

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

PERFIL **Sistemas Inteligentes**

UNIDADE CURRICULAR DE **Computação Natural**

EDIÇÃO 2016/2017

COMPUTAÇÃO INTELIGENTE

AUTORES:

Diana Oliveira (a67652)

Gil Gonçalves (a67738)

Pedro Duarte (a61071)

Pedro Lima (a61061)

Braga, 3 de Abril de 2017



Resumo

O presente relatório apresenta a descrição do desenvolvimento de um trabalho teórico-prático, realizando em grupo, relacionado com as temáticas abordadas nas aulas de Computação Natural, em particular a Computação Especulativa contendo as opções de desenvolvimento do caso de estudo e considerações realizadas no protótipo.

O objetivo seria o desenvolvimento e implementação de um sistema capaz de prever eventos e reagir a esses mesmos eventos, mesmo quando a informação recolhida fosse desconhecida ou incompleta.

Conteúdo

0.1	Introdução	2
1	Caso de estudo	3
1.1	Enquadramento geral do Tema	3
1.2	Regras do Jogo	3
1.3	Posições	4
1.4	Ações	4
1.5	Rondas de Apostas e desenvolvimento de uma mão	5
1.6	Ranking de Mãos	8
2	Fases da Computação Especulativa	12
2.1	O que é a computação especulativa?	12
2.2	Formalização da <i>Framework</i>	12
2.3	Definições preliminares	14
2.4	Fases da Computação especulativa	15
2.4.1	Redução de processos	15
2.4.2	Obtenção de Factos	16
2.4.3	Revisão de valores por defeito	16
3	Implementação	17
3.1	Agentes	17
3.2	Comunicação entre os agentes	18
3.3	Menu Inicial	19
3.4	Informação trocada entre os agentes	20
4	Conclusão	21

0.1 Introdução

A tomada de decisão pode ser vista um processo cognitivo pelo qual se escolhe um plano de ação entre vários outros, que pode ser baseado em vários cenários, ambientes, análises e fatores, para uma situação-problema. Todo este processo produz uma escolha/-decisão final em que a “saída” pode ser uma opinião ou uma ação. Neste sentido a tomada de decisão é fundamental para a maioria das ações que realizamos diariamente de acordo com a situação e/ou problema com que nos deparamos.

Este processo gera uma escolha/decisão final podendo ser uma opinião ou uma ação. Para termos a perceção de todos os cenários possíveis numa tomada de decisão é necessário que tenhamos a possibilidade de especular sobre todos os casos, de forma a que possamos medir os prós e os contras consoante a ação que levamos a cabo.

O trabalho descrito no decorrer deste documento tem como objetivo ser capaz, de acordo com uma determinada situação, especular todos os cenários possíveis de modo a produzir decisões sendo necessário identificar um domínio de aplicação para formalizar um problema de acordo com a *framework* $(\Sigma, \varepsilon, \Delta, P)$ fornecida.

Posteriormente é descrito o caso de estudo que permite observar os efeitos de todas as fases da Computação Especulativa, nomeadamente: redução de processos, chegada de factos e revisão de valores por defeito assim como as formas adequadas de implementação para obtenção dos valores por defeito no contexto do caso de estudo.

Capítulo 1

Caso de estudo

1.1 Enquadramento geral do Tema

O tema escolhido para desenvolvimento foi uma simulação de um jogo de cartas **Poker**.

O objetivo seria prever as movimentações dos jogadores através de informações incompletas/desconhecidas, sendo capaz de tomar decisões nas jogadas.

1.2 Regras do Jogo

No jogo de cartas Poker jogamos numa mesa onde se podem sentar desde dois jogadores até dez. Neste jogo, poderás ter uma quantidade de fichas, ou dinheiro que te servirá para realizar as apostas. Através destas apostas, irão se formar potes em cada mão que se joga. O objetivo deste jogo é ganhar esses potes e assim aumentar a tua quantidade de fichas ou dinheiro.

Podemos ganhar um pote de duas formas:

- Quando no final de cada jogada, os jogadores ainda envolvidos mostrarem as suas cartas e a nossa mão seja mais forte que a(s) do(s) adversário(s).
- Quando durante a mão tivermos realizado apostas e todos os nossos adversários abandonarem sem chegarmos à fase em que se mostram as cartas de cada um.

