UNIVERSIDADE DE SÃO PAULO
Escola Politécnica da USP

PCS5703 – Exercício Prático 3
Planejamento

Profa. Dra. Anarosa Alves Franco Brandão

23/11/2020

O objetivo deste exercício é familiarizar o aluno com os conceitos de planejamento em Sistemas Multiagentes. O exercício deverá ser feito **em grupo**, com entrega até o dia 14/12/2020 através do GitHub.

1 Requisitos:

Os alunos deverão ter instalado, em seu computador:

- Java SDK versão 1.8 ou superior
- Um ambiente de desenvolvimento JAVA (IDE) de sua preferência
- Ferramentas necessárias para submeter o código via GitHub, tais como: Git client, GitHub client, GitKraken, ou outras similares

2 Contexto:

Para este exercício prático, os alunos irão utilizar o arcabouço de sistemas multiagentes **Hecate** e os exercícios práticos anteriores. Diferentemente dos outros arcabouços vistos em aula, o Hecate não requer nenhuma dependência local, uma vez que todos os agentes e artefatos são criados remotamente.

Durante a realização deste exercício, os alunos irão utilizar uma versão modificada do Hecate para fins didáticos. Um arquivo JAR será disponibilizado via GitHub, e os alunos deverão implementar um sistema local que reflita o cenário proposto (abaixo).

3 Cenário:

O cenário de implementação do exercício será baseado no exemplo "house building" do JaCaMo, conforme visto em sala de aula.

Neste cenário, considera-se que Giacomo (representado, no sistema, pelo leiloeiro) quer construir uma casa. Para tanto, ele deve contratar diversas empresas (representadas por agentes) especializadas em diferentes etapas do processo de construção. Serão utilizadas as mesmas etapas do exemplo "house building":

- Site preparation
- Lay floors
- Build walls
- Build roof
- Fit windows
- Fit doors
- Install the plumbing
- Install the electrical system
- Paint the exterior of the house
- Paint the interior of the house

Cada uma destas etapas possui uma representação paramétrica única, que pode ser recuperada através da chamada apropriada ao Hecate (ver lista de chamadas mais abaixo).

Cada etapa será leiloada separadamente. Em cada leilão, Giacomo irá, **de forma aleatória**, escolher uma das etapas de construção e definir qual o máximo valor que ele está disposto a pagar pelo serviço relacionado. Cada empresa, então, deverá ofertar um lance. A empresa que ofertar o menor lance será eleita para realizar o serviço, e o próximo leilão irá começar automaticamente.

Cada leilão segue as seguintes regras:

- Os lances só serão aceitos se forem feitos por empresas capazes de executar a tarefa sendo leiloada.
- A qualquer momento do leilão, pode-se recuperar qual é o valor do lance que está ganhando, mas não é possível identificar a empresa responsável pelo leilão.
- Os valores dos lances deverão ser múltiplos de 5 (cinco), e superiores a 0 (zero).
- Assim que todas as empresas capacitadas a executar a tarefa leiloada fizerem um lance, o leilão será automaticamente finalizado e o próximo leilão começará automaticamente, até que não restem mais etapas a se leiloar.

- Será possível consultar todos os leilões passados, bem como a empresa ganhadora e o valor do lance aceito em cada caso.
- Será possível consultar os leilões remanescentes (a começar).
- Cada agente (empresa) possuirá uma lista com todos os lances ganhadores (separados por leilão).
- O leilão não será finalizado enquanto todos os agentes capacitados a executar a tarefa leiloada não oferecerem um lance **válido** (igual ou inferior ao máximo valor do leilão, e superior a zero).

Ao final de todos os leilões, todas as tarefas leiloadas deverão ter sido ganhas por uma das empresas.

