

IA941 - Laboratório de Arquiteturas Cognitivas

R1

Patrick de Carvalho Tavares Rezende Ferreira - 175480

20 de março de 2022

1 Introdução

A proposta desta primeira atividade (atividade R1) foi a instalação e familiarização com o ambiente de simulação WS3D[1], uma ferramenta desenvolvida por pesquisadores da FEEC e que será utilizada pelos alunos ao longo do semestre para os experimentos.

2 Métodos

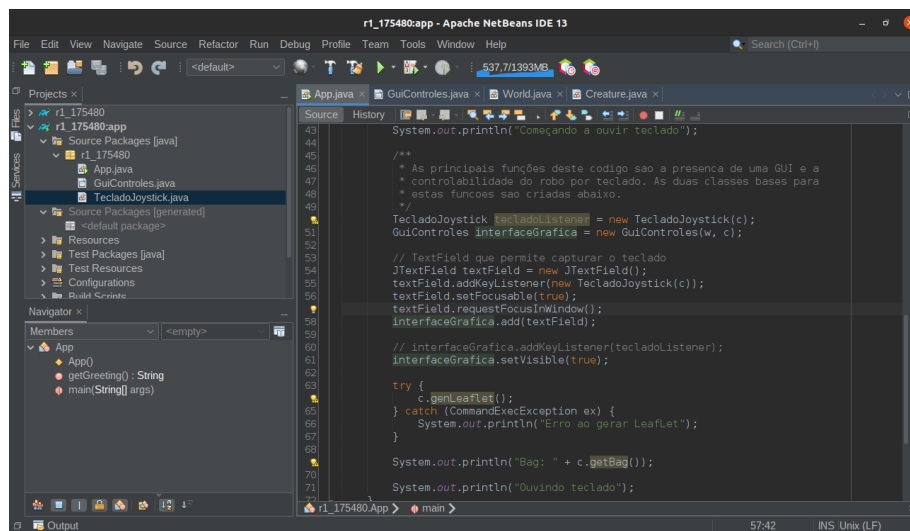


Figura 1: IDE Netbeans[3]

O primeiro passo foi a instalação do ambiente de desenvolvimento *NetBeans* (figura 1), uma IDE com suporte à linguagem Java e ferramentas de interface gráfica (GUI).

Para usuários Linux, a única configuração de ambiente necessária foi a instalação do *opendjdk-8*. O próximo passo foi o *download* e instalação do WS3D através de seu repositório.

A interface de comunicação nativa deste ambiente se dá via sockets, que são porém um meio trabalhoso de se lidar com o ambiente. Para solucionar tal situação, foi desenvolvido um segundo *toolkit* para comunicação com o WS3D, o WS3dDProxy[2], que também é fornecido através do repositório do grupo de pesquisa. Após seguir as instruções de instalação dos dois projetos, inclusive com alterações no *Gradle* do projeto da tarefa R1, foi iniciada a construção da interface gráfica de controle manual do ambiente WS3D com uma única criatura em operação.

