidades” acertasse consistentemente nas melhores opções para as diversas decisões que o agente tem que tomar: quanto aposta, onde plantar, onde escolher o canal, com que quantia subornar o “channel overseer”, etc.

### Jogador Gastador (High-Roller)

O nosso agente high roller seguirá uma arquitectura de agente “reactivo-simples” pois terá um conjunto de regras “situação-acção” (Oliveira), o seu comportamento será gastar extravagantemente em cada ronda, grande parte do dinheiro que possuir.

O Jogador Gastador baseia-se no que se está a passar actualmente para decidir o que vai fazer. Ele vai olhar para o estado actual e mediante o número de jogadores que já tenha jogado e o seu dinheiro actual vai decidir qual a melhor acção a tomar. No caso do leilão de plantações, o agente tem uma probabilidade de passar o leilão, aliada ao seu dinheiro, em que se o seu dinheiro não for suficientemente alto para ter a certeza que ganha o leilão ele passa, senão “vai a jogo” (se a probabilidade for muito baixa de ir a jogo mas o agente Gastador tiver muito dinheiro ele escolhe ir a jogo, visto que o objectivo dele é gastar). Aqui, o agente olha para as licitações dos outros jogadores, se dois ou mais jogadores já tiverem licitado (será o terceiro, quarto ou quinto a jogar o que lhe pode dar mais possibilidade de ganhar o leilão) e baseia-se na máxima aposta feita pelos jogadores que já licitaram. Se a máxima aposta tiver sido fraca, o jogador cobre a aposta dando um pouco mais do que o máximo, se a máxima aposta for média o jogador aposta mais forte para ter a certeza que irá ganhar, e se a aposta máxima for alta o jogador aposta o que tem (ALL IN) para garantir que ninguém irá colocar licitações acima da dele. Se não houver jogadores que tenham feito licitação ele baseia-se apenas no seu dinheiro para fazer a aposta. Se tiver pouco, ele passa para tentar ser Gestor de Canal e ganhar dinheiro para o próximo round, senão “vai a jogo” com uma percentagem do seu dinheiro.

Na fase de plantação, este agente tentará sempre escolher a plantação com mais trabalhadores ( o que lhe dará mais pontos) e colocá-la junto de um canal com água (terreno irrigado). Se isto não for possível o agente baseia-se no algoritmo de Monte Carlo para escolher a posição do seu terreno.

Para fazer o suborno, este agente continua com uma estratégia parecida com a estratégia de licitação sendo que se tiverem feito um suborno dois ou mais jogadores ele baseia-se no valor do máximo suborno para fazer o seu suborno e se não houverem jogadores que já tenham colocado o seu suborno este agente dá o máximo que puder para tentar colocar o canal onde lhe der mais jeito (possíveis terrenos que tenha plantado e não estejam irrigados).

Finalmente, na fase de escolher onde colocará o canal com água o agente verifica se tem algum terreno dele que tenha trabalhadores e não esteja irrigado, onde seja possível colocar água. Se for possível será aí que ele colocará o canal, senão, verifica se existe algum terreno sem trabalhadores e coloca-o lá, visto que pode dar-lhe mais pontos. Caso nenhuma destas duas acções seja possível o agente tentará colocar mais água no seu último terreno plantado com o intuito de poder plantar a seguir junto a um terreno seu e ganhar assim mais pontos.

### Jogador Ganancioso (Greedy)

O Agente Greedy um Agente Baseado em Utilidade. Dado um estado de jogo, ele vai gerar possíveis estados sucessores a esse, avalia-os e decide executar aquele com melhor avaliação, ou «medida de satisfação relativamente aos diversos estados» (Oliveira).  
A designação de “Greedy” vem do comportamento ganancioso que este agente vai ter em cada vez que seja interpelado para tomar uma decisão. Foi neste tipo de agente em que o grupo dedicou mais tempo e recursos de forma a conseguir que este recorresse a heurísticas que aproximassem o seu comportamento daquele que teria um jogador humano.  
Alguns dos seus processos de raciocínio e estratégias podem ser traduzidos nos seguintes diagramas de actividade

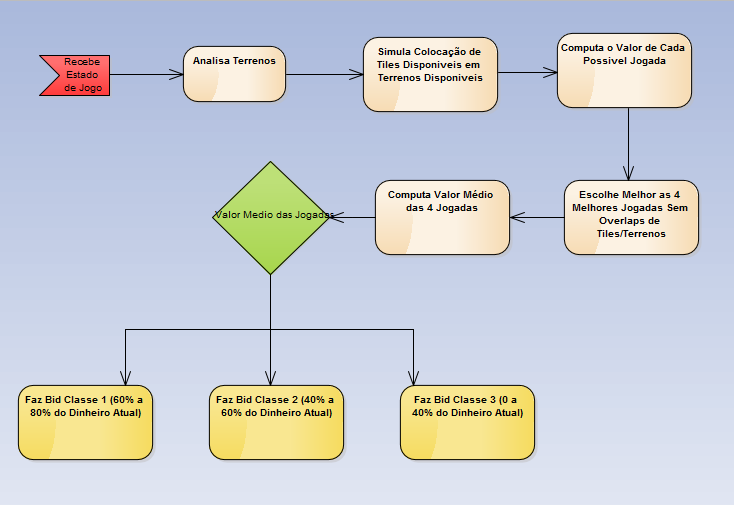


Diagrama de Actividade 1

Neste diagrama de actividade apresenta-se a estratégia do agente ganancioso para determinar que valor deve investir num leilão de plantações de forma a não investir recursos em plantações que depois lhe tragam recompensas baixas. Neste caso particular o agente gera várias combinações de “Plantação disponível para compra” em lotes de terrenos ainda inutilizados. Calcula o valor pontual de todas as jogadas possíveis, escolhe as quatro melhores que não sejam mutualmente exclusivas (com quatro plantações diferentes em quatro terrenos diferentes) e com base no valor pontual médio dessas quatro jogadas decide entre qual das 3 “categorias de apostas” deve utilizar. De seguida o valor final é determinado aleatoriamente dentro dos valores aceites pela categoria.

