

Aula 13 – LIDA: Controlando o WorldServer3D

Objetivo

Utilizar a arquitetura cognitiva LIDA para controlar uma criatura artificial no ambiente virtual WordServer3D (WS3D) através de implementação de um aplicativo utilizando linguagem de programação Java e modificação dos arquivos de configuração XML.

Atividade 1

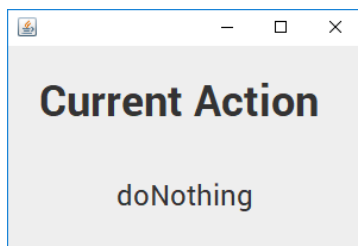
Após realizar o download do código da aplicação base DemoLIDA, foi executada a sincronização com os códigos da biblioteca WS3DProxy e a aplicação WS3D. Ao estudar o código da aplicação, foi percebido o uso do código Java e a integração com os arquivos de configuração XML.

Atividade 2

Nesta segunda atividade, foi proposta a modificação da estrutura dos painéis do Lida Framework para adicionar a visualização das ações a serem tomadas pela criatura a cada ciclo de execução.

Dois painéis foram implementados para disponibilizar esta informação:

- a) Painel externo com a visualização da ação corrente (JFrame)



Para implementar este painel, um novo JFrame foi implementado através da classe CurrentActionFrame para visualização da ação corrente. A atualização do painel é realizada no método Environment::processAction().

- b) Painel interno no Lida Framework

Para tanto o código do arquivo guipanel.properties foi modificado, adicionando um novo painel na posição A (linhas 17 e 18) para efetuar a visualização conforme abaixo:

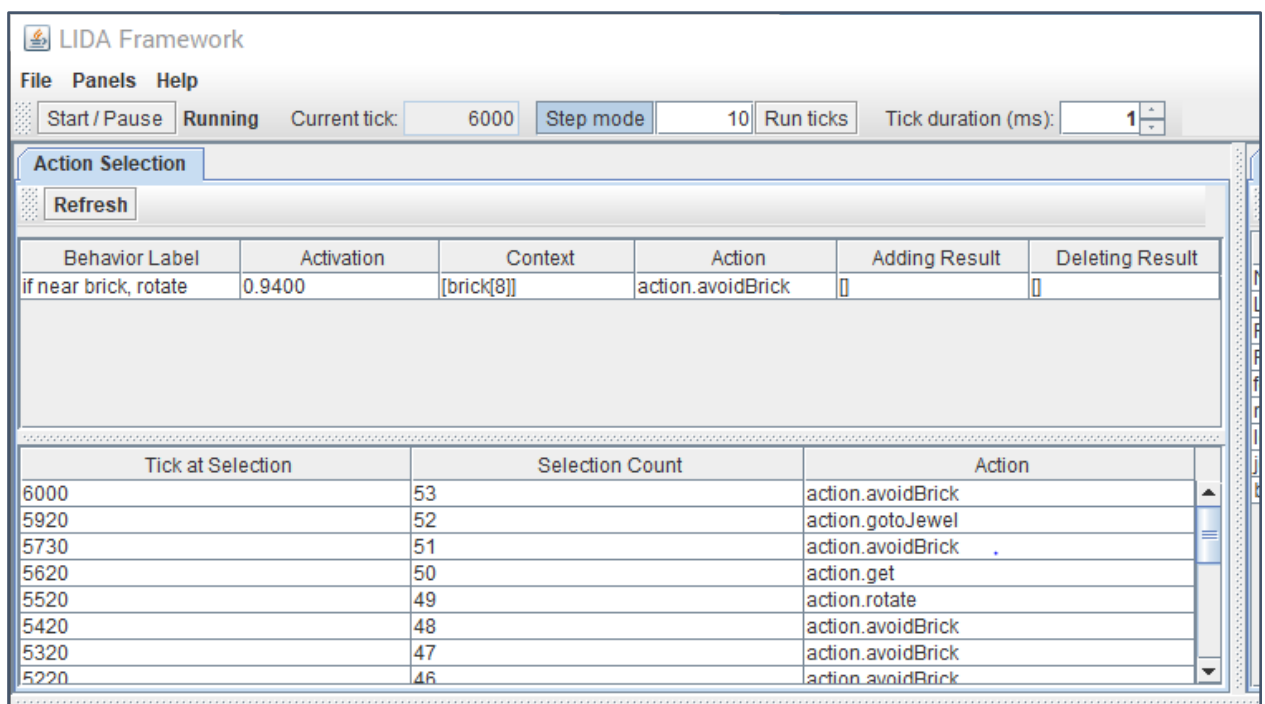
```
1. #####
2. # Copyright (c) 2009, 2011 The University of Memphis. All rights reserved.
3. # This program and the accompanying materials are made available
4. # under the terms of the LIDA Software Framework Non-Commercial License v1.0
5. # which accompanies this distribution, and is available at
6. # http://ccrg.cs.memphis.edu/assets/papers/2010/LIDA-framework-non-commercial-
7. # v1.0.pdf
8. #####
9. #name = panel title, class name, Position [A,B,C,FLOAT, TOOL], tab order, Refresh af
   ter load
10.
11. # ____Tool Bar____
12. ToolbarToolbar=Toolbar,edu.memphis.ccrq.lida.framework.gui.panels.ControlToolBarPane
   1,TOOL,1,Y,1,20
```

```

13.
14. # ----A Section----
15. #environ =AlifeEnvironment, alifeagent.guipanels.ALifeGuiPanel, A, 1, Y, configs/icons.pr
    operties, 40
16.
17. # Action Selection Panel - Use A position.
18. actionPanel=Action Selection, edu.memphis.ccrq.lida.framework.gui.panels.ActionSelect
    ionPanel, A, 0, Y, 20
19.
20. # ----B Section----
21. pamTable = PAM Table, edu.memphis.ccrq.lida.framework.gui.panels.NodeStructureTable, B
    , 0, Y, PerceptualAssociativeMemory
22. pamGraph = PAM Graph, edu.memphis.ccrq.lida.framework.gui.panels.NodeStructurePanel, B
    , 1, Y, PerceptualAssociativeMemory
23. activationChart=Activation Chart, edu.memphis.ccrq.lida.framework.gui.panels.Activati
    onChartPanel, B, 2, Y, 100, leafletJewel
24. perceptualBufferGraph = Perceptual Buffer, edu.memphis.ccrq.lida.framework.gui.panels
    .NodeStructurePanel, B, 3, Y, Workspace.PerceptualBuffer
25. globalWorkspace = Global Workspace, edu.memphis.ccrq.lida.framework.gui.panels.Global
    WorkspaceTablePanel, B, 4, Y, 10
26. proceduralMemory=Procedural Memory, edu.memphis.ccrq.lida.framework.gui.panels.Proced
    uralMemoryPanel, B, 5, Y
27. #actionSelection=Action Selection, edu.memphis.ccrq.lida.framework.gui.panels.ActionS
    electionPanel, B, 6, Y, 5
28.
29. # ----C Section----
30. LogPanel=Logging, edu.memphis.ccrq.lida.framework.gui.panels.LoggingPanel, C, 1, N, alif
    eagent, edu.memphis.ccrq.alife
31. runningTasks = Running Tasks, edu.memphis.ccrq.lida.framework.gui.panels.FrameworkTas
    kPanel, C, 2, Y, PerceptualAssociativeMemory
32. taskQueue=Task Queue, edu.memphis.ccrq.lida.framework.gui.panels.TaskQueuePanel, C, 3, N
33. configFiles = Configuration Files, edu.memphis.ccrq.lida.framework.gui.panels.Configu
    rationFilesPanel, C, 4, N

