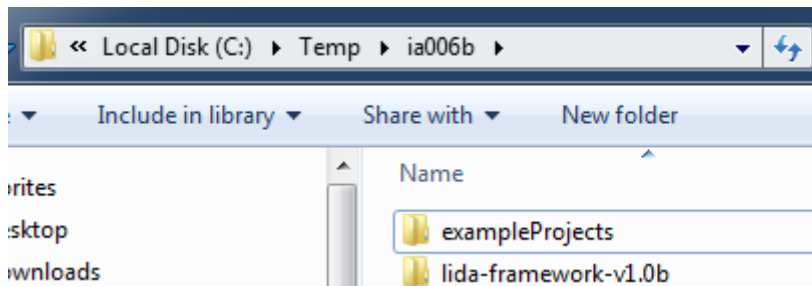


Aula 13

Atividade 1

Feito o download a partir do email recebido e tudo instalado:

- LIDA: <http://ccrg.cs.memphis.edu/assets/code/vce985f830q4j9/lida-framework-v1.2b.zip>
- Java 1.8.0u45: <http://www.oracle.com/technetwork/java/javase/downloads>
- NetBeans IDE 8.0.2: <http://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-142931.html>



Atividade 2

Seguindo o roteiro no arquivo **LIDA-Tutorial-Exercises.pdf**.

Basic Agent Exercise 0: a meta é explorar um agente básico funcional.

Tarefa 1: Explorar estrutura de **Projects**.

myagent.Run	Runs the application
myagent.modules.ButtonEnvironment	Button Environment implementation
myagent.modules.ButtonSensoryMemory	Agent's SensoryMemory implementation
myagent.featuredetectors.ColorDetector	Color feature detector
myagent.featuredetectors.ShapeDetector	Shape feature detector

Explorar estrutura de **Files**. Vale lembrar que os nomes abaixo não são mandatórios pois são configuráveis, inclusive o nome da configuração primária.

lidaConfig.properties	Main configuration file that serves as a directory for the other configuration files needed to build an agent application.
basicAgent.xml	Agent declaration file. This file defines the agent's architecture including the modules and processes the agent application will use.
factoryData.xml	Definition of the elements that can be obtained from the ElementFactory. This file defines several Node, Link, Strategy, and Task types. These element types are referenced <i>by name</i> , e.g. "defaultDecay", in the agent declaration file. (These same names are also used by several framework classes to obtain new elements.)
guiPanels.properties	Configuration file for the GuiPanels used by the framework's GUI. These panels can be added, removed, and/or customized using this file.

Basic Agent Exercise 1: as metas são familiarizar-se com a GUI, compreender a barra de tarefas e seus comandos e controles, compreender o **Button Environment** e aprender sobre **Logging e ConfigurationFiles**.

Tarefa 1: Encontrar o botão **Start/Pause**, pressioná-lo algumas vezes para começar e parar a simulação, identificar o status da simulação (pausada ou rodando) e observar o avanço do **ticks**.

Start / Pause Paused Current tick: 0 Step mode 0 Run ticks Tick duration (ms): 1

Start / Pause Running Current tick: 10782 Step mode 0 Run ticks Tick duration (ms): 1

Start / Pause Paused Current tick: 17060 Step mode 0 Run ticks Tick duration (ms): 1

Tarefa 2: Compreender o controle **Step Mode** que roda uma certa quantidade de iterações. Informamos 100 para analisar e apertamos algumas vezes.

Start / Pause Running Current tick: 17887 Step mode 100 Run ticks Tick duration (ms): 1

Start / Pause Running Current tick: 17987 Step mode 100 Run ticks Tick duration (ms): 1

Tarefa 3: Sair do **Step Mode**, encontrar o controle da duração do tempo de **tick**, ou **tick duration**, com o sistema rodando alterar com as setas de 20 a 0 e observar a mudança da velocidade da aplicação.

Start / Pause Running Current tick: 22046 Step mode 100 Run ticks Tick duration (ms): 20

Start / Pause Running Current tick: 220593 Step mode 100 Run ticks Tick duration (ms): 0

Start / Pause Running Current tick: 396922 Step mode 100 Run ticks Tick duration (ms): 1

Tarefa 4: Com a aplicação rodando, explorar a aba **Logging** da aplicação onde cada linha indica uma entrada do logger (mais detalhes na API do Java), alteramos a seleção para compreender o funcionamento e foi dado um aviso que grandes volumes de log podem reduzir a velocidade de funcionamento do sistema.