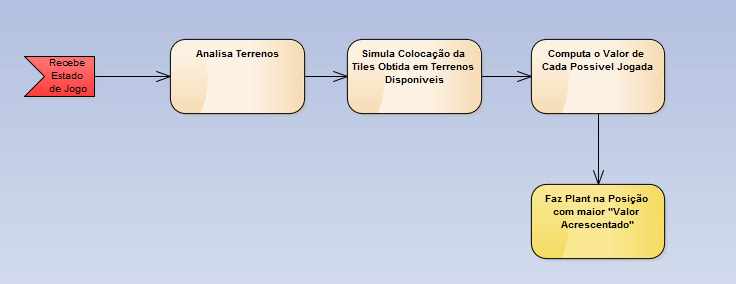


Diagrama de Actividade 2

Em Diagrama de Actividade 2 descreve-se o processo que o agente ganancioso utiliza para decidir em que posição livre do tabuleiro deve colocar a plantação que ganhou no leilão. A semelhança do caso anterior, geram-se combinações “Plantação” – “Terreno Disponível” e selecciona-se aquela cujo valor acrescentado para o jogador seja maior. O cálculo deste valor acrescentado tem em conta que a colocação de uma plantação junto de um grupo da mesma plantação funciona como multiplicador de pontos, algo que iremos descrever posteriormente neste relatório.

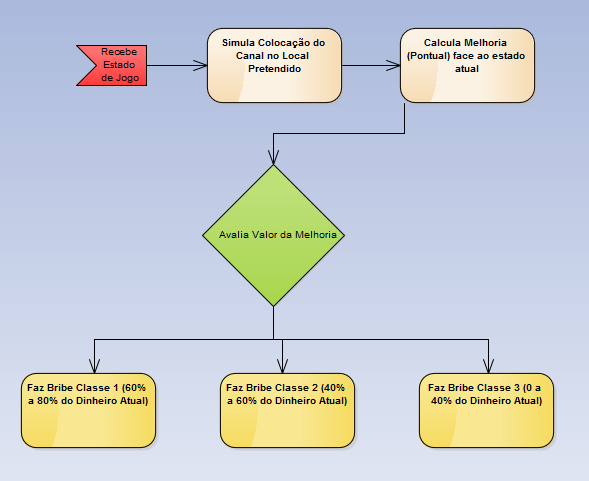


Diagrama de Actividade 3

No Diagrama de Actividade 3 descreve-se o processo de suborno do gestor de canal, muito semelhante ao processo para determinar a licitação no leilão, computa-se a melhoria a existência de um canal na posição escolhida para um jogador, e com base nesse valor categorizamos o valor que o suborno deve ter. De seguida o valor final é determinado aleatoriamente dentro dos valores aceites pela categoria.

### Game Managing Agent

Ao contrário dos anteriores agentes, o Game Managing Agent não desempenhará um papel de jogador, a sua função será apenas coordenar os momentos de jogo com todos os jogadores. Entre outras coisas, será ele o responsável por pedir aos jogadores para fazerem a sua próxima jogada.

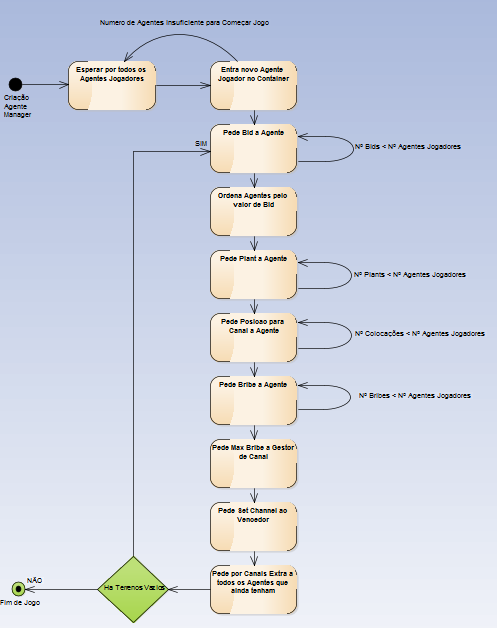


Diagrama de Actividade 4

Em Diagrama de Actividade 4 apresenta-se o processo de funcionamento do agente gestor de jogo. Ele coordena todos os momentos do jogo, passando a fase seguinte apenas quando tiver obtido a resposta de todos os jogadores ao pedido da fase actual.

## Protocolos de Interacção

Para a interacção entre os agentes, procedemos ao envio de mensagens segundo o *standard* proposto pela “Foundation for Inteligent Physical Agents” em (FIPA ACL Message Structure Specification, 2002). Seguimos também outra especificação proposta pela mesma fundação para a identificação de cada mensagem de forma a garantir a interoperabilidade e distinguir claramente diferentes processos numa comunicação inter-agente utilizando diferentes “labels” para cada mensagem, designadas “Performativas” (FIPA Communicative Act Library Specification, 2002)