3 Resultados

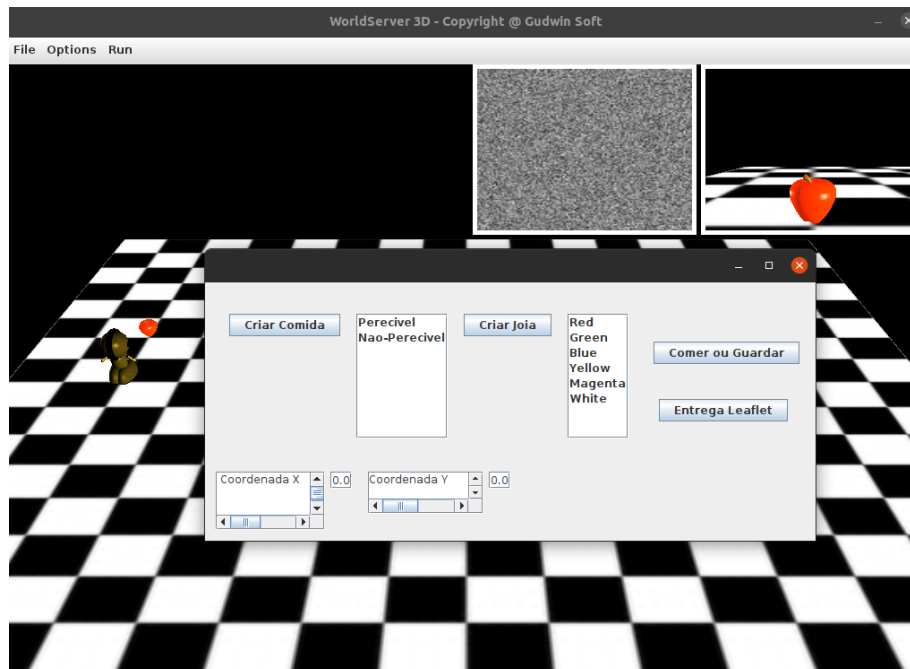


Figura 2: Interface gráfica.

Foi desenvolvida uma GUI (figura 2) de interação com o ambiente que permite ao usuário criar jóias e comidas em coordenadas específicas, além de guardar jóias capturadas pela criatura e comer alimentos capturados pela mesma. Para implementar a interface gráfica, foi criada uma classe *GuiControles* (figura 3) que herda da classe *JFrame* do Java, que fornece métodos rápidos para criação de interface visual.

Também foi desenvolvida uma classe *TecladoJoystick()* (figura 4) que herda de *KeyListener()*, de forma que seja possível controlar a direção da criatura através do teclado. O padrão de controle escolhido é o "WASD", explicado na tabela 1. A classe *TecladoJoystick* recebe um objeto do tipo *Creature* (a única criatura em jogo) como referência para poder executar as ações.

Por último, é válido ressaltar que a GUI (figura 2) disponibiliza um botão para converter o *Leaflet* da criatura (um mapa de objetivo em termos de capturas de jóias) em pontos no *score*, depois que esta já houver capturado o conjunto de jóias especificado no próprio *Leaflet*, o qual é

```

public class GuiControles extends javax.swing.JFrame {

    public World w;
    public Creature c;

    private double coordX;    //coordenada X selecionada na interface
    private double coordY;    //coordenada Y selecionada na interface
    private int itemjListCores; //item selecionado na lista de cores
    private int itemjListTipoDeComida; //item selecionado na lista de comidas.

    /**
     * Creates new form gui_controles
     */
    public GuiControles(World w, Creature c) {
        this.coordX = 0;
        this.coordY = 0;
        this.itemjListTipoDeComida = 0;
        this.itemjListCores = 0;
        this.w = w;
        this.c = c;
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
}

```

Figura 3: Classe *GuiControles*.

Tabela 1: Padrão de controle "WASD".

Tecla	Ação
W	À Frente
A	Giro Anti-horário
S	Ré
D	Giro horário
Espaço	Parar

gerado de aleatoriamente ao início da simulação pela função *main()*. A pontuação obtida pode ser verificada pelo próprio painel do simulador, conforme exibido na figura 5.

Para executar o código desenvolvido para esta tarefa, basta executar o arquivo “run_r1_175480.sh”, o qual inicializará o WS3D e, após 10 segundos, executará o código da tarefa R1.