O funcionamento do jogo é simples: São dadas duas cartas tapadas (privadas) a cada jogador e irão sair até 5 cartas, visíveis a todos (cartas comunitárias), na mesa durante as rondas de apostas. Dado que uma mão de Poker é composta por cinco cartas, escolherá as cinco que formam a melhor combinação entre as suas duas cartas tapadas e as cinco que saem na mesa.

1.3 Posições

Small Blind: A *Small Blind* é a posição imediatamente à esquerda da última posição. O jogador que está neste lugar está obrigado a colocar uma aposta “às cegas” e daí o seu nome.

Big Blind: A *Big Blind* é a posição logo à esquerda da *Small Blind* e por isso estão as duas Blinds à esquerda da Última posição. Como no caso da *Small Blind*, a *Big Blind* tem de fazer uma aposta “às cegas”. O valor desta aposta dependerá do tipo de mesa em que estiveres a jogar, e será sempre o dobro da aposta da *Small Blind*.

Última posição (Botão): Esta posição é a mais importante da mesa já que marca a ordem dos restantes jogadores. Logo à esquerda encontra-se a *Small Blind* e logo a seguir a *Big Blind*. A última posição irá mover-se para a esquerda em cada nova jogada (rodando à volta da mesa). Ou seja, a *Small Blind* será a última posição na jogada seguinte. O jogador que estava na última posição ficará agora na penúltima posição e assim sucessivamente.

1.4 Ações

No Poker teremos, habitualmente, três ações possíveis quando for a nossa vez de tomar uma decisão: subir uma aposta (*raise*), igualar uma aposta (*call*) ou abandonar perante uma aposta (*fold*). Em algumas situações, quando qualquer outro jogador não tiver apostado, em lugar de subirmos, faremos uma aposta (*bet*) e em vez de abandonar, poderemos simplesmente passar (*check*). Vejamos as diferentes ações, uma por uma:

Apostar (*bet*): Poderemos apostar quando for a nossa vez e ainda ninguém apostou. Na modalidade “Sem Limite” poderemos apostar a quantidade que pretendermos, acima do valor mínimo permitido que falaremos mais à frente quando falarmos das rondas de apostas.

Subir (*raise*): Subiremos uma aposta quando algum jogador já tiver realizado uma aposta e nós pretendermos apostar uma quantidade maior. Num jogo “Sem limite” podemos subir a aposta tanto quanto queiramos até à totalidade das fichas que tenhas na mesa. Também existe um mínimo que devemos conhecer e que também falaremos mais à frente.

Igualar (*call*): Igualaremos uma aposta quando algum jogador tiver já apostado e nós queiramos apostar a mesma quantidade. Deveremos igualar o número de *Big Blinds* que o jogador tiver apostado.

Abandonar (*fold*): Consiste em desistir da mão quando alguém aposta. Podemos abandonar em qualquer ponto de uma mão se considerarmos oportuno.

Passar (*check*) Passar é uma acção que só poderemos tomar quando ainda nenhum

adversário tiver apostado e pretendemos saber o que irão fazer os restantes jogadores. Se, depois de passarmos, alguém apostar, voltará a chegar a nossa vez (pela ordem) para que voltemos a decidir o que fazer na mesma ronda de apostas.

1.5 Rondas de Apostas e desenvolvimento de uma mão

Agora vamos ver como se desenvolve uma mão de Poker e as fases em que se compõe.

Em primeiro lugar, logo que os jogadores estiverem distribuídos pela mesa, serão dadas as cartas. A entrega das cartas privadas começa na *Small Blind*, que recebe uma carta. Depois, o jogador na *Big Blind* recebe a sua primeira carta e assim sucessivamente para os restantes no sentido horário. Quando o jogador na última posição receber a primeira carta, o jogador na *Small Blind* começa a receber a sua segunda carta e repete-se o processo até que todos os jogadores tenham recebido as suas duas cartas privadas, e dá-se início à primeira ronda de apostas. No Poker existem quatro rondas de apostas.