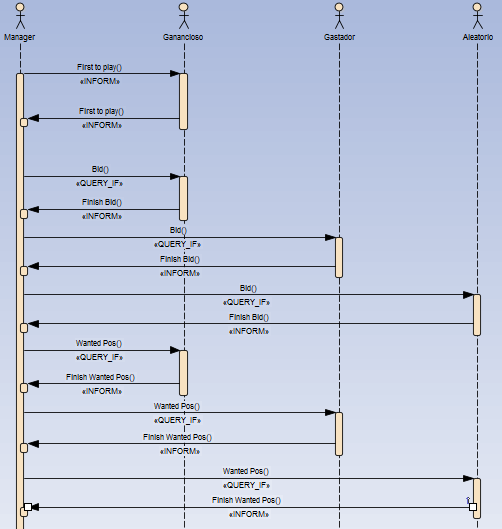


Diagrama de Interacção 1 – Parte 1 de 2

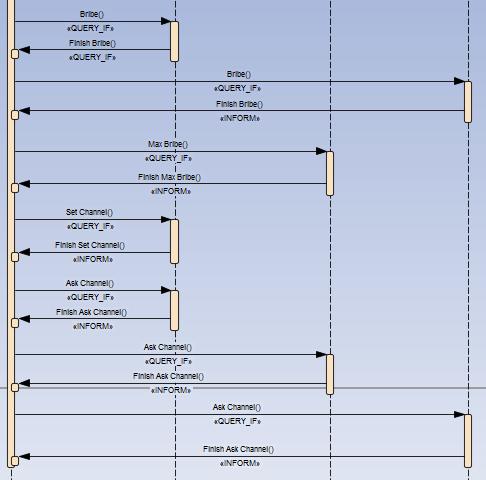


Diagrama de Interacção 2 – Parte 2 de 2

# Desenvolvimento

## Plataforma/Ferramentas Utilizadas

Este projecto tirou especial partido da framework “Jade”, desenvolvida e mantida por

Jade é uma framework criada com o proposito de facilitar o desenvolvimento de sistemas multi-agente. Permite entre outras coisas, a criação, execução e comunicação entre agentes (seguindo as normas da FIPA para a comunicação) de uma forma prática e num ambiente de programação em Java.

Algumas das carecteristicas mais interessantes desta framework são (Bellifemine, Caire, & Greenwood, 2004):

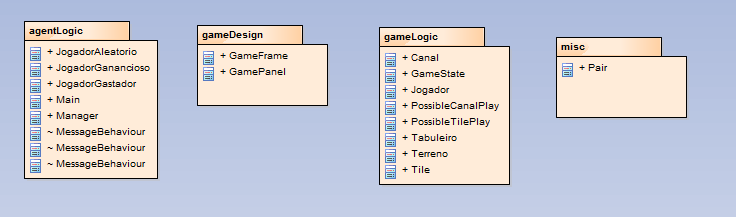
* API User Friendly em Java
* Possibilidade de ter um sistema totalmente distribuído – diferentes threads em diferentes máquinas – em que toda a infraestrutura de comunicação é abstraída para o programador
* Segue as normas e especificações da FIPA
* Suporte á mobilidade dos agentes, ou seja, é possível mover um agente de uma maquina para outra de forma transparente
* Ferramentas gráficas que facilitam ao programador tarefas de *debugging*

O projecto foi desenvolvido na linguagem de programação Java em Windows, no entanto não deverão haver entraves à sua execução em outros sistemas operativos. Foi utilizada a biblioteca “commons.org.lang” desenvolvida pela Oracle, para tirar partido de métodos que facilitam a cópia de objectos serializáveis.

O IDE utilizado pelo grupo foi o Eclipse Luna.

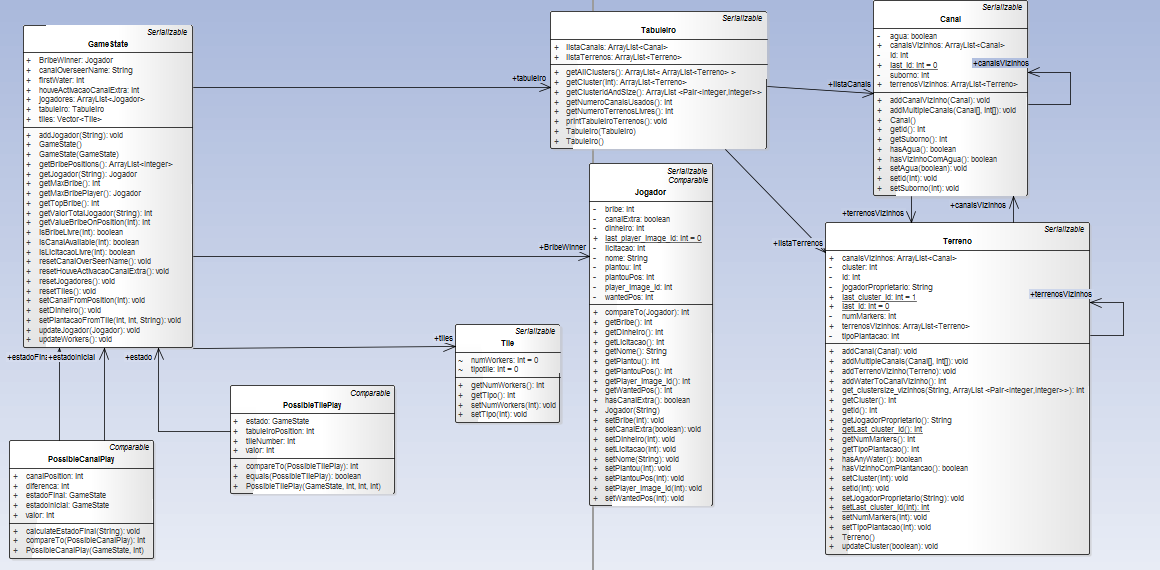
## Estrutura da Aplicação

A aplicação encontra-se organizada em 4 packages, um com todos os agentes, um outro package para toda a logica do jogo, organização do tabuleiro, dos jogadores, etc. um terceiro package para a interface gráfica e um ultimo com utilidades extra.