A fim de atingir os objetivos deste exercício, os grupos deverão implementar um sistema local que decida como os lances serão feitos por cada empresa, de acordo com suas especializações (abaixo). Os requisitos para este sistema são:

- O sistema será responsável por inicializar o Hecate em três etapas: criação do mundo (início), criação das empresas (agentes), e criação do processo de leilões.
- Cada agente deverá implementar as especialidades de uma empresa específica (definidas abaixo).
- Cada agente deverá ter suas especialidades registradas no sistema **ANTES** do início do processo de leilões.
- Uma vez que o processo de leilões for iniciado, o sistema deve ser capaz de recuperar, a qualquer momento, qual leilão está em andamento, e qual é o atual valor ganhador do lance.
- Deve-se implementar um sistema de controle de lances **local** que permita, a qualquer momento, saber quais empresas fizeram quais lances para quais leilões. Esse histórico deverá ser **cronológico** (incluindo *timestamp*) e fará parte dos entregáveis.
- O sistema será responsável por garantir que cada lance dado, em cada leilão, é válido (de acordo com as regras acima e com as restrições de cada empresa).
- O sistema é responsável por determinar a estratégia de lances, ou seja, qual lance será dado por cada empresa (agente) a cada momento. As estratégias deverão implementar as restrições de cada empresa (listadas abaixo).
- O sistema deverá ser capaz de identificar o término de um leilão, de forma a se reiniciar o processo de lances.
- O sistema deverá ser capaz de recuperar uma lista de leilões passados que será utilizada no controle geral de monitoramento das empresas, de forma a se garantir que todas as tarefas sejam leiloadas com sucesso.

Tendo em mente os requisitos acima listados, pede-se:

- O aluno deve criar, no ambiente remoto, **cinco** agentes distintos, representando empresas de acordo com as especialidades listadas abaixo.
- Uma vez criado o ambiente inicial, o aluno deverá registrar as especializações de cada empresa usando as chamadas apropriadas do Hecate.
- Uma vez que os agentes estiverem com suas especialidades definidas, o processo de leilão deverá ser iniciado.
- Todos os lances deverão ocorrer de acordo com as estratégias implementadas pelos alunos de forma a se garantir que todas as etapas sejam ganhas por pelo menos uma empresa, e que todas as empresas respeitem suas limitações (listadas abaixo).
- Após a finalização de todos os leilões, os alunos deverão salvar um histórico **cronológico** de todos os lances feitos por todos os agentes (incluindo *timestamp*), e uma lista contendo todos os ganhadores de todos os leilões (com os valores de lance correspondentes). As listas produzidas deverão ser salvos em arquivos de texto (nos formatos "lances.txt" e "leiloes.txt").

4 Entregáveis:

- Código-fonte completo do ambiente local, incluindo instruções de execução
- Arquivos texto produzidos (conforme descrição acima).
- Descrição da estratégia utilizada pelos agentes durante o processo de leilão, incluindo detalhes sobre a implementação.

5 Documentação técnica:

Para este exercício, o arcabouço Hecate utiliza três estruturas principais: **agentes**, **leilões** e **lances**. Além disso, utiliza-se uma estrutura auxiliar para se enviar ao Hecate as especializações de cada agente. Cada uma dessas estruturas pode ser vista como um objeto (classe) com as seguintes propriedades (os parâmetros JSON serão sempre no formato **string**, salvo observação específica):

Agente:

- ID: Um número de identificação único no sistema
- Name: Um identificador descritivo (texto)
- Auction Types: Uma lista de especializações, sendo cada uma representada por uma palavra de texto (string)
- Winning Bids: Uma lista (coleção) de lances ganhadores (objetos)

Lance:

- Agent ID: Agente responsável pelo lance

- Bid Value: Valor do lance (número inteiro)

Leilão:

- AuctionType: Tipo de leilão, dentro dos tipos de leilão possíveis (recuperáveis a partir do Hecate)
- Maximum value: Valor máximo permitido para o leilão (número inteiro)
- Best bid: Valor do melhor lance do leilão até o momento (número inteiro)
- Auction Winner: ID do agente ganhador do leilão, é preenchido quando o leilão for finalizado

Especializações: Uma lista de especializações é uma lista de strings (texto) contendo cada especialização possível (lista recuperada diretamente do Hecate). Ao ser utilizada como parâmetro nas chamadas, esta lista possui o nome **auctionTypes**.

A criação de um novo agente no sistema deverá ser feita através de seu nome. O ID do agente criado será fornecido pelo sistema após sua criação.

