

Relatório 04 - LIDA: Controlando o WorldServer3D

Leonardo de Oliveira Ramos RA171941

Junho 2018

1 Entrega

Na pasta DemoLida se encontra o código fonte da criatura. Para executar apenas abra a pasta dist e digitar make run para rodar, make stop para matar os processos e make restart pra reiniciar o programa.

2 LIDA

Lida é uma arquitetura cognitiva que se caracteriza pelo seu ciclo cognitivo, onde se seguem etapas de sensoriamento, atenção, consciência e ação.

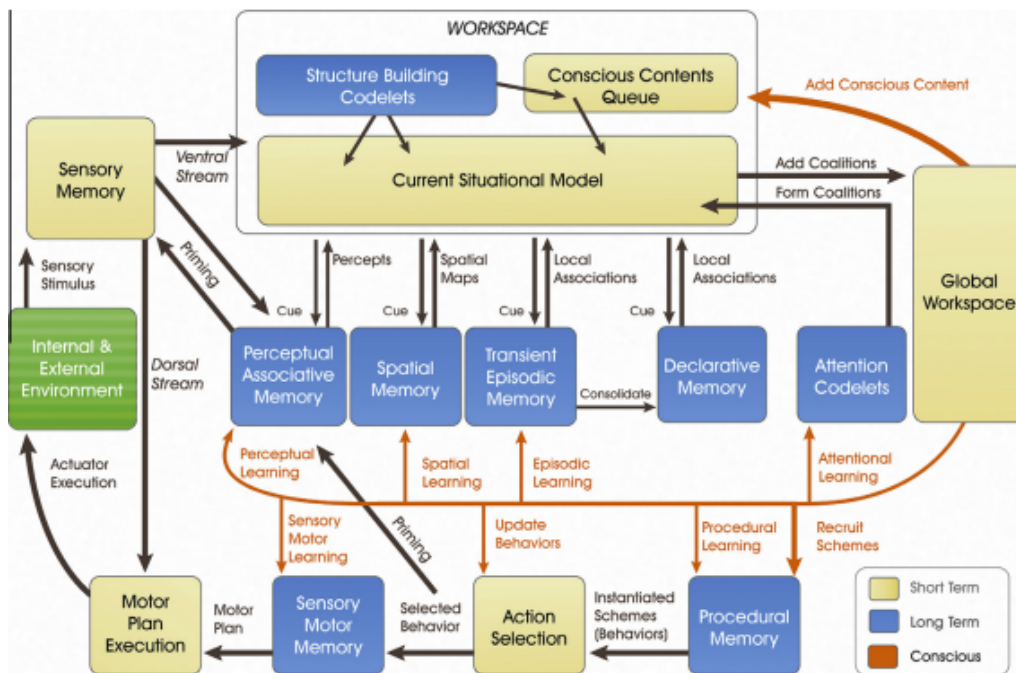


Figure 1: Ciclo cognitivo do modelo LIDA

Desta forma a etapa sensória se dá através da memória sensória, que é enviada tanto a execução diretamente através da dorsal Stream, quanto para etapa de atenção pelo ventral stream. A etapa de atenção que foi inspirada pela teoria do Global Workspace, que Posner sugeriu três funções de atenção:

- alerting: "maintaining an alert state"
- orienting: "focusing our senses on the information we want"
- executive attention: "the ability to manage attention towards goals and planning"

Onde o objeto sensório é trazido para a memória através desta atenção, e em seguida faz coalitions, que disputam pela consciência, se estabelecendo como um objeto em uma das memórias.

3 Algoritmo

Desta vez a criatura tem como abjetivo chegar em um ponto determinado, assim eu escolhi o ponto (0,0) como objetivo, seguindo um comportamento reativo.

A criatura quando encontra uma parede reage a ela, desviando, para andar em direção a origem, assim ele percebe se a parede está muito próxima da borda superior do cenário, caso sim, ele vai pela borda inferior, caso não, desvia pela borda superior.

Caso ele encontre uma joia ou comida muito próxima, ele pega ou come, respectivamente.

4 Teste

Para testar o comportamento da criatura eu coloquei no cenário 4 paredes bem largas que ele tem que desviar para chegar a origem.

5 Funcionamento do código

5.1 Atividade 1 - Baixando o código do LIDA

Primeiramente foi baixado o código do DemoLIDA e o framework da LIDA para se basear e criar o modelo da criatura da atividade 3.

5.2 Atividade 2 - Janela de ações

Para criar uma janela com as ações foi preciso mudar o arquivo XML guiPanels.properties adicionando uma linha da chamada do painel criado por mim.

Para mostrar a ação eu criei uma variável command que logo antes de agir recebe a string da ação que será feita, e através de uma JLabel dentro de um painel, toda vez que tem refresh ele checa o valor de command e coloca na JLabel.

O código que atualiza a label:

```
@Override
public void refresh()
{
    action.setText(environment.getCommand());
}
```

5.3 Atividade 3 - Comportamento da criatura

No código é desenvolvido os seguintes inputs sensórios e as seguintes ações correspondentes:

5.4 Inputs sensórios

- thingAhead: Detecta joia ou comida perto.
- wall: Detecta parede a frente.
- nowall: Detecta quando não tem parede a frete.

5.5 Ações

- gotoOrigin: Vai para (0,0) no mapa, que responde a nowall.
- avoidWall: Desvia de um objeto do tipo BRICK em direção a (0,0), andando até a quina da parede, depois a atravessando, responde a wall.
- get: Pega uma joia ou come uma comida próxima, que responde a thingAhead.

Então para cada um destes estímulos ele faz a ação através do seguinte código.

```
private void performAction(String currentAction) {
    try {
        command = currentAction;
        switch (currentAction) {
            case "avoidWall":
                if (wall != null)
                    creature.moveto(3.0, wallX, wallY);
                break;
            case "gotoOrigin":
                creature.moveto(6.0, 0, 0);
                break;
            case "get":
                creature.move(0.0, 0.0, 0.0);
                if (thingAhead != null) {
                    for (Thing thing : thingAhead) {
                        if (thing.getCategory() == Constants.categoryJEWEL) {
```

```

        creature.putInSack(thing.getName());
    } else if (thing.getCategory() == Constants.categoryFOOD ||
        creature.eatIt(thing.getName()));
    }
    }
    this.resetState();
    break;
default:
    break;
}
} catch (Exception e) {
    e.printStackTrace();
}
}

```

6 Desafios

Nesta atividade eu tentei anteriormente fazer uma estratégia deliberativa primeiro, mas tive dificuldades em 2 aspectos, que foram: mapear o cenário para ter conhecimento do ambiente, e desenvolver uma maneira de fazer ações discretas, ou seja, mover para um ponto do cenário e parar lá, por exemplo. Por estes dois motivos eu decidi fazer um comportamento reativo.