Package Diagram 1

Dentro do package gameLogic temos toda a logica que rege o jogo, o diagrama de classes na página seguinte representa as classes desse mesmo package.

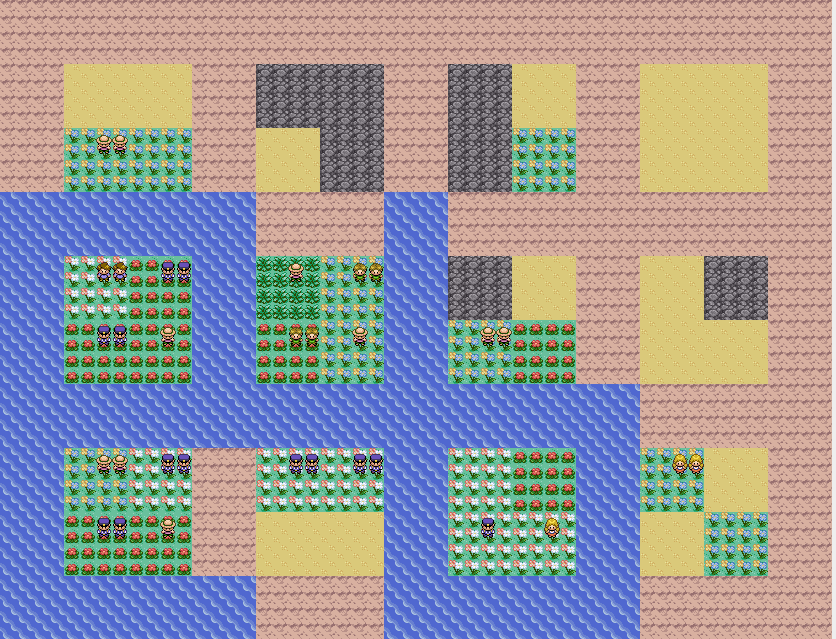


Class Diagram 1

## Detalhes relevantes da implementação

Alguns pormenores interessantes da nossa implementação são por exemplo:

* Utilização de uma estrutura grafo bi-conexo para a representação do tabuleiro e de todas as ligações entre terrenos e canais de água.
* Utilização de um algoritmo flood-fill para resolver o problema do agrupamento de terrenos do mesmo tipo em clusters de terrenos. Um problema que á primeira vista parecia trivial mas que posteriormente se demonstrou um desafio interessante.
* Design da interface gráfica representativa do estado do jogo, dos seus intervenientes e da evolução do preenchimento do tabuleiro.
* Desenvolvimento de um módulo para recolha de dados estatísticos mais facilmente e sem necessidade de um utilizador a recolher os dados manualmente



Interface Gráfica 1

Interface Gráfica 1 : um exemplo de uma representação de tabuleiro na interface gráfica desenvolvida para o projecto.

# Experiencias

## Objectivo de Cada Experiencia

Após a criação dos diversos agentes, era necessário efectuar experiências para se retirarem melhores conclusões sobre qual a melhor estratégia para ganhar no jogo Santiago.   
  
Foram efectuados os seguintes testes:

1. 5 Jogadores, 1 Aleatório, 2 Gananciosos e 2 Gastadores.
2. 5 Jogadores, 5 Gananciosos.
3. 3 Jogadores, 1 Aleatório, 1 Ganancioso e 1 Gastador.

O teste número 1 e 3 servem para verificar a interacção entre os diversos tipos de agentes e como se comportam nas várias alterações que são feitas ao estado do jogo no decorrer das rondas. Serve também para verificar a raridade do estado de vitória do jogador Aleatório, que se defronta com adversários mais inteligentes e arrojados.

O teste número 2 mostra como se comportam vários jogadores inteligentes a jogar entre si e qual o agente que mais vitórias conseguiu no conjunto de jogos.

Ambos os testes dão uma ideia de média de valor para se sair vitorioso e qual a maior pontuação conseguida.

## Resultado de Cada Experiencia

### Teste 1 [5 Jogadores, 1 Aleatório, 2 Gananciosos e 2 Gastadores.]

Este teste foi efectuado em 2 partes de 50 jogos cada uma.   
  
Na 1ª Parte de 50 jogos obtivemos os seguintes resultados:



Na 2ª Parte de 50 jogos obtivemos os seguintes resultados:

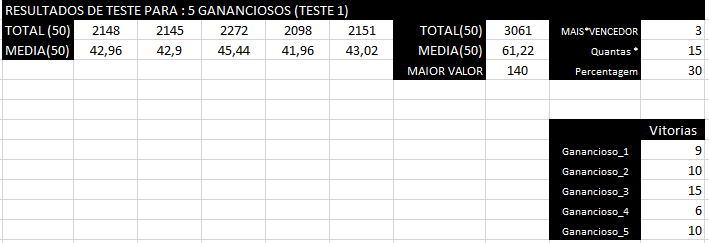


Como podemos verificar os resultados em ambas as partes mostraram-se bastante coerentes.   
O jogador Aleatório nunca conseguiu vencer em nenhum dos 100 testes, os jogadores Gastadores em conjunto conseguiram uma percentagem de 2.5% de vitórias no conjunto dos 100 testes e os jogadores Gananciosos dominaram repartindo sempre entre si o número de jogos ganhos.

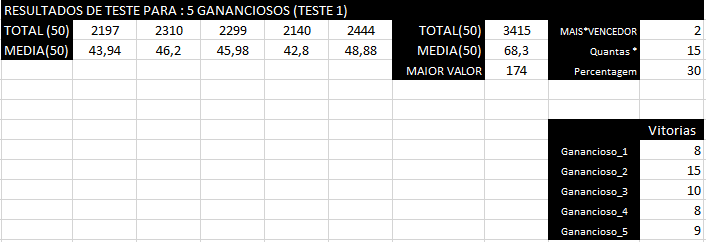
A Media de valor para se vencer e o valor máximo de vitória, estão muito próximos nos dois casos e bastante coerentes com o esperado.