Logger: GLOBAL Logging level: INFO Clear log

Logger: myagent.modules.ButtonEnvironment Logging level: FINEST Clear log

```
0000028214 :0000430685 :INFO :myagent.modules.ButtonEnvironment -> Button 2 pressed
0000028215 :0000430700 :FINE :myagent.modules.ButtonEnvironment -> Blue circle displayed.
0000028216 :0000430800 :FINE :myagent.modules.ButtonEnvironment -> Blue circle displayed.
0000028217 :0000430805 :INFO :myagent.modules.ButtonEnvironment -> Button 2 pressed
0000028218 :0000430900 :FINE :myagent.modules.ButtonEnvironment -> Red square displayed.
0000028219 :0000430925 :INFO :myagent.modules.ButtonEnvironment -> Button 1 pressed
0000028220 :0000431000 :FINE :myagent.modules.ButtonEnvironment -> Red square displayed.
0000028221 :0000431045 :INFO :myagent.modules.ButtonEnvironment -> Button 1 pressed
0000028222 :0000431100 :FINE :myagent.modules.ButtonEnvironment -> Red square displayed.
0000028223 :0000431165 :INFO :myagent.modules.ButtonEnvironment -> Button 1 pressed
```

Tarefa 5: Estudar a aba **ConfigurationFiles** do painel de controle e observar que ele apresenta os dados do arquivo **LidaConfig.properties**.

Key	Value
lida.gui.commands	configs/guiCommands.properties
lida.agentdata	configs/basicAgent.xml
lida.gui.enable	true
lida.gui.panels	configs/guiPanels.properties
lida.logging.configuration	configs/logging.properties
lida.elementfactory.data	configs/factoryData.xml

Basic Agent Exercise 2: as metas são familiarizar-se com o **feature detector**, com o **running attention codelet**, estudar diversos painéis pré-definidos e customizar o painel. Na preparação, tivemos que alterar o **lida.agentdata** para outro arquivo de configuração.

```
#Agent properties
lida.agentdata=configs/basicAgent_ex2.xml.
lida.elementfactory.data=configs/factoryData.xml
```

Tarefa 1: Configuramos e executamos a aplicação com a nova configuração. Observado, como sugerido e visto abaixo, que a detecção de **red e square** acontece, porém de **blue e circle** não acontece.

The image displays two screenshots of the LIDA application interface, illustrating the state of the ButtonEnvironment and the PAM Table during a feature detection task.

Top Screenshot: The ButtonEnvironment panel is empty. The PAM Table shows the following state:

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,0000	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	0,0000	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000

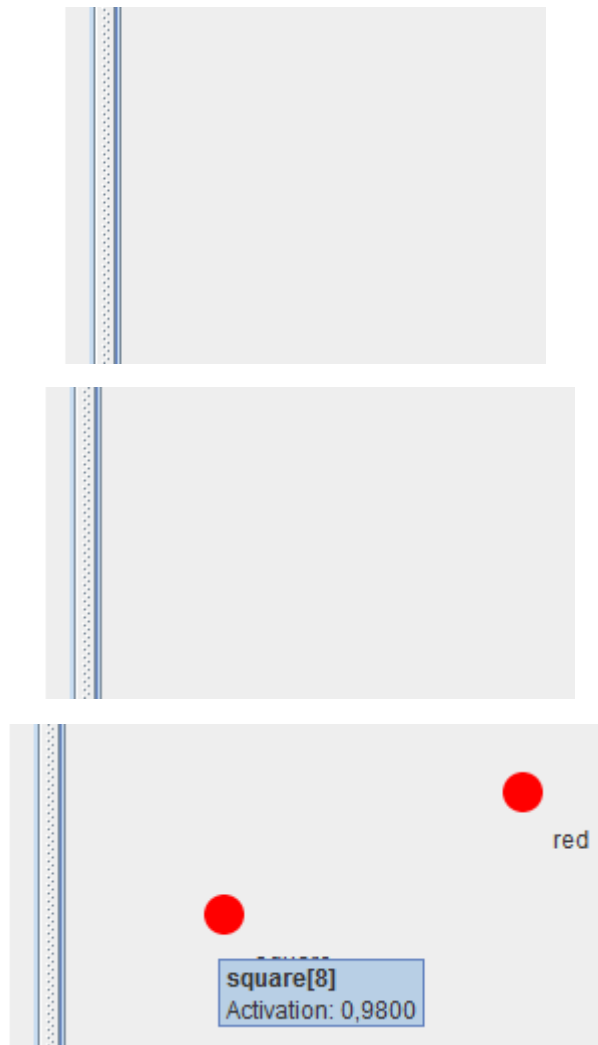
Bottom Screenshot: The ButtonEnvironment panel displays a blue circle. The PAM Table shows the following state:

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,0000	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	0,0000	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000

The screenshot shows the LIDA software interface. On the left, the 'ButtonEnvironment' window displays a red square in the center of a white workspace, with 'Button 1' and 'Button 2' at the bottom. On the right, the 'Perceptual Buffer' window is active, showing a table of nodes and their activation levels.

