README.md

Aluno: Ricardo Keigo de Sales Andrade

Disciplina: IA941A - Prof. Ricardo Gudwin

Período: 1o Semestre de 2018

Aula 13 - LIDA controlando o WorldServer3D

O codigo-fonte das atividades desenvolvidas pelo aluno encontra-se disponivel em: https://github.com/papeldeorigami/ia941

Atividade 1

O DemoLida foi baixado através do link indicado na página da disciplina. Seu código foi estudado.

Atividade 2

A janela do Lida foi habilitada com a propriedade lida.gui.enable=true.

Foi implementado um painel que exibe a acao selecionada atualmente. Para desenvolver esse painel, tomou-se como base o código de um painel do lida-framework, disponivel aqui: [https://github.com/CognitiveComputingResearchGroup/lida-framework/blob/master/src/edu/memphis/ccrg/lida/framework/gui/panels/ActionSelectionPanel.java]

O painel desenvolvido contém apenas um jLabel cujo texto é atualizado com a primeira acao selecionada.

```
public class CurrentActionPanel extends GuiPanelImpl implements
        ActionSelectionListener {
```

...

```
    @Override
     public void receiveAction(Action action) {
         ActionDetail detail = new ActionDetail(TaskManager.getCurrentTick(),
                 currentSelectionCount++, action);
         synchronized (this) {
             selectedActions.addFirst(detail);
             if (selectedActions.size() > selectedActionsSize) {
                 selectedActions.pollLast();
             }
             if (selectedActions.size() > 0) {
                 currentActionLabel.setText(selectedActions.getFirst().getAction().getLabel());
             } else {
                 currentActionLabel.setText("None");
             }
         }
     }
```

Depois, configuramos o painel no guiPanels.properties:

```
# ____A Section____
currentAction=Current action,gui.panels.CurrentActionPanel,A,1,Y
```

Atividade 3

O detector de blocos foi desenvolvido basicamente copiando e colando o detector de joias.

- Criamos um atributo "block" no Environment e no SensoryMemory
- O método updateEnvironment seta o bloco quando está próximo do agente
- O método getState do Environment passa esse bloco quando solicitado pelo SensoryMemory.runSensors
- O método getSensoryContent do SensoryMemory passa o bloco para o BlockDetector
- Desenvolvemos a classe BlockDetector, que retorna o valor de ativação 1 sempre que um bloco é detectado no SensoryMemory
- Configuramos o BlockDetector no Agent.xml e factoryData.xml, seguindo o modelo do JewelDetector.

O problema de se levar o agente a um determinado ponto, desviando de obstáculos, foi dividido nas seguintes etapas:

1 - Modelar o mundo como um grafo, com destino e obstaculos imaginarios

```
- Importamos a biblioteca [https://github.com/xaguzman/pathfinding](pathfinding)
- Definimos um grid de 80 por 60 (o tamanho default do mundo dividido por dez)
- Ocupamos alguns pontos do grid com obstaculos "imaginários", i.e. setamos com o valor 1, sem ter
nenhum objeto criado no mundo ainda
- Apenas para auxiliar o desenvolvimento, determinamos um caminho fixo com um destino fixo
- O caminho é representado por um vetor chamado "path"
```

2 - Estabelecer um mecanismo para planejar e estimular a o agente a andar em direcao a um objetivo

```
- Salvar o grid e o plano no sensory memory
- Fazer o agente detectar o "próximo passo": o primeiro elemento do vetor path que está na sensory
memory
```

Dentro do SensoryMemory, portanto, definimos um atributo "destination", que é inicializado com o targetDestination definido no Environment. Para o destination, criamos um DestinationDetector. O Environment utiliza o creature.getPosition() para saber se já chegamos no destino. Enquanto não chegar no destino, retorna o targetDestination. O SensoryMemory, por sua vez, irá executar o algoritmo de busca, e retornar o primeiro elemento do vetor de melhor caminho para definir como destino.

3 - Implementar Acao Replanejar - Detectar objetos e estimular o agente a replanejar o destino (Não implementado ainda)

4 - preencher com objetos do mundo

```
- Fazer link com o WS3D para: detectar clique do mouse e detectar obstaculos
```

(Não implementado ainda)