A lista de empresas a se criar no Hecate encontra-se abaixo, bem como suas especializações e restrições:

- Empresa 1: Encanamento (plumbing), pintura interior e portas (doors)
- Empresa 2: Preparação do terreno (site preparation) e instalação de janelas (windows). Não pode ofertar valores inferiores a 10.
- Empresa 3: Preparação do terreno (site preparation), chão (floors), paredes (walls), telhado (roof), portas (doors), encanamento (plumbing), sistema elétrico (electrical system) e pintura exterior. Não pode ofertar valores inferiores a:
  + Terreno: 15
  + Chão: 5
  + Paredes: 10
  + Telhado: 15
  + Portas: 10
  + Encanamento: 20
  + Sistema elétrico: 5
  + Pintura exterior:10
- Empresa 4: Preparação do terreno (site preparation), chão (floors), paredes (walls), telhado (roof), portas (doors), encanamento (plumbing), sistema elétrico (electrical system), pintura exterior e pintura interior. Não pode realizar mais que quatro tarefas.
- Empresa 5: Pode realizar todas as tarefas, desde que o ganho total (soma de todos os lances ganhadores) não seja superior a 80.

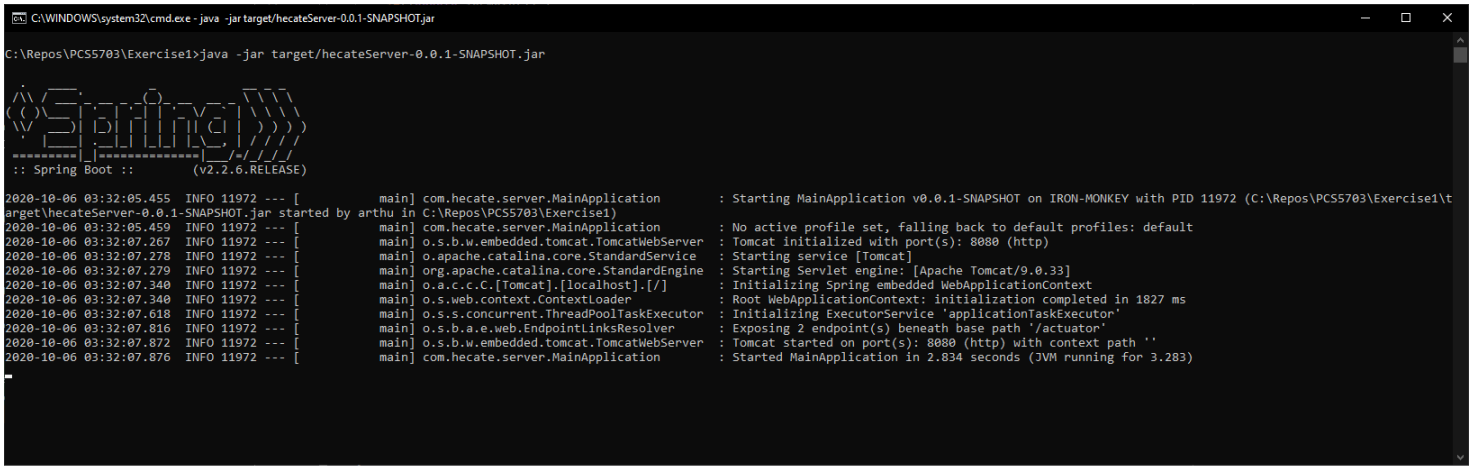

Figure 1: Hecate: Inicialização

Todo o acesso ao Hecate é feito através de chamadas REST. O resultado de todas as chamadas é processado no console do servidor, de forma que o aluno pode acompanhar o que está acontecendo após cada chamada de forma visual.

Abaixo encontra-se uma figura do console no momento em que o ambiente é iniciado:

Ao ser iniciado, o aluno deverá inicialmente criar o mundo no sistema através da chamada apropriada (abaixo). Durante esse processo, o sistema cria três salas já populadas por três diferentes agentes. Estas salas (e seus agentes) deverão ser desconsideradas pelo aluno durante a realização do exercício prático (ou seja, nem as salas inicialmente criadas, nem os agentes nelas contidos fazem parte dos requisitos de implementação do sistema).