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,9800	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	0,9800	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000

Tarefa 2: Abrir o **PerceptualBuffer** e analisar como ele muda no tempo. Quando a entrada está vazia, nada é percebido. Quando tem **red square** é percebido e aparece o nível de ativação atual quando apontamos o mouse, e quando aparece **blue circle** também fica vazio pois a configuração não detecta o **blue circle**.



What causes the panel to become empty? Entrada vazia ou **blue circle** já que não tem detecção. **What module of the LIDA Model is the Perceptual Buffer in?** Está no modulo Workspace. **Where do the nodes in this buffer come from?** Vêm da Perceptual Associative

Memory (PAM) quando atingem um limiar mínimo de ativação. **How do they differ from the nodes in other modules?** São cópias dos nós da PAM, são Activatable e não Learnable.

Tarefa 3: Abrir o **GlobalWorkspace** e analisar os dados apresentados, na parte de cima temos as coalizões atualmente formadas e suas informações e abaixo um histórico recente das coalizões vencedoras (a mais recente no topo).

Coalition ID	Activation	Coalition NodeStructure	Creating AttentionCodelet	Sought Content

Tick at broadcast	Broadcast count	Coalition Activation	Broadcast NodeStructure	Broadcast Trigger
45750	390	0.4000	Nodes (red[4, square[8]) Links ()	IndividualCoalitionActivationTrigger
45650	389	0.9800	Nodes (red[4, square[8]) Links ()	IndividualCoalitionActivationTrigger
45590	388	0.9200	Nodes (red[4, square[8]) Links ()	IndividualCoalitionActivationTrigger
45530	387	0.9500	Nodes (red[4, square[8]) Links ()	IndividualCoalitionActivationTrigger

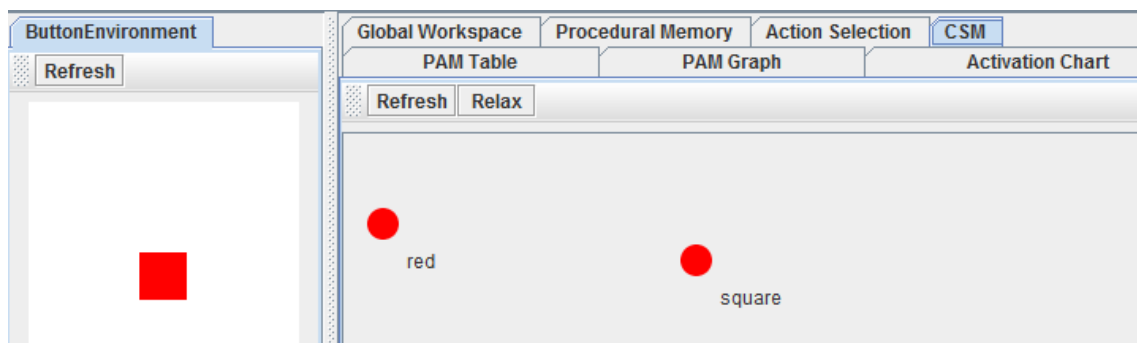
Coalition ID	Activation	Coalition NodeStructure	Creating AttentionCodelet	Sought Content
708	0.5400	Nodes (red[4, square[8]) Links ()	BasicAttentionCodelet[7]	Nodes (red[4, square[8]) Links ()
709	0.2400	Nodes (red[4, square[8]) Links ()	BasicAttentionCodelet[7]	Nodes (red[4, square[8]) Links ()

Tick at broadcast	Broadcast count	Coalition Activation	Broadcast NodeStructure	Broadcast Trigger
46275	392	0.9600	Nodes (red[4, square[8]) Links ()	IndividualCoalitionActivationTrigger
46215	391	0.9600	Nodes (red[4, square[8]) Links ()	NoBroadcastOccurringTrigger
45750	390	0.4000	Nodes (red[4, square[8]) Links ()	IndividualCoalitionActivationTrigger
45650	389	0.9800	Nodes (red[4, square[8]) Links ()	IndividualCoalitionActivationTrigger

Where do the coalitions in the Global Workspace come from? As coalizões vêm do Workspace e a vencedora faz o broadcast para os outros. **Where do they go? Think about how the answer to these questions depends on whether we are discussing the LIDA Model or a specific agent implemented using the Framework.** Somente ocorre o broadcast para objetos que implementem BroadcastListener e se registram no GlobalWorkspace.

Tarefa 4: Fechamos a aplicação para alterar a configuração do painel e adicionar uma aba nova, alterando o arquivo **guiPanels.properties**.