Os seus nomes são de origem anglo-saxónica mas é importante aprender estes nomes pois não existe tradução válida em português para eles. As diferentes rondas de apostas chamam-se: *Preflop*, *Flop*, *Turn*, e *River*. Vamos conhece-las um pouco mais a fundo:

- ***Preflop***

O *Preflop* é a primeira ronda de apostas. É a ronda em que só contamos com as nossas cartas privadas. O seu nome vem de "Antes do Flop", dado que o Flop é a ronda seguinte de apostas.

Na ronda *Preflop*, antes de serem dadas as duas cartas a cada jogador, realizam-se obrigatoriamente duas apostas. Como já explicámos quando vimos as Posições. As apostas obrigatórias são a *Small Blind* e a *Big Blind*.

Depois de realizadas as apostas obrigatórias e dadas as duas cartas a cada jogador, começa a jogada. **O primeiro jogador a tomar a sua decisão Preflop será o jogador situado à esquerda da *Big Blind*.** Este jogador terá 3 opções:

- **Subir (*raise*)**: o jogador pode subir (aumentar) a aposta da *Big Blind* já que é a aposta mais elevada actualmente na mesa. Deve subir no mínimo o dobro da *Big Blind* e no máximo, todas as suas fichas.
- **Igualar (*call*)**: o jogador pode igualar a *Big Blind*.
- **Abandonar (*fold*)**: o jogador pode atirar as suas cartas e retirar-se da jogada que não lhe parece ser suficientemente boa. Ao abandonar, deixará de jogar até ao final da mão.

Depois de escolher e tomar a sua ação, será a vez do jogador sentado à sua esquerda, e assim sucessivamente. Todos os jogadores têm as mesmas opções, excepto

o jogador da *Big Blind* que têm a opção de passar (*check*) se ninguém tiver subido a sua aposta.

A primeira ronda de apostas (*Preflop*) termina quando todos os jogadores tiverem escolhido as suas acções e se tiverem igualado as apostas na mesa.

- ***Postflop***

- **Flop**

A partir deste ponto, o primeiro jogador a falar (tomar a sua decisão) já não será o primeiro jogador à esquerda da *Big Blind* mas o jogador situado na *Small Blind* ou, se este já não estiver em jogo, o jogador seguinte à esquerda deste, e assim sucessivamente.

Nesta ronda de apostas são mostradas 3 cartas no centro da mesa, que serão comunitárias para todos os jogadores. Ou seja, com as nossas duas cartas privadas e as 3 cartas comuns compomos a mão nesta rondas de apostas.

A partir desta ronda, a aposta mínima será sempre igual ao tamanho da *Big Blind*.

Como nesta ronda não existem as apostas obrigatórias que existiam *preflop*, todos os jogadores terão a opção (para além das que já tinham *preflop*) de passar (*check*) se ainda ninguém tiver apostado antes nesta ronda.

A ronda de aposta termina quando todos os jogadores tiverem falado e todas as apostas feitas tiverem sido igualadas. Se todos os jogadores decidirem passar, também acabará a ronda de apostas e passaremos à seguinte.

- **Turn**

Chegados ao *Turn*, mostra-se mais uma carta comunitária que também fará parte das nossas opções na formação da nossa mão. Agora dispomos de duas cartas privadas e quatro comuns, o que totalizam seis cartas disponíveis.

Inicia-se então o mesmo procedimento do flop, começando uma ronda nova de apostas com os jogadores que ainda se mantêm em jogo.

- **River**

A última ronda de apostas é o River. Nesta ronda é mostrada a quinta e última carta comunitária que poderemos utilizar para a nossa mão. Do mesmo modo que no Turn, no River a nossa mão será a mais forte de todas as possibilidades que podemos fazer com as sete cartas que dispomos.

Tal como no Flop e no Turn, começa uma ronda de apostas nova com o mesmo procedimento das anteriores. Uma vez terminada (e pelo menos dois jogadores tenham igualado os valores apostados) mostram-se as cartas de todos os jogadores que tiverem chegado até este ponto e verifica-se que tem a jogada mais alta.

Jogamos sempre todas as rondas de apostas?