4 Conclusão

Este experimento foi útil para fazer os alunos se familiarizem com as interfaces de comunicação e principais comandos do simulador WS3D, conhecimento que será útil nas próximas aulas quando estivermos interagindo com códigos de arquiteturas cognitivas que irão interagir com a simulação de maneira automática.

```

public class TecladoJoystick implements KeyListener {

    public TecladoJoystick(Creature creature) {
        this.creature = creature;
    }

    private Creature creature = null;

    public int direcao = 0;

    @Override
    public void keyPressed(KeyEvent e) {

        System.out.println("Tecla pressionada!");

        int key = e.getKeyCode();
        try {
            // Codigos convencionados para andar em cada direcao (ou rodar)
            if (key == KeyEvent.VK_A) {
                direcao = 1;
                this.creature.rotate(-1);
                System.out.println("esquerda");
                return;
            }

            if (key == KeyEvent.VK_D) {
                direcao = 2;
                this.creature.rotate(1);
            }
        }
    }
}

```

Figura 4: Classe *TecladoJoystick*.

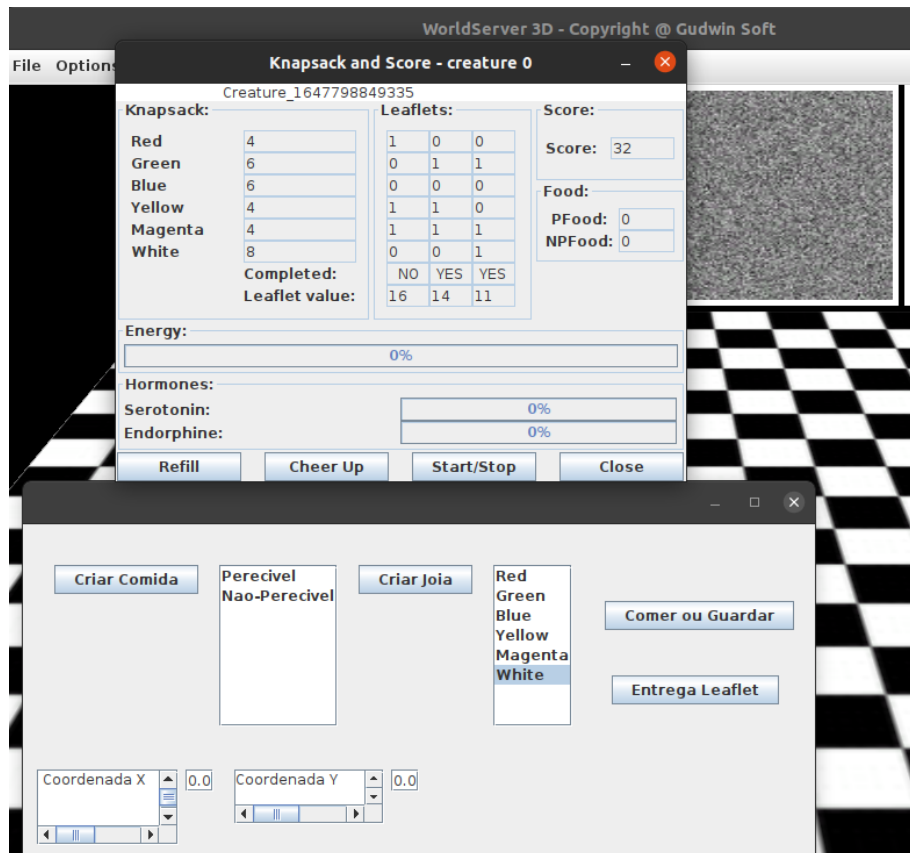


Figura 5: *Score* da criatura após entregar alguns *Leaflets*.

Referências

- [1] *CST-Group/ws3d: WorldServer3D - A Virtual Environment for Experiments with Artificial Creatures*. URL: <https://github.com/CST-Group/ws3d> (acesso em 20/03/2022).
- [2] *CST-Group/WS3DProxy: Proxy for accessing the ws3d Virtual Environment directly from Java, without requiring socket communication details*. URL: <https://github.com/CST-Group/WS3DProxy> (acesso em 20/03/2022).
- [3] Fundación de Software Apache. *Welcome to Apache NetBeans*. 2020. URL: <https://netbeans.apache.org/> (acesso em 20/03/2022).