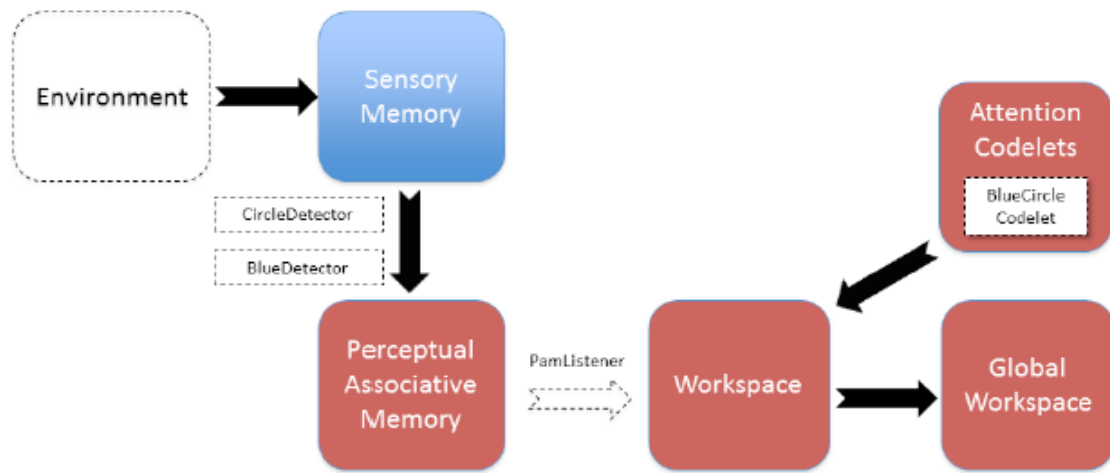
```
# TASK 4 INSERT YOUR CODE HERE *****
#name = panel title, class name, Position [A,B,C,FLOAT, TOOL], tab order, Refresh after load, parameters
perceptualBufferGraph = CSM,edu.memphis.corg.lida.framework.gui.panels.NodeStructurePanel,B,7,Y,Workspace.CurrentSituationalModel
*****
```



How are the contents of the Current Situational Model related to the contents of the Perceptual Buffer? to the contents of PAM? Think about how the answer to these questions depends on whether we are discussing the LIDA Model, or a specific agent implemented in the Framework. CSM apresenta o que acontece atualmente no ambiente externo/interno com entradas advindas da PAM e da memória declarativa. No caso de teste, não existe declarativa.

Basic Agent Exercise 3: as metas são alterar a declaração do agente e detectar os **blue circle** com uma série de adições. Na preparação vamos apontar o **lida.agentdata** para **basicAgent_ex3.xml** e continuar. Nossas alterações serão os espaço em branco tracejados.

```
#Agent properties
lida.agentdata=configs/basicAgent_ex3.xml
lida.elementfactory.data=configs/factoryData.xml
```



Tarefa 1: modo mais fácil de edição usando o navegador inferior

The screenshot shows an IDE interface for editing an XML file. On the left is a file explorer showing a project structure with folders like 'dist', 'lib', 'nbproject', 'src', 'test', and files like 'build.xml' and 'manifest.mf'. The main editor displays XML code with line numbers from 17 to 174. A `<submodules>` tag is highlighted in yellow. The XML code includes taskspawners, submodules, and listeners sections.

Tarefa 2: adicionando declaração de **Environment** em `<submodule>` abaixo de `ActionSelection`. Observe o vermelho ativado e o azul não.

The screenshot shows the IDE interface for editing XML code. The file explorer on the left shows a list of modules: 'PerceptualAssociativeMemory', 'Workspace', 'AttentionModule', 'StructureBuildingCodelet', 'GlobalWorkspace', 'ProceduralMemory', 'ActionSelection', 'Environment', 'SensoryMotorMemory', and 'listeners'. The XML editor on the right shows the code for the 'ActionSelection' module, with a new `<module name="Environment">` tag added below it, highlighted in red. The XML code includes class declarations, parameters, and taskspawners.

LIDA Framework

File Panels Help

Start / Pause Running Current tick: 7967 Step mode 0 Run ticks Tick duration (ms): 20

ButtonEnvironment Refresh

Global Workspace Procedural Memory Action Selection CSM

PAM Table PAM Graph Activation Chart

Refresh

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,9600	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	0,9600	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000

Button 1 Button 2

LIDA Framework

File Panels Help

Start / Pause Running Current tick: 29117 Step mode 0 Run ticks Tick duration (ms): 100

ButtonEnvironment Refresh

Global Workspace Procedural Memory Action Selection CSM

PAM Table PAM Graph Activation Chart

Refresh

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,0000	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	0,0000	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000

Button 1 Button 2

Tarefa 3: Adicionamos o <listener> do tipo PamListener e observamos abaixo o comportamento.