Não, nem sempre. Se, em algum ponto da mão, apenas ficar um jogador em jogo, é considerado vencedor do pote sem haver necessidade de mostrar as suas cartas e a mão terminará. Isto significa que em muitas ocasiões haverá um jogador que efetuará uma aposta que ninguém quererá igualar e assim, não chegaremos às rondas seguintes dessa mão.

O que acontece se eu tiver apostado todas as minhas fichas?

Este movimento é conhecido como “*All-in*”. Se já tiveres colocado todas as tuas fichas em jogo numa ronda de apostas e enfrentares um só rival ou mais do que um que já não podem tomar mais decisões, serão mostradas as cartas comunitárias que ainda faltarem (se faltar alguma), e depois serão mostradas as cartas privadas para se verificar quem é que ganhou a mão.

Se houver mais do que um jogador com mais fichas do que outro que foi all-in, os restantes continuarão a jogar na disputa de um pote secundário no caso de ser realizarem mais apostas. Estas situações são geridas pelo próprio software da sala em que estiveres a jogar, pelo que não precisarás de te preocupar muito com isso.

1.6 Ranking de Mãos

Quando são mostradas as cartas, a tua jogada será a melhor possível de todas as combinações que se possam fazer das tuas duas cartas com as cinco cartas comunitárias. Em algumas situações usarás as tuas duas cartas, uma única ou simplesmente usarás as cinco cartas comunitárias da mesa. Ganha quem tiver a mão mais alta. Para saber qual é essa jogada, mostramos o ranking das mãos possíveis, ordenadas da maior ao menor valor:



Figura 1.1: Royal flush (Sequência máxima de cor)

A mão mais famosa do Poker, o royal flush, não pode ser derrotada. Consiste em ás, rei, dama, valete e dez de um único naipe.



Figura 1.2: Straight flush (Sequência de cor)

Cinco cartas em sequência do mesmo naipe. Na eventualidade de um empate, vence a carta com o valor mais alto no topo da sequência.



Figura 1.3: Four of a kind (poker)

Quatro cartas do mesmo valor, e uma carta de desempate ou ‘kicker’. Na eventualidade de um empate, vence o jogador com a carta lateral mais alta (‘kicker’).



Figura 1.4: Full house

Três cartas do mesmo valor (trio), e outras duas cartas do mesmo valor (par). Na eventualidade de um empate, vence o trio mais alto.



Figura 1.5: Flush

Cinco cartas de naipes diferentes, em sequência. Na eventualidade de um empate, vence a carta mais alta, do topo da sequência.



Figura 1.6: Straight (sequência)

Cinco cartas de naipes diferentes, em sequência. Na eventualidade de um empate, vence a carta mais alta, do topo da sequência.



Figura 1.7: Three of a Kind (Trio)

Três cartas do mesmo valor e duas cartas laterais não relacionadas. Na eventualidade de um empate, vence o jogador com a carta mais alta e se necessário, com a segunda carta lateral mais alta ('kicker').



Figura 1.8: Two pair (Dois pares)

Duas cartas de valor idêntico, outras duas cartas de outro valor idêntico entre si e uma carta de desempate. Se os dois jogadores tiverem Dois Pares de valor idêntico, vence a carta mais alta (kicker).



Figura 1.9: Pair (Par)

Duas cartas de valor idêntico e três cartas laterais não relacionadas. Na eventualidade de um empate, vence o jogador com a carta mais alta e se necessário, com a segunda ou terceira carta lateral mais alta.



Figura 1.10: High card (Carta alta)

Qualquer mão que não se qualifique numa categoria listada acima. Na eventualidade de um empate, vence a carta mais alta

Capítulo 2

Fases da Computação Especulativa

Neste capítulo iremos apresentar o que é a computação especulativa e as suas várias fases.

2.1 O que é a computação especulativa?

"Speculative computation [1] is an implementation technique that aims at speeding up the execution of programs, by computing pieces of code in advance, possibly in parallel with the rest of the program, without being sure that these computations are actually needed."

A computação especulativa para o nosso contexto significa prever um acontecimento a um determinado evento sendo que a informação sobre esse evento pode ser :

- Conhecida;
- Desconhecida;
- Parcialmente conhecida.