### Teste 2 [5 Jogadores, 5 Gananciosos.]

Este teste foi efectuado em 2 partes de 50 jogos cada uma.   
  
Na 1ª Parte de 50 jogos obtivemos os seguintes resultados:



Na 2ª Parte de 50 jogos obtivemos os seguintes resultados:

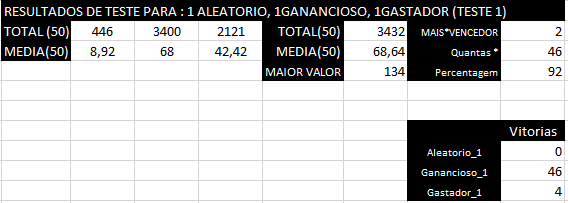


Como podemos verificar os resultados em ambas as partes mostraram-se bastante coerentes.   
Os jogadores Gananciosos repartiram sempre entre si o número de jogos ganhos.

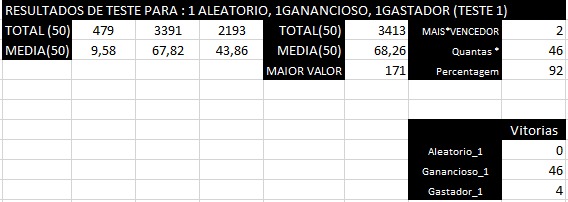
A Media de valor para se vencer e o valor máximo de vitória, estão muito próximos nos dois casos e bastante coerentes com o esperado.

### Teste 3 [3 Jogadores, 1 Aleatório, 1 Ganancioso e 1 Gastador.]

Este teste foi efectuado em 2 partes de 50 jogos cada uma.   
  
Na 1ª Parte de 50 jogos obtivemos os seguintes resultados:



Na 2ª Parte de 50 jogos obtivemos os seguintes resultados:



Como podemos verificar os resultados em ambas as partes mostraram-se bastante coerentes.   
O jogador Aleatório nunca consegui vencer em nenhum dos 100 testes, o jogador Gastador conseguiu uma percentagem de 2.0% de vitórias no conjunto dos 100 testes e o jogador Ganancioso dominou o número de jogos ganhos.

A Media de valor para se vencer e o valor máximo de vitória, estão muito próximos nos dois casos e bastante coerentes com o esperado.

# 

# Conclusões

## Da análise dos resultados

Os valores obtidos em todos os testes estão de acordo com o esperado. Os jogadores Gananciosos demonstram-se bastante eficazes mostrando-se como os mais vezes vencedores. Em segundo lugar aparecem os Gastadores que conseguem por vezes ganhar alguns jogos mas tornando-se muito esporádico e com as suas pontuações sempre ameaçadas pelas dos jogadores gananciosos. O jogador aleatório, neste número de testes, nunca conseguiu mostrar o seu valor e ganhar pelo menos um jogo.

## Do desenvolvimento do trabalho e aplicabilidade de SMA ao cenário proposto

Ter a oportunidade de desenvolver este projecto foi extremamente gratificante para todos os elementos do grupo, notamos arquitecturas como a de agentes distribuídos são extremamente poderosas e, no entanto, a sua complexidade de implementação consegue ser bastante reduzida por frameworks como é o caso do Jade.  
Este tipo de arquitectura aplicada a um jogo de tabuleiro foi algo que todos achamos curioso, e fora do vulgar. Normalmente tecnologias como estas aplicam-se a cenários da vida real, a simulações, etc. No entanto, foi essa díspar característica cativou-nos a escolher este projecto, tal era a determinação em fazer algo diferente. Concluído o projecto, ficamos bastante impressionados com a qualidade da solução que conseguimos apresentar, e com a forma como conseguimos aplicar uma arquitectura de agentes distribuídos para simular um vulgar jogo de tabuleiro.

# Melhoramentos

Numa futura iteração deste projecto haveriam certas *features* que gostaríamos de acrescentar de forma expandir a nossa solução, nomeadamente:

* Adição de novos tipos de agentes com características inovadoras
* Desenvolvimento do agente ganancioso numa tentativa de melhorar ainda mais a sua *performance*.
* Desenvolvimento de uma interface que permitisse um jogo entre jogadores humanos e agentes, para extracção de conhecimento e optimização futura dos agentes.

# Recursos

## Bibliografia

Bellifemine, F., Caire, G., & Greenwood, D. (2004). Developing multi-agent systems with Jade. In F. Bellifemine, G. Caire, & D. Greenwood, *Developing multi-agent systems with Jade* (pp. 30-32). Wiley.

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. (03 de 12 de 2002). *FIPA ACL Message Structure Specification.* Obtido de FIPA ACL Message Structure Specification: http://www.fipa.org/specs/fipa00061/SC00061G.html

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. (03 de 12 de 2002). *FIPA Communicative Act Library Specification.* Obtido de http://www.fipa.org/specs/fipa00037/SC00037J.html

Hely, C., & Pelek, R. (2003). *Santiago, The River of Money determines the Course of the Canals.* Obtido de boardgamegeek.com: http://boardgamegeek.com/filepage/4075/santiago-english-rules-11pdf

Oliveira, E. (s.d.). *Capítulos 1, 2abc Introdução; Arquitecturas básicas de Agentes; Lógicas.* Obtido de http://paginas.fe.up.pt/~eol/AIAD/aulas/1415/AIAD1\_2abc.pdf