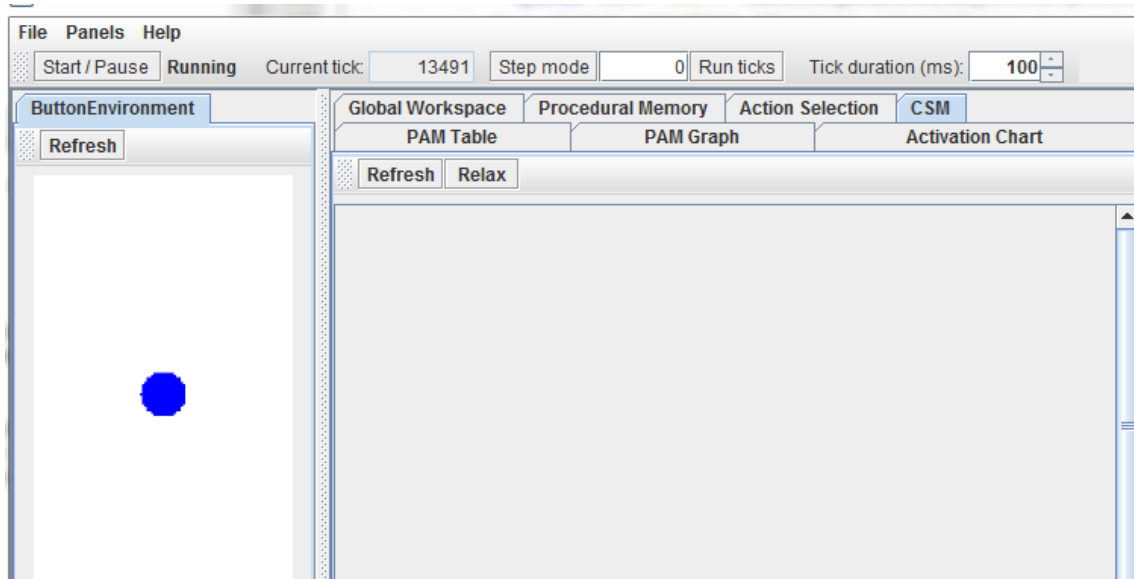
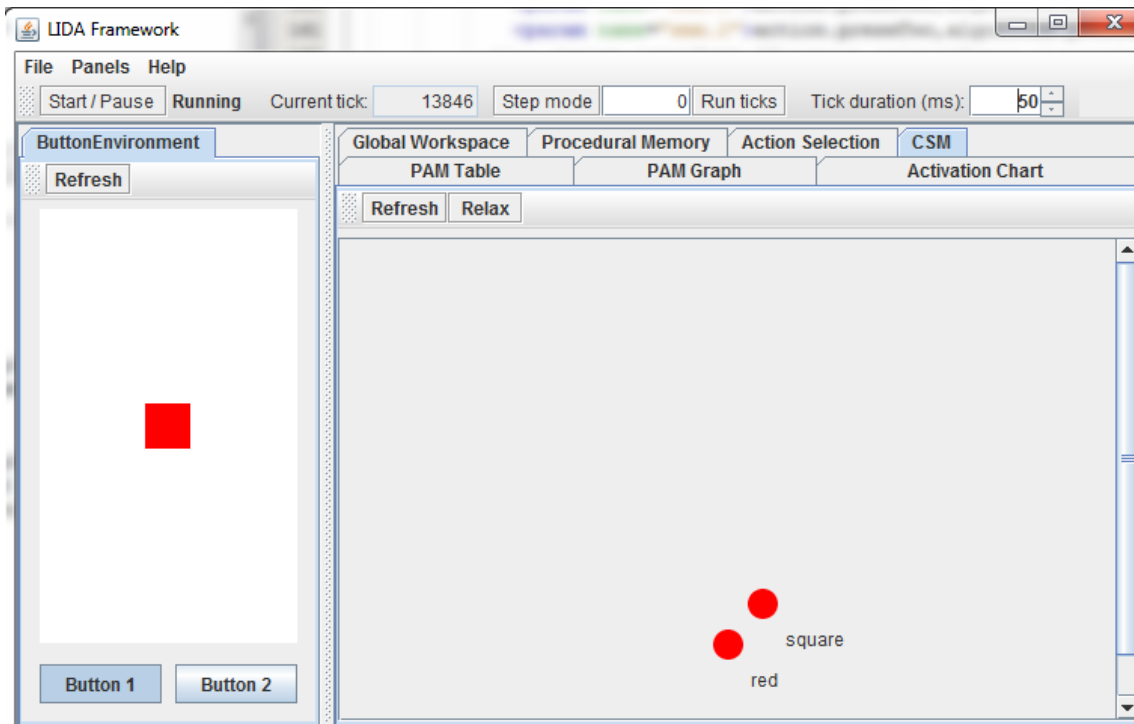
listeners