2.2 Formalização da *Framework*

Σ - Fontes de Informação

- *Dealer* que vai ser o responsável por dar as cartas aos jogadores, anunciar o vencedor/es, informar quantos jogadores continuam na jogada, anunciar o vencedor do torneio e dar as cartas para a mesa;

ε - Predicados Externos

- Cartas que estão na mesa;
- Escolha dos jogadores:
 - Se o jogador desistiu.
 - Se o jogador apostou mais do que era suposto (raise);
 - Se o jogador apostou normal;
 - Se o jogador fez *check*.
- Número de jogadores que estão a jogar a “mão”;
- Ronda.

Δ - Valores por defeito

- *Odd* mínima para o jogador ir a jogo=10;
- Na mesa têm de haver mais do que um jogador;
- Para começar o torneio têm de haver mais do que um jogador.

P- É um programa de lógica de restrição.

Apresentamos algumas regras aplicadas na nossa aplicação:

O jogador aposta normal se tiver uma carta maior que a carta número sete e se tiver na primeira ronda.

```
call(playerX,Valor)<- x>=CARD_7, Y=0 || cartasMao(playerX,X),ronda(Y);
```

O jogador aposta mais que o normal se tiver uma carta maior que um *Jack*, tiver um par na mão e estiver na primeira ronda.

```
raise(playerX,Valor)<-x>=JACK , Y=0, Z=ONE_PAIR || cartaMao(playerX,X),ronda(Y),rankingCartas
```

O jogador desiste da mão, se tiver na primeira ronda não tiver jogo nenhum, a carta que tem na mão ser menor que um *Jack* e o número de pessoas da mesa ser maior que um, sem contar com o jogador.

```
desisto(playerX) <- x=HIGH_CARD, Y=1 ,Z<JACK,L>1——rankingCartas(playerX,X),ronda(Y),car
```

;

O jogador aposta mais do que é suposto se o *ranking* for maior que um *STRAIGHT* tiver no mínimo penúltima ronda e tiver feito três *raises*.

`raise(playerX,Valor)<- X>=STRAIGHT , Y>=2,Z=3 || rankingCartas(playerX,X),ronda(Y), numeroRaiseFeitos(X,Y,Z).`

2.3 Definições preliminares

Os processos ativos são tuplos $\langle \langle -C || GS, UD \rangle$ onde:

- **GS**- É o conjunto de regras a ser provadas;
- **UD**- É o conjunto de regras que foram provadas através das regras por defeito.

Os processos ativos são tuplos $\langle SAS \langle -C || GS, UD \rangle$ onde:

- **SAS** são os predicados suspensos.

O **CBS** proporciona o atual estado de informação e especifica que tipo de informação estamos a lidar se é um facto ou valores por defeito.

AAQ são as perguntas já respondidas.

RF são as respostas retornadas da fonte de informação.

Existem dois passos do processo:

- Processos ativos **APS** -> Contém os processos que são consistentes com os valores por defeito da fonte de informação ;
- Processos suspensos **SPS** -> Contém os processos que são inconsistentes com os valores por defeito da fonte de informação.

2.4 Fases da Computação especulativa

A seguinte imagem ilustra as fases da computação especulativa.