## Elementos do Grupo

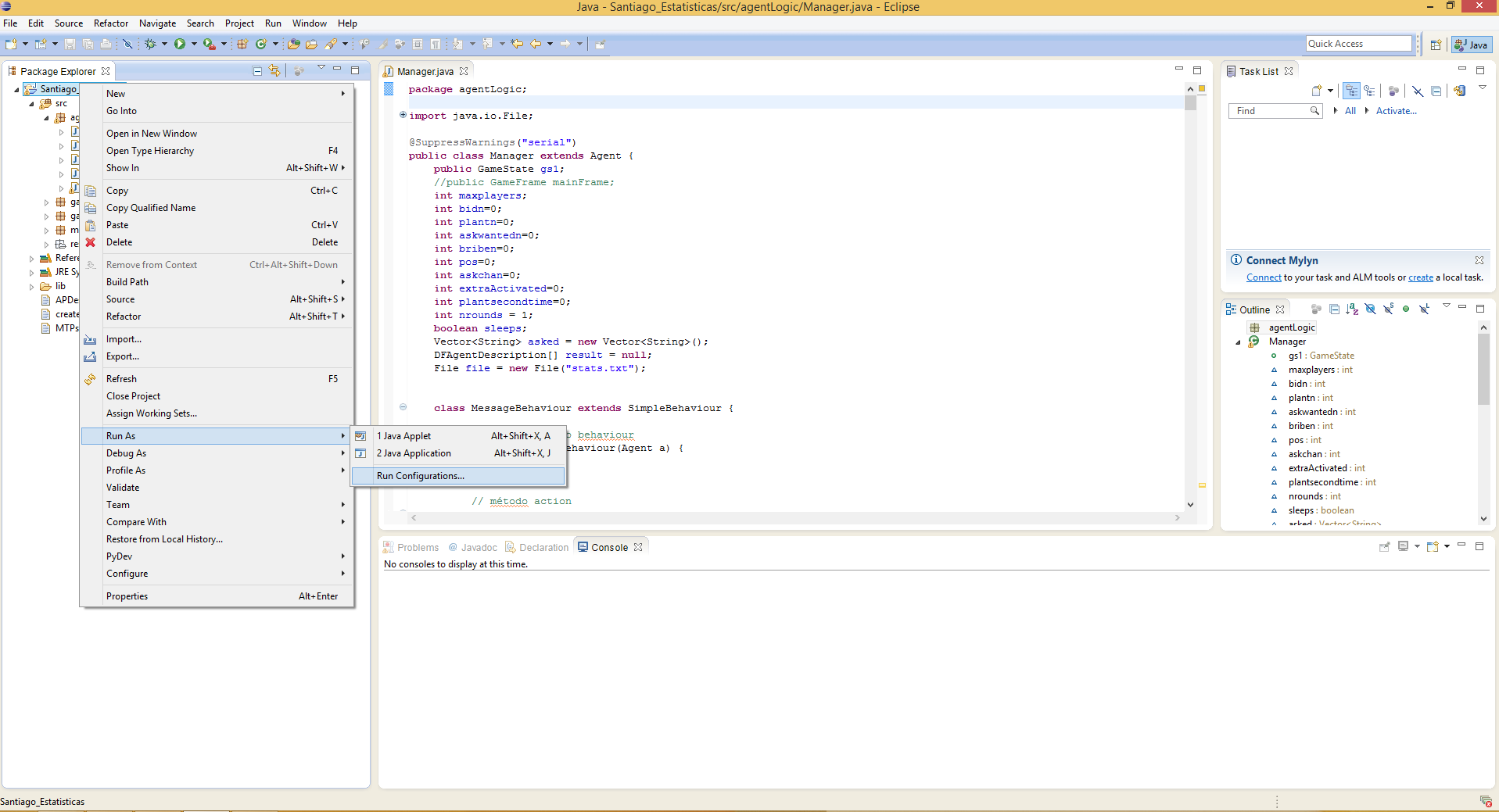
Todos os elementos do grupo trabalharam de igual forma e dedicada para o projeto.

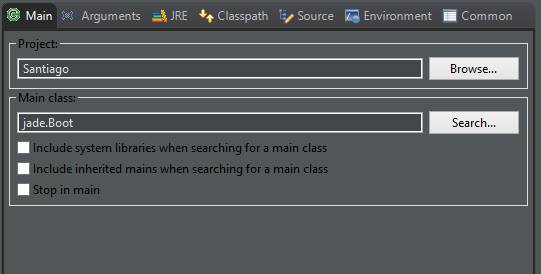
Numa perspetiva mais geral, cada elemento do grupo trabalhou: 33.(3)%.

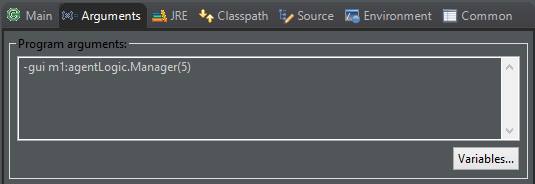
# Apêndice

## Manual do Utilizador

No nosso projeto integrado com GUI (Santigo) apenas temos que importá-lo e executá-lo com as seguintes “Run Configurations”:



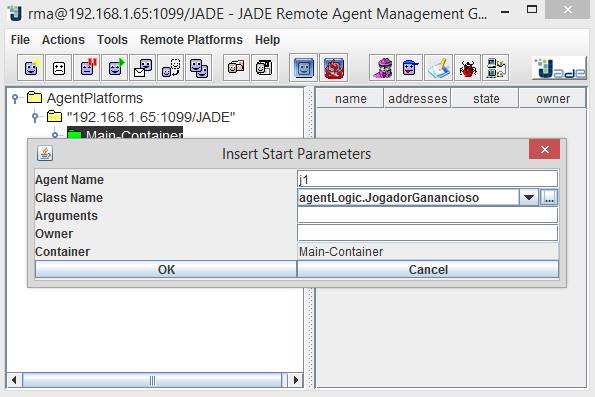




Depois corremos o projeto e a interface do JADE irá aparecer.