```

O resultado na janela do Lida Framework pode ser visualizado na imagem:



Atividade 3

Nesta terceira atividade, foi proposta a alteração do código do DemoLIDA, criando um mecanismo por meio do qual a criatura pudesse detectar blocos a partir do ambiente e se movimentasse de uma origem até um destino sem colidir com os blocos.

Para execução das alterações são providos scripts nos sistemas operacionais Linux e Windows:

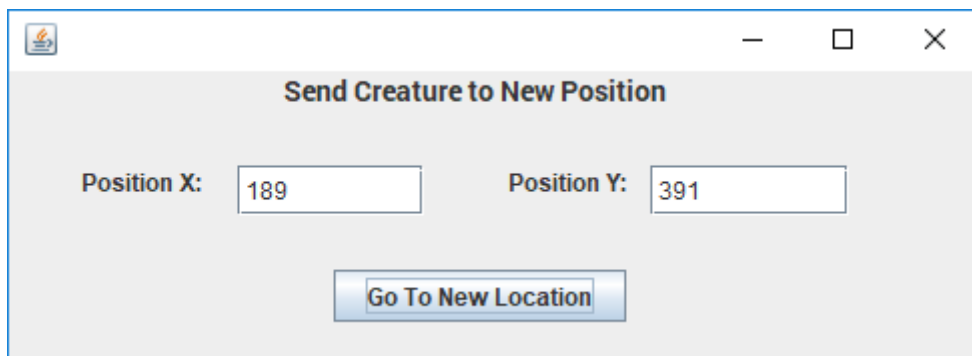
run-R4.sh / run-R4.bat

Descrição da implementação

Esta seção do relatório visa descrever as principais alterações necessárias ao código fonte do DemoLIDA.

1. Alterações no WS3D e WS3dProxy

Os dois componentes foram alterados para permitir que ao o usuário clicar com o botão do meio do mouse, uma janela de confirmação fosse visualizada para confirmar o novo destino da criatura. Ao se aceitar a nova posição, a criatura é roteada para o novo destino.



