

Monopólio com agentes JADE

Guilherme Sousa,
Hugo Fernandes,
Ricardo Pinto,

up201909575
up201909576
up201909580

Índice

- Descrição do problema;
- Esquema global do projeto;
- Comunicação entre agentes;
- Arquiteturas dos agentes e estratégias utilizadas;
- Outros mecanismos;
- Experiências realizadas e análise dos resultados;
- Conclusões;
- Exemplos detalhados de execução;
- Classes implementadas;
- Outras observações.

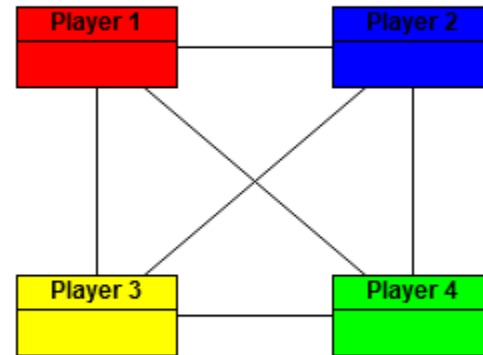
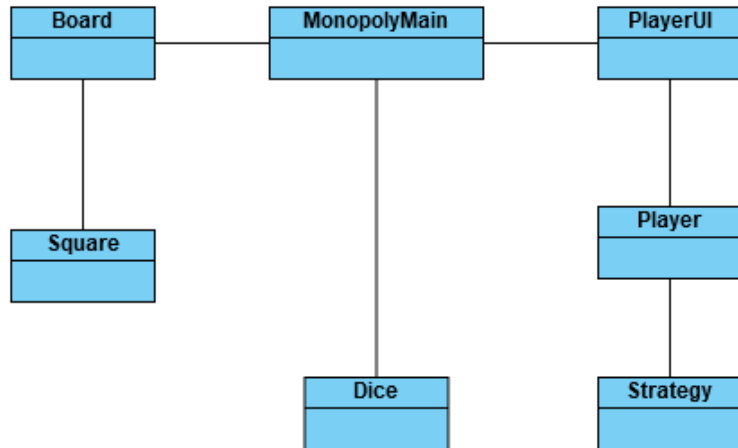
Descrição do Problema

- O jogo de monopólio jogado por agentes JADE;

As regras do jogo implementado são iguais às regras do jogo de tabuleiro com algumas exceções:

- Não há trocas de propriedades entre jogadores;
- Não há hotéis;
- Não existe hipoteca;
- Não há companhia de águas nem de eletricidade;
- Na casa sorte o jogador anda duas casas para a frente ou duas casa para trás;
- Na casa da caixa da comunidade o jogador recebe ou perde um montante entre 10-200;
- Não existem estações;
- Não é necessário ter todas as localidades da mesma cor para a construção de casas;

Esquema global do projeto



Comunicação entre agentes

Em relação a protocolos, o escolhido para o nosso projeto foi o *FIPA-Subscribe*, nomeadamente através da utilização da classe *DFSubscriptionInit* que implementa o behavior descrito no referido protocolo designado por *SubscriptionInitiator*.

Quanto à troca de mensagens entre agentes, são utilizadas mensagens ACL, que são enviadas por cada *Player* individualmente e recebidas e tratadas na classe *PlayerListeningBehavior*, que como o nome refere, corresponde a um *behavior* definido para todos os agentes da nossa aplicação.

Estratégias utilizadas

Cada um dos agentes tem uma estratégia atribuída:

1. Comprar todas as propriedades que pode sempre que a compra não fizer o *player* ficar sem dinheiro;
2. Comprar todas as propriedades possíveis a partir de um ronda diferente dos outros *players* (3ª);
3. Comprar propriedades apenas de algumas cores (3 cores diferentes), determinadas aleatoriamente;
4. Comprar ou não comprar propriedades baseado na eficácia atribuída a cada propriedades do mapa no algoritmo.

Qualquer *player* pode adicionar casas às suas propriedades. O preço da renda é aumentado baseado no preço inicial e no quadrante do tabuleiro.

Outros mecanismos

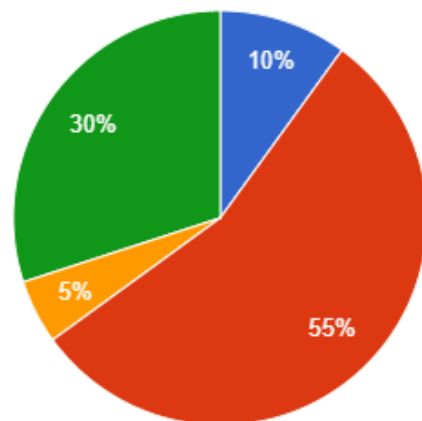
Em termos de utilização de outros mecanismos, o mecanismo utilizado por nós no desenvolvimento da solução foi o serviço das páginas amarelas. Este serviço é fornecido, pelo JADE que implementa um agente *Directory Facilitator*(DF), especificado pelo FIPA. Com este serviço conseguimos registar os agentes quando estes são criados e permitem também a cada agente individual pesquisar por outros agentes registados no DF.