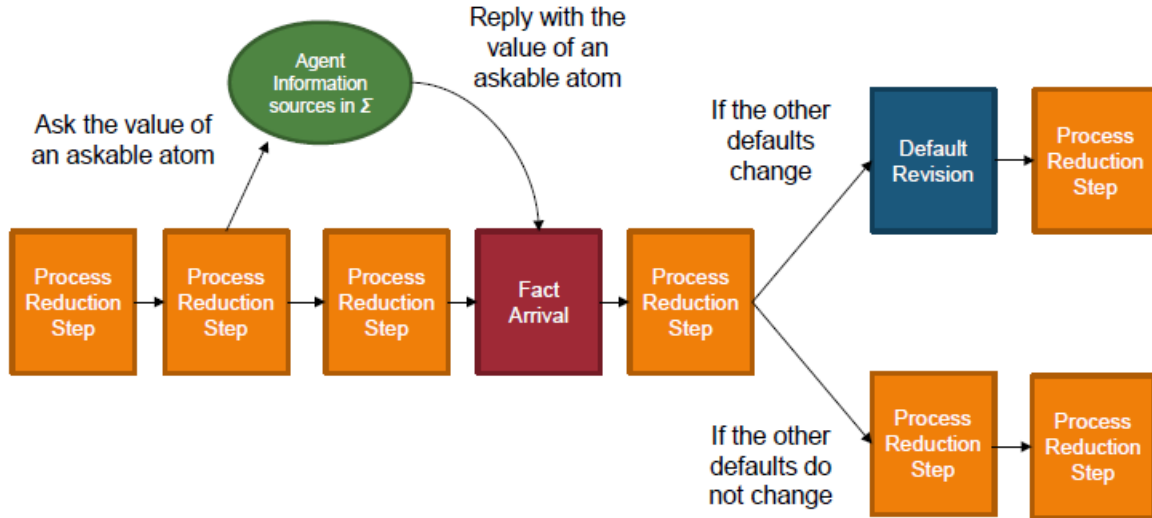


Figura 2.1: Fases da Computação Especulativa

2.4.1 Redução de processos

O *dealer* pergunta a todos os jogadores um a um qual irá ser a sua jogada.

O jogador antes de decidir qual é o seu movimento ele tem em conta os seguintes parâmetros:

- Cartas que tem não;
- As cartas que estão na mesa;
- O número de jogadores na mesa;
- E a ronda em que se encontra;
- O dinheiro que tem para jogar.

No decorrer das rondas pode acontecer que o jogador fica com um jogo pior do que tinha anteriormente, nesse caso o jogador faz *raise* na tentativa de "forçar" a saída de algum jogador.

A medida que os agentes vão respondendo ao *dealer* ele vai guardando essa informação e só avança para outra fase do jogo quando acaba de perguntar a todos os jogadores e obtiver resposta dos mesmos.

2.4.2 Obtenção de Factos

A medida que os jogadores vão respondendo ao *dealer* este avalia as respostas:

- Se o jogador desistiu da mão então o *dealer* atualiza a lista de jogadores que estão na mão;
- Se o jogador fizer *raise* é necessário percorrer os jogadores que tinham feito apostas até então e voltar a perguntar se querem ir a jogo;
- Se o jogador fizer *call* então o *dealer* acrescenta na lista o jogador e aumenta o *pot*;
- Se o jogador não tiver dinheiro é removido da mesa.

2.4.3 Revisão de valores por defeito

A *odd* mínima não sofre qualquer tipo de alteração.

Sempre que só haja um jogador a jogar a mão esse jogador é o vencedor da mão.

Se só houver um jogador no torneio então esse jogador é o vencedor do torneio.

Capítulo 3

Implementação

3.1 Agentes

A aplicação é constituída por dois agentes base que são:

- Software
- *Dealer*

estes agentes estão iniciados sempre que a aplicação é iniciada.

Para se iniciar a aplicação é necessário a criação de vários jogadores e atribui-lhes uma quantia monetária para entrarem na mesa, quantia essa que tem de ser maior que zero.

Cada jogador será um agente diferente que irá ter como nome **agentNick + n**, onde **n** é o número de agentes criados até ao momento, sendo que n começa em zero. Para haver uma separação dos agentes base e dos agentes jogadores criou-se outro *container*.

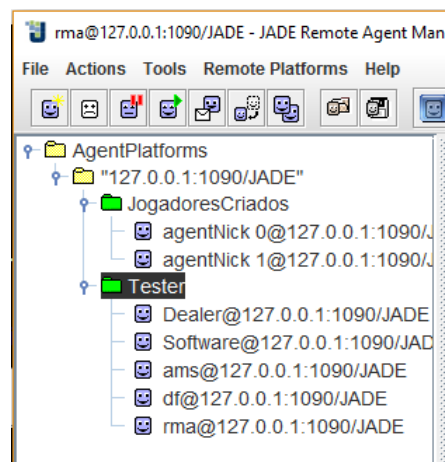


Figura 3.1: Estrutura dos agentes em Jade com a criação de dois jogadores.