2. Alterações no código Java do DemoLIDA

- Remoção do código de detecção de jóias, comidas e jóias de leaflet para tornar o código mais compreensível, uma vez que para o objetivo da simulação não seriam mais necessários.
- O código do arquivo Environment.java foi modificado para inserir de forma aleatória alguns blocos (*bricks*) no ambiente no método Environment::init().
- Uma nova classe no arquivo PathFinder.java foi criada para utilizar a biblioteca GridNav-java (<https://github.com/elnygren/GridNav-java>) para executar o roteamento da criatura no ambiente até o novo destino. Essa biblioteca fornece *pathfinding* usando alguns algoritmos distintos. Para o DemoLIDA foi selecionado o A-Star.
- Alterações diversas no arquivo Environment.java para executar a detecção de blocos no ambiente de simulação e execução de ações para alterar a trajetória da criatura, nos métodos getState(), updateEnvironment() e performAction().
- Finalmente, foi adicionada a implementação de um detector adicional para os blocos do ambiente, no arquivo BrickDetector.java.

3. Alterações nos arquivos de configuração XML

Para implementação da detecção dos blocos no ambiente, foi adicionada ao arquivo **factory.xml** uma task inicial extra para inicializar o detector de blocos:

```
1. <task name="BrickDetector">
2.   <class>detectors.BrickDetector</class>
3.   <ticksperrun>3</ticksperrun>
4.   <associatedmodule>SensoryMemory</associatedmodule>
5.   <associatedmodule>PerceptualAssociativeMemory</associatedmodule>
6. </task>
```

No arquivo de definição do agente, Agent.xml, as seguintes alterações foram realizadas:

- Adição de uma nova task no módulo de memória perceptual associativa (*PerceptualAssociativeMemory*) e novo node para detecção de bloco (*brick*) no ambiente:

```
1. <param name="nodes">
2.   food,nearObject,leafletJewel,jewel,brick
3. </param>

1. <task name="BrickDetector">
2.   <tasktype>BrickDetector</tasktype>
3.   <ticksperrun>3</ticksperrun>
4.   <param name="node" type="string">brick</param>
5.   <param name="learnable.baseLevelActivation" type="double">0.0</param>
6.   <param name="learnable.baseLevelRemovalThreshold" type="double">-1.0</param>
7.   <param name="learnable.baseLevelDecayStrategy" type="string">defaultDecay</param>
8.   <param name="learnable.baseLevelExciteStrategy" type="string">defaultExcite</param>
9.   <param name="learnable.totalActivationStrategy" type="string">DefaultTotalActivationStrategy
10. </param>
10. </task>
```

- No módulo de atenção (*Attention Module*), um novo codelet foi adicionado para trazer para a consciência o evento de detecção de um bloco:

```
1. <task name="BrickAttentionCodelet">
2.   <tasktype>BasicAttentionCodelet</tasktype>
3.   <ticksperrun>5</ticksperrun>
4.   <param name="nodes" type="string">brick</param>
5.   <param name="refractoryPeriod" type="int">50</param>
6.   <param name="initialActivation" type="double">1.0</param>
7.   <param name="learnable.baseLevelActivation" type="double">1.0</param>
8.   <param name="learnable.baseLevelRemovalThreshold" type="double">-1.0</param>
9.   <param name="learnable.baseLevelDecayStrategy" type="string">defaultDecay</param>
10.  <param name="learnable.baseLevelExciteStrategy" type="string">defaultExcite</param>
11.  <param name="learnable.totalActivationStrategy" type="string">DefaultTotalActivationStrategy
12. </param>
12. </task>
```

- No módulo de memória procedural, foi adicionada a ação a ser tomada quando da detecção de um bloco no ambiente. Procurou-se ajustar o nível base de detecção de forma a privilegiar a ação de se desviar de blocos:

```
1. <module name="ProceduralMemory">
2.   <class>edu.memphis.ccrq.lida.proceduralmemory.ProceduralMemoryImpl</class>
```

```

3.     <param name="proceduralMemory.ticksPerStep" type="int">14 </param>
4.     <param name="proceduralMemory.conditionDecayStrategy">defaultDecay</param>
5.     <param name="proceduralMemory.schemeSelectionThreshold" type="double">0.1</param>
6.     <param name="proceduralMemory.contextWeight" type="double">1.0</param>
7.     <param name="proceduralMemory.addingListWeight" type="double">1.0</param>
8.     <param name="proceduralMemory.schemeClass">edu.memphis.ccrp.lida.proceduralMemory.SchemeImpl</param>
9.     <!--
10.    - ____scheme_name__ ____ scheme_label____|_context____|__action_name____|result|baseLabelActiv. -->
11.     <param name="scheme.1">if no leaflet jewel, rotate|(jewel())|action.rotate|()|0.2</param>
12.     <param name="scheme.2">if near brick, avoid it|(brick())|action.avoidBrick|()|0.5</param>
13.     <taskspawner>defaultTS</taskspawner>
14.     <initializerclass>edu.memphis.ccrp.lida.proceduralMemory.BasicProceduralMemoryInitializer</initializerclass>
15. </module>

```

Resultados

Notou-se que com as alterações implementadas o agente consegue selecionar a ação de desviar dos blocos quando necessário.

Possíveis Melhorias

- O algoritmo de roteamento da criatura empregado opera em um grid. Por vezes foi possível notar que não foi possível encontrar um caminho. Esse algoritmo poderia ser melhorado ou outra biblioteca poderia ser utilizada.
- Por vezes ocorrem colisões com blocos no ambiente. Essa situação ocorre provavelmente devido ao replanejamento da rota.