Na nossa implementação os *Players* registam-se aquando da sua criação e cada *Player* realiza apenas uma pesquisa ao DF, aquando da sua primeira jogada. Os resultados desta pesquisa são armazenados numa lista que depois é alterada aquando da desistência de algum jogador.

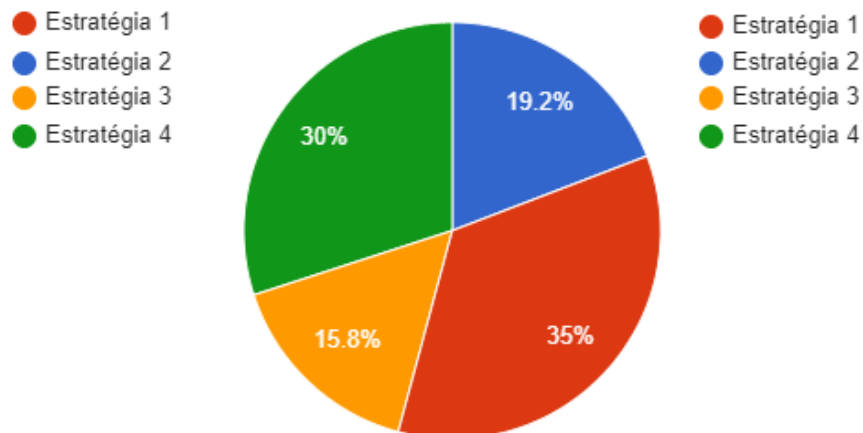
Experiências realizadas e análise dos resultados

	vencedor	2ºClass.	3ºClass.	4ºClass.
1	est4	est2	est3	est1
2	est1	est4	est2	est3
3	est3	est2	est4	est1
4	est4	est1	est3	est2
5	est4	est1	est3	est2
6	est2	est1	est3	est4
7	est4	est3	est2	est1
8	est4	est2	est1	est3
9	est2	est4	est1	est3
10	est1	est4	est2	est3
11	est1	est4	est3	est2
12	est1	est2	est4	est3
13	est1	est3	est2	est4
14	est1	est4	est3	est2
15	est4	est2	est1	est3
16	est1	est4	est3	est2
17	est1	est4	est2	est3
18	est1	est4	est3	est2
19	est1	est3	est2	est4
20	est1	est3	est2	est4

Vitória



Atribuição de Pontos



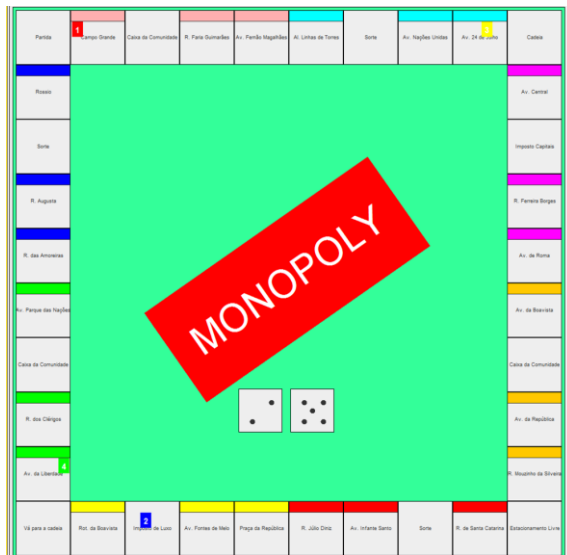
Conclusões

Com os resultados obtidos verificamos que independentemente da atribuição de pontos, o pódio permanece sempre igual. a melhor das 4 estratégias é a primeira (comprar tudo o que pode), seguida pela quarta (comprar considerando a eficiência), em terceiro lugar fica a segunda estratégia (comprar tudo a partir de uma certa ronda) e por fim a terceira estratégia (compra apenas propriedades de cores específicas).

Como trabalho futuro, podemos listar:

- Incrementar as regras implementadas;
- Desenvolver mais estratégias;
- Melhorar a UI.

Exemplos detalhados da execução



Player 1 Information

Player 1 Now Playing
Current Balance: 1336
Title Deeds:
- Campo Grande*
- R. de Santa Catarina**
- Rossio
- Av. Nações Unidas*
- Praça da República
- Av. Parque das Nações

Player 1 is at Av. da República

Player 1 Information

Player 1 Now Playing
Current Balance: 994
Title Deeds:
- R. Ferreira Borges*
- R. Mouzinho da Silveira
- Av. Fontes de Melo
- R. dos Clérigos
- R. Augusta
- Al. Linhas de Torres**
- Av. da Boavista**
- Av. Nações Unidas
- Campo Grande
- Av. Parque das Nações
- R. das Amoreiras
- Rossio

YOU WON PLAYER 1

Classes implementadas

O projeto divide-se, principalmente, em sete classes:

- *MonopolyMain;*
- *Board;*
- *Square;*
- *Dice;*
- *Player;*
- *Strategy;*
- *PlayerUI.*
- *PlayListeningBehaviour*