3.2 Comunicação entre os agentes

Quando o utilizador acaba de escolher os jogadores que quer para o torneio, tem de escolher no mínimo dois jogadores, o utilizador inicia o torneio.

Ao iniciar o torneio o agente **Software** fica responsável por distribuir o dinheiro pelos respetivos jogadores e por avisar o agente **Dealer** para que este dê início ao torneio.

A partir daqui o agente **Software** acaba a sua execução passando agora existindo comunicação entre os jogadores e o agente **Dealer**.

O agente **Dealer** fica responsável por dar as cartas aos jogadores, informar os agentes quais são as cartas que estão na mesa, perguntar ao agente qual a ação que ele quer tomar fase ao estado atual do jogo.

Se o jogador responder que quer abandonar a mão então o agente **Dealer** remove o jogador da mão e só o volta a "chamar" quando a mão acabar. Se o jogador informar o agente **Dealer** que não tem dinheiro para continuar no torneio então o agente **Dealer** remove-o da mesa e nunca mais o volta a "chamar".

Se o jogador der *raise* então o agente **Dealer** informa os restantes jogadores que se quiserem ir para a ronda seguinte terão de apostar aquele valor. Os jogadores que já tenham apostado e quiserem ir para a ronda seguinte terão de apostar apenas a diferença entre o valor antigo apostado e o novo valor.

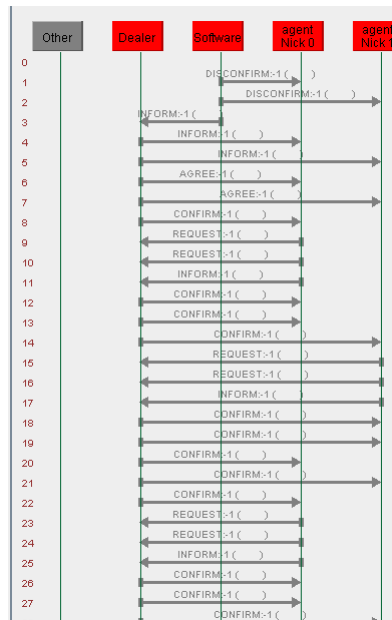


Figura 3.2: Comunicação entre os agentes.

3.3 Menu Inicial

Como é visível pela imagem abaixo o utilizador escolhe quantos jogadores quer para o torneio. Para iniciar o torneio o utilizador o que tem de fazer é carregar no botão **começar jogo** e desde que tenha dois ou mais jogadores o agente **Software** distribui o dinheiro pelos jogadores e aciona o agente ***Dealer*** para que este inicie o torneio.