Escolhemos o símbolo para criar agente e criamos os nossos agentes Jogadores de entre as seguintes classes:

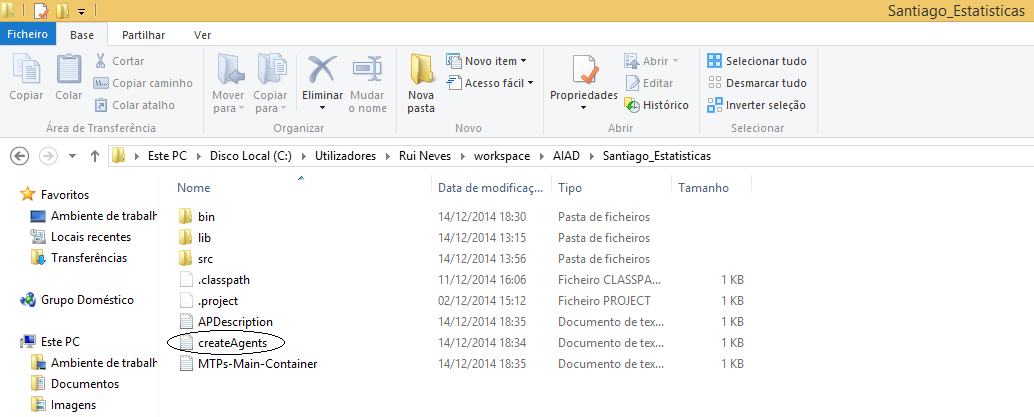
* agentLogic.JogadorAleatorio
* agentLogic.JogadorGastador
* agentLogic.JogadorGanancioso



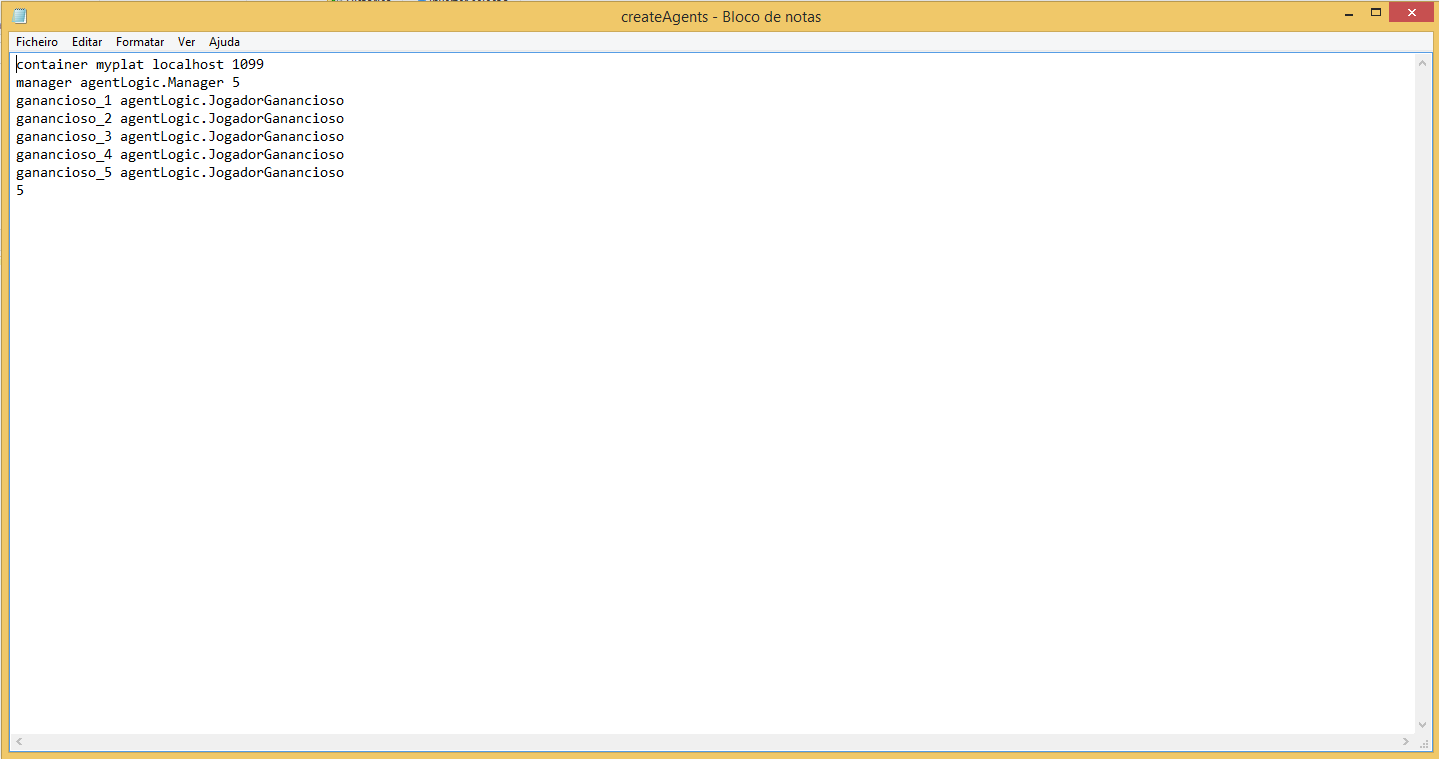
Após criar o número de jogadores enviado pelos argumentos do nosso Gestor de Rondas (agentLogic.Manager) o jogo começará automaticamente e verá as jogadas graficamente.