- listener
 - listenerType (edu.memphis.corg.lida.pam.PamListener)
 - moduleName (PerceptualAssociativeMemory)
 - listenerName (Workspace)
- listener
 - listenerType (edu.memphis.corg.lida.globalworkspace.BroadcastLis...)
 - moduleName (GlobalWorkspace)
 - listenerName (ProceduralMemory)
- listener
- listener

```

172 <!-- ----->
173 <listener>
174 <listenerType>edu.memphis.corg.lida.pam.PamListener</listenerType>
175 <moduleName>PerceptualAssociativeMemory</moduleName>
176 <listenerName>Workspace</listenerName>
177 </listener>
178
179
180 <!-- ----->
181 <listener>
182 <listenerType>edu.memphis.corg.lida.globalworkspace.BroadcastListener</listenerType>
183 <moduleName>GlobalWorkspace</moduleName>

```



Tarefa 4/5: Adicionados o detector de azul e o detector de círculo. Vemos abaixo a alteração em código e as detecções funcionais.

Navigation tree for `srcAgent_ex3.xml`:

- dist
- lib
- srcAgent_ex3.xml - Navigator
 - version="1.0" encoding="UTF-8"
 - lida xmlns="http://ccrg.cs.memphis.edu/LidaXMLSchema", xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://ccrg.cs.memphis.edu/LidaXMLSchema http://ccrg.cs.memphis.edu/LidaXMLSchema/LidaXMLSchema.xsd"
 - taskmanager
 - taskspawners
 - submodules
 - module name="SensoryMemory"
 - module name="PerceptualAssociativeMemory"
 - class (edu.memphis.ccrq.lida.pam.PerceptualAssociativeMem...)
 - param name="pam.Upscale", type="double" (.7)
 - param name="pam.Downscales", type="double" (.6)
 - param name="pam.Selectivity", type="double" (.5)
 - param name="nodes" (red,blue,square,circle)
 - taskspawner (defaultTS)
 - initialTasks
 - task name="RedDetector"
 - task name="SquareDetector"
 - task name="BlueDetector"
 - task name="CircleDetector"
 - initializerclass (edu.memphis.ccrq.lida.pam.BasicPamInitializer)

```

55 .....
56 <task name="SquareDetector">
57   <tasktype>ShapeDetector</tasktype>
58   <ticksperrun>3</ticksperrun>
59   <param name="area" type="int">40</param>
60   <param name="backgroundColor" type="int">-1</param>
61   <param name="node" type="string">square</param>
62 </task>
63 <!-- INSERT YOUR CODE HERE ***** -->
64
65 <task name="BlueDetector">
66   <tasktype>ColorDetector</tasktype>
67   <ticksperrun>3</ticksperrun>
68   <param name="color" type="int">-16776961</param>
69   <param name="node" type="string">blue</param>
70 </task>
71 <task name="CircleDetector">
72   <tasktype>ShapeDetector</tasktype>
73   <ticksperrun>3</ticksperrun>
74   <param name="area" type="int">31</param>
75   <param name="backgroundColor" type="int">-1</param>
76   <param name="node" type="string">circle</param>
77 </task>

```

LIDA Framework

File Panels Help

Start / Pause Running Current tick: 5681 Step mode 0 Run ticks Tick duration (ms): 50

ButtonEnvironment Toggles system operation

Refresh

Global Workspace Procedural Memory Action Selection CSM

PAM Table PAM Graph Activation Chart

Refresh

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,0000	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	0,0000	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000

Button 1 Button 2

LIDA Framework

File Panels Help

Start / Pause Running Current tick: 4471 Step mode 0 Run ticks Tick duration (ms): 20