Abaixo encontra-se uma figura do console no momento em que o mundo é criado:

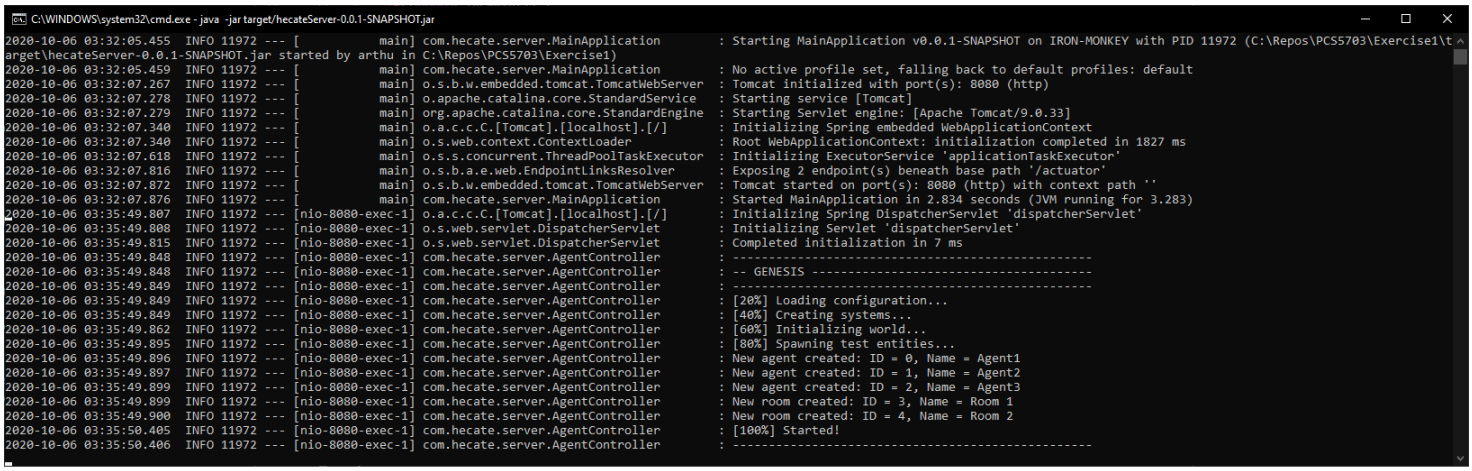

Figure 2: Hecate: Criação do mundo

Uma vez criado o mundo, o aluno deverá utilizar as chamadas REST associadas a cada funcionalidade do sistema. Cada chamada é uma URL composta pelo endereço do servidor (que nesse caso

roda localmente na porta 8080) e o caminho para a funcionalidade específica. Exemplificando: a criação de mundo é disponibilizada no caminho "/createWorld". Assim, considerando-se o endereço local do servidor, o endereço completo para esta chamada é: "http://localhost:8080/createWorld". **IMPORTANTE: os caminhos são case-sensitive.**

Abaixo encontram-se descritas as chamadas disponíveis nesta versão do Hecate. **Todas as chamadas retornam um objeto JSON (salvo observação específica)**. Algumas dessas chamadas requerem parâmetros específicos; todos os parâmetros listados abaixo podem ser enviados na forma de texto:

- **/createWorld** : Cria o mundo dentro do Hecate. Esta chamada deve ser sempre executada após a inicialização do sistema. Retorna mensagem simples de confirmação (não é JSON). Nesta chamada também são criadas entidades para teste, pré-populadas (ver log do console).
- **/createAgent** : Cria um novo agente no mundo. Retorno: representação em JSON do agente criado. Parâmetros necessários: **name** contendo o nome do agente.
- **/startAuctions** : Inicia o processo de leilões.
- **/agent/id** : Recupera uma representação do agente em JSON.
- **/availableAuctionTypes**: Recupera todos os leilões que ainda não foram realizados(JSON).
- **/maxAuctionValues** : Recupera uma lista de todos os leilões possíveis, com seus respectivos valores máximos.
- **/currentAuction** : Recupera o leilão em andamento.
- **/finishedAuctions** : Recupera a lista de todos os leilões já finalizados.
- **/auctionParticipants** : Recupera os participantes que já fizeram lances no leilão em andamento.
- **/agent/setAuctionTypes** : Adiciona uma lista de especializações a um determinado agente. Retorno: representação em JSON do agente modificado. Parâmetros necessários: **id** contendo o ID do agente; **auctionTypes** contendo uma lista (arrays) de especialidades a serem adicionadas ao agente.
- **/placeBid** : Efetua um lance no leilão em andamento. Retorno: representação em JSON do leilão em andamento (ou do novo leilão, caso tenha sido o último lance possível). Parâmetros necessários: objeto em JSON contendo os parâmetros **agentID** e **bidValue**, conforme descritos acima.