Para o modo de Estatísticas (Santiago\_Estatisticas) precisamos de ter um ficheiro .txt na pasta do projeto com o nome createAgents.txt



Com a seguinte configuração:

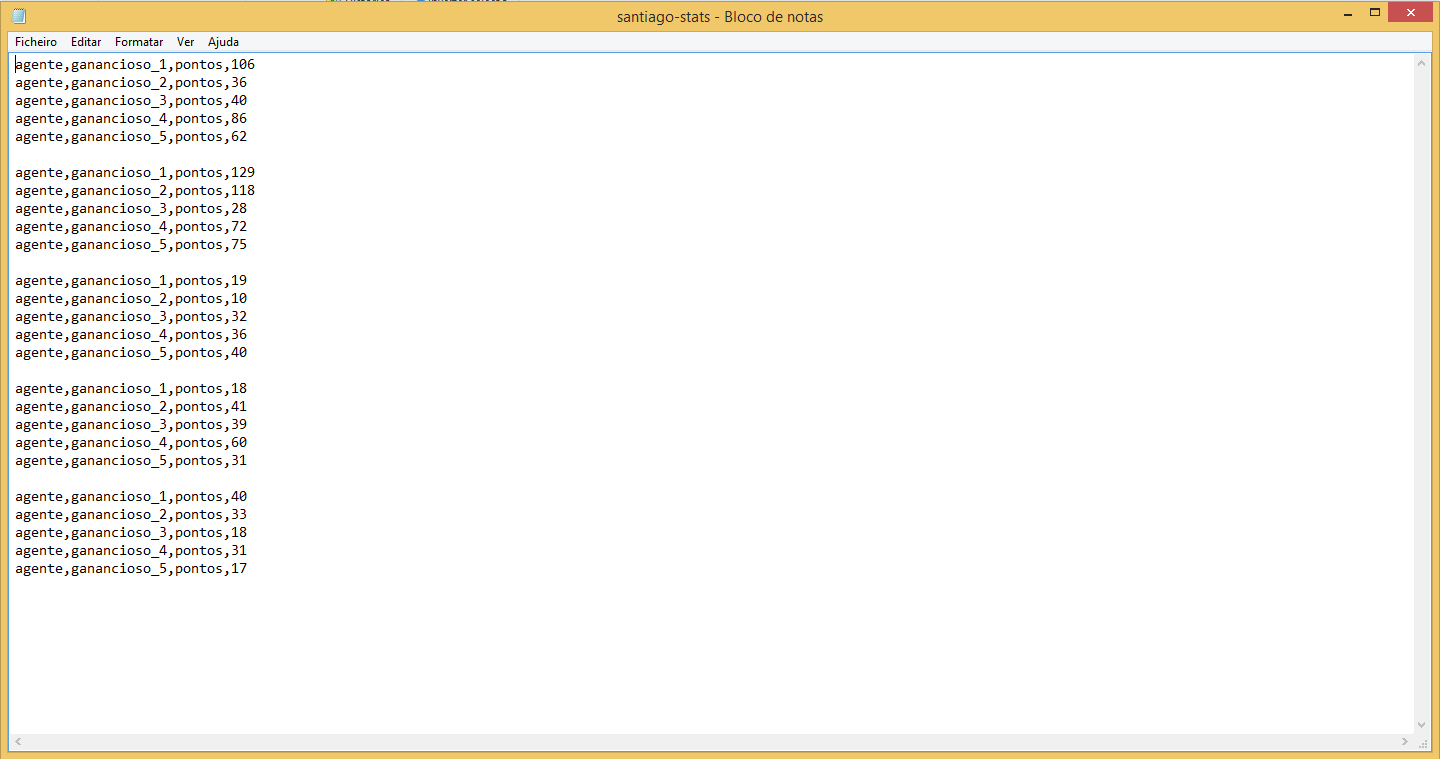
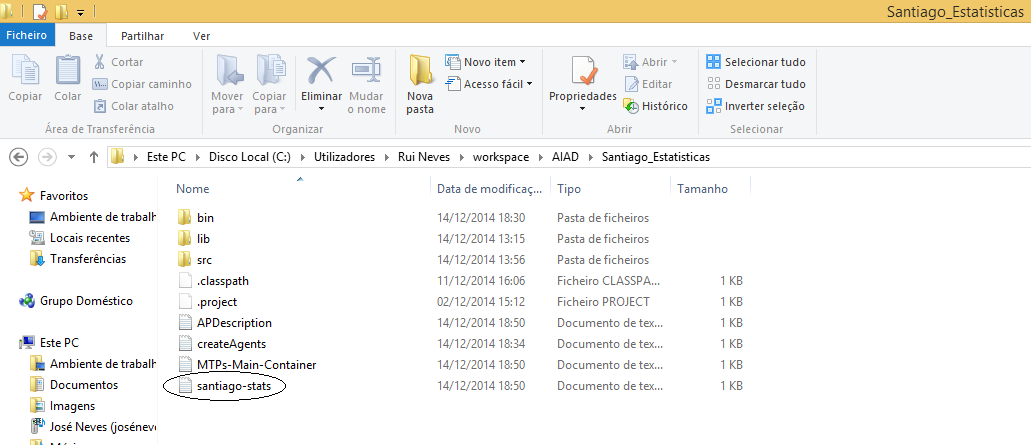


Em que na primeira linha definimos o container, platform-ID, host e porta.

Na segunda linha criamos o nosso agentGestor de jogo (nome seguido de classe seguido de argumento -> número de jogadores que vão jogar)

Nas próximas linhas definimos os jogadores e na última linha temos o número de jogos que queremos correr para aqueles jogadores.

No final será criado um .txt na pasta do projeto com as informações relativas a cada jogo com o nome santiago-stats.txt



Indicando o nome do jogador e os seus pontos.