ButtonEnvironment

Refresh

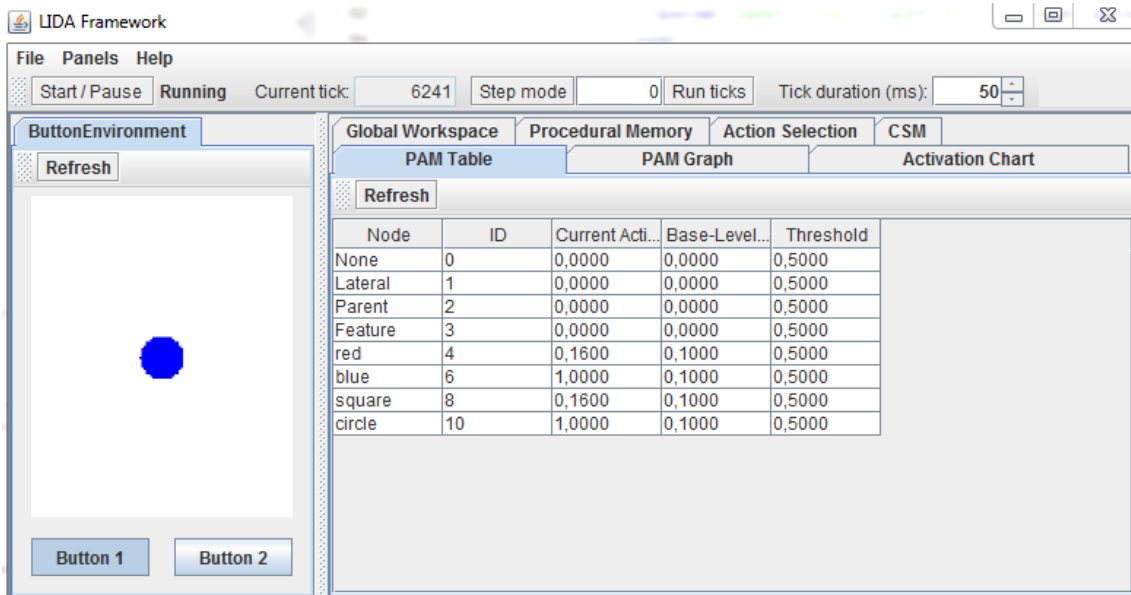
Global Workspace Procedural Memory Action Selection CSM

PAM Table PAM Graph Activation Chart

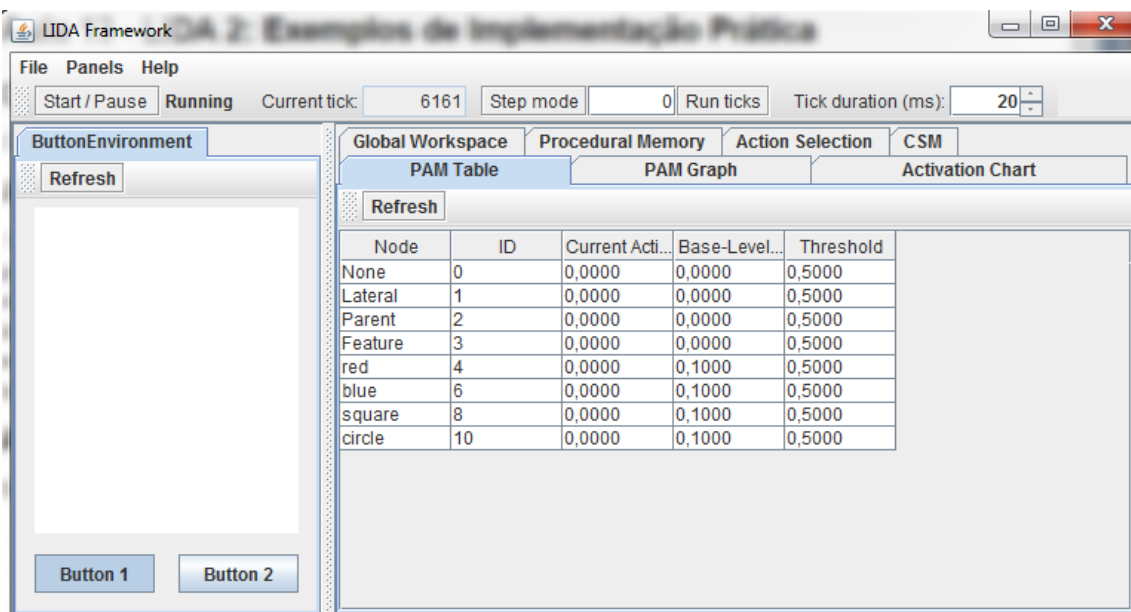
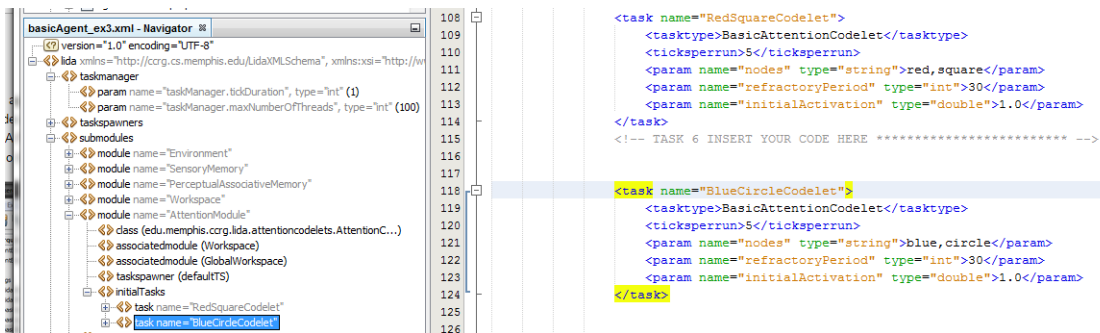
Refresh

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	1,0000	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	1,0000	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000

Button 1 Button 2



Tarefa 6: Abrir o AttentionModule encontrar <initialtasks> copiando RedSquareCodelet e editando para detectar o círculo azul. Abaixo a detecção e cada botão pressionado.