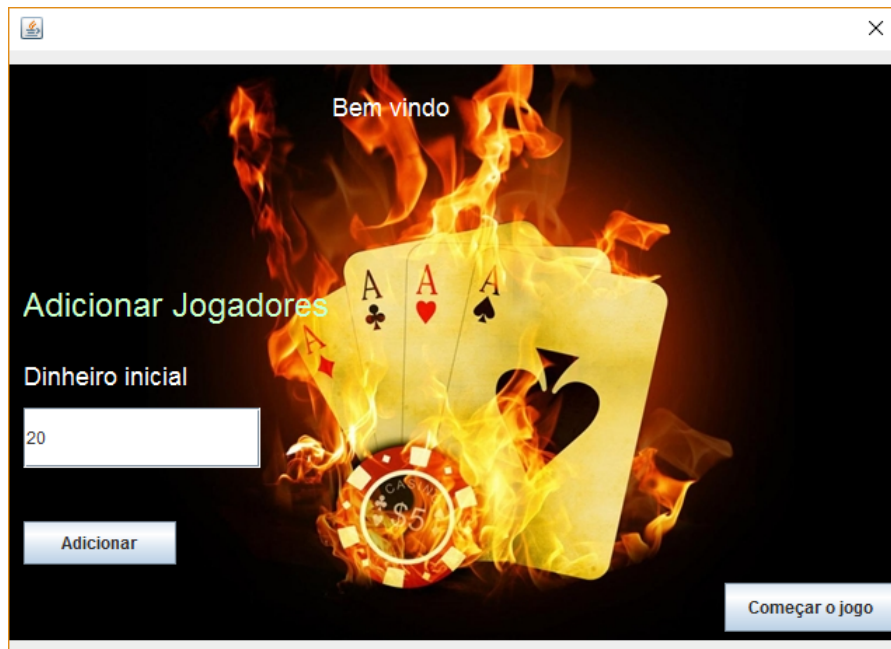


Figura 3.3: Tela inicial

3.4 Informação trocada entre os agentes

Na imagem é apresentada um excerto da troca de informação entre agentes a informação - >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>> EU vou para casa agent-Nick 0”acontece primeiro que o resto da informação porque está informação é empresa no lado do cliente e o agente só chega a receber essa informação segundos depois o que faz com que ele imprima a restante informação.

```
Saída - SI_Cn (run)
A perguntar agentNick 1
Joga 7.5 agentNick 1
Ganhei:: 87.5 agentNick 1
CLEAN Hand:: false
Mao nr 14
Cartas:: [Suit: SPADES, Rank :CARD_5, Suit: CLUBS, Rank :CARD_10] para agentNick 0
Cartas:: [Suit: SPADES, Rank :CARD_8, Suit: CLUBS, Rank :CARD_8] para agentNick 1
A perguntar agentNick 0
Joga 7.5 agentNick 0
A perguntar agentNick 1
nova mao: 2 Nome: agentNick 1
nova mao: 2 Nome: agentNick 0
->===== EU vou para casa agentNick 0
Joga 10.0 agentNick 1
FLOP:: [Suit: DIAMONDS, Rank :JACK, Suit: SPADES, Rank :QUEEN, Suit: SPADES, Rank :CARD_10]
A perguntar agentNick 0
Raise 5.0 agentNick 0
A perguntar agentNick 1
Joga 10.0 agentNick 0
A perguntar agentNick 0
Joga 5.0 agentNick 0
TURN:: Suit: CLUBS, Rank :QUEEN
A perguntar agentNick 1
Joga 5.0 agentNick 1
A perguntar agentNick 0
Joga 5.0 agentNick 1
RIVER:: Suit: HEARTS, Rank :CARD_7
A perguntar agentNick 1
Joga 5.0 agentNick 0
A perguntar agentNick 0
Raise 3.75 agentNick 1
A perguntar agentNick 1
Raise 3.75 agentNick 0
A perguntar agentNick 0
Joga 3.75 agentNick 1
Ganhei:: 63.75 agentNick 0
CLEAN Hand:: false
GRANDE VENCEDOR:: agentNick 1
```

Figura 3.4: Mensagens obtidas execução

Capítulo 4

Conclusão

Perante o trabalho desenvolvido é possível assumir que produzir programas que tratam informação desconhecida ou incompleta é uma mais-valia para o nosso quotidiano. Atualmente existem inúmeros sistemas capazes de prever comportamento dos utilizadores baseando-se ou em comportamentos antigos, ou então no comportamento dos outros utilizadores. Um exemplo claro deste sistema é a *Amazon* que tenta através de experiências com outros utilizadores prever o comportamento de outros.

A computação especulativa pode ser usada para fins comerciais permitindo as empresas ganharem muito dinheiro, mas também pode ser usada na área da saúde permitindo que os médicos através de outros casos com pacientes consigam chegar a um possível diagnóstico.

Na realização deste trabalho tentou-se que o agente tentasse perceber o jogo do adversário através das movimentações que ele fazia.

Bibliografia

- [1] Boudol, Gérard and Petri, Gustavo. *A Theory of Speculative Computation* INRIA, 06902 Sophia Antipolis, France.
- [2] Computação especulativa em inteligência ambiente
Fernandes, João Pedro Araújo
<http://repositorium.sdum.uminho.pt/handle/1822/28544>
- [3] <http://www.pokerpt.com/conceitos-regras-basicas-poker.html>
- [4] Oliveira, Tiago; Neves, José e Novais, Paulo. *SPECULATIVE COMPUTATION*. Intelligent Systems Labs, Universidade do Minho.