LIDA Framework - DA 2: Exemplos de Implementação Prática

File Panels Help

Start / Pause Running Current tick: 4866 Step mode 0 Run ticks Tick duration (ms): 20

ButtonEnvironment

Refresh



Button 1 Button 2

Global Workspace Procedural Memory Action Selection CSM

PAM Table PAM Graph Activation Chart

Refresh

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,9600	0,1000	0,5000
blue	6	0,0000	0,1000	0,5000
square	8	0,9600	0,1000	0,5000
circle	10	0,0000	0,1000	0,5000


LIDA Framework - DA 2: Exemplos de Implementação Prática

File Panels Help

Start / Pause Running Current tick: 5446 Step mode 0 Run ticks Tick duration (ms): 20

ButtonEnvironment

Refresh



Button 1 Button 2

Global Workspace Procedural Memory Action Selection CSM

PAM Table PAM Graph Activation Chart

Refresh

Node	ID	Current Acti...	Base-Level...	Threshold
None	0	0,0000	0,0000	0,5000
Lateral	1	0,0000	0,0000	0,5000
Parent	2	0,0000	0,0000	0,5000
Feature	3	0,0000	0,0000	0,5000
red	4	0,0000	0,1000	0,5000
blue	6	1,0000	0,1000	0,5000
square	8	0,0000	0,1000	0,5000
circle	10	1,0000	0,1000	0,5000

Opcional: Analisando os detectores. Primeiro o ColorDetector e depois ShapeDetector, ambos abaixo.

```

24 L */
25 public class ColorFeatureDetector extends BasicDetectionAlgorithm{
26
27     /*
28     * Red rgb value
29     */
30     private int soughtColor = 0xFFFF0000;
31     private Map<String, Object> smParams = new HashMap<String, Object>();
32
33     @Override
34     public void init() {
35         super.init();
36         smParams.put("mode", "color");
37         smParams.put("x", 50);
38         smParams.put("y", 50);
39         soughtColor = (Integer) getParam("color", 0xFFFF0000);
40     }
41
42     @Override
43     public double detect() {
44         int color = (Integer) sensoryMemory.getSensoryContent("visual", smParams);
45         //cosine is a better comparison
46         if(soughtColor == color){
47             return 1.0;
48         }
49         return 0.0;
50     }
51 }

```

```

public class ShapeFeatureDetector extends BasicDetectionAlgorithm {

    public static final int TOLERANCE = 5;
    private int soughtArea = 1000;
    private int backgroundColor = 0xFFFFFFFF;
    private Map<String, Object> smParams = new HashMap<String, Object>();

    @Override
    public void init() {
        super.init();
        smParams.put("mode", "all");

        soughtArea = (Integer) getParam("area", 1000);
        backgroundColor = (Integer) getParam("backgroundColor", 0xFFFFFFFF);
    }

    @Override
    public double detect() {
        int[] layer = (int[]) sensoryMemory.getSensoryContent("visual", smParams);
        int area=0;
        for(int i=0;i<layer.length;i++){
            if(layer[i]!=backgroundColor){
                area++;
            }
        }
        area=(area*1000)/layer.length;
        if(Math.abs(area-soughtArea) < TOLERANCE){
            return 1.0;
        }
        return 0.0;
    }
}

```

Advanced Exercise 1: podemos executar também sem a GUI. Para isso, acessamos o `lidaConfig.properties` e alterar a propriedade `property lida.gui.enable` para `false`. Os logs podem ser enviados para arquivos ou pra Console. Vale lembrar que o LIDA utiliza o mecanismo padrão de logging do Java.

Advanced Exercise 2: Este exercício visa demonstrar os efeitos de alteração em **ticksPerRun**, para isso lançamos a aplicação com **tickDuration** em 10 e fazemos testes com a configuração normal, depois fizemos com os parâmetros do **RedSquareCodelet** dentro de **Attention** em **<initialtasks>** tag com parâmetros **ticksPerRun=50** e **refractoryPeriod=300**. Observa-se uma alteração na taxa de aparição de coalizões quando o quadrado vermelho é percebido e também altera a propagação/broadcast.