

Nome: Fabio Grassiotto  
RA: 890441  
Disciplina: IA941A, 1º S 2018

## Aula 1 – Controlador Manual Para Ambiente Virtual

### Objetivo

O objetivo desta primeira aula foi o contato com o ambiente de trabalho a ser utilizado no curso.

### Atividade 1

No site da disciplina (<http://faculty.dca.fee.unicamp.br/gudwin/courses/IA941/aula1>) estavam disponíveis as atividades da primeira aula.

### Atividade 2

Foi executado o download do código do ambiente virtual WordServer3D (WS3D) diretamente para um novo projeto na IDE do Netbeans.

### Atividade 3

Conforme a orientação do site, foi desenvolvido um sistema de controle para uma criatura no ambiente virtual do WS3D.

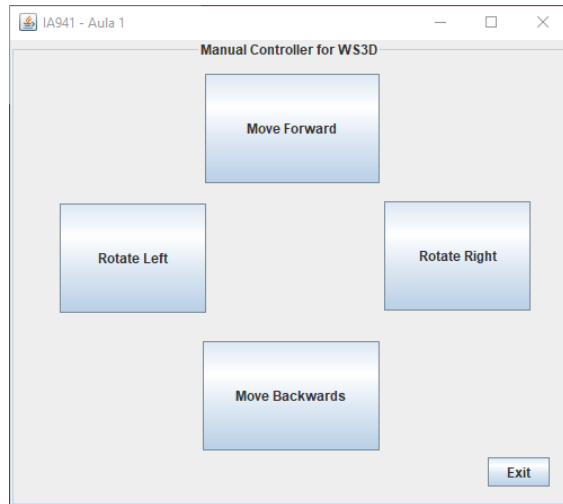
### Instruções para Execução do Controlador Manual

Executar o script run-aula1.sh em uma shell de comando do Linux. O ambiente virtual será inicializado automaticamente e posteriormente a tela do Controlador Manual será exibida.

### Detalhamento da Implementação do Controlador Manual

Foram seguidos os seguintes passos para a implementação do controlador:

- Integração da biblioteca WS3DProxy a partir do repositório do Github no código do sistema de controle.
- Implementação de uma interface gráfica na IDE do Netbeans utilizando para tanto o recurso de construção de interfaces gráficas (GUI Builder) conforme a figura abaixo.



Botões definidos na interface:

**Move Forward:** Move a criatura para a frente.

**Move Backwards:** Move a criatura para trás.

**Rotate Right:** Gira a criatura no sentido horário cerca de 90 graus.

**Rotate Left:** Gira a criatura no sentido anti-horário cerca de 90 graus.

- Estudo da interface de programação da biblioteca WS3DProxy para implementação da movimentação da criatura no ambiente proposto.

## Dificuldades Encontradas

Através das interfaces utilizadas para movimentação não foi possível determinar uma nova posição cartesiana para a criatura no ambiente virtual. Portanto a movimentação implementada consiste nos seguintes passos:

1. Execução da interface que inicializa a movimentação da criatura: **Creature.start.**
2. Execução da interface que especifica uma movimentação com velocidades horizontais e verticais: **Creature.move.**
3. Implementação de uma interface assíncrona em Java utilizando para tanto a classe `SchedulerExecutorService` para determinar a finalização do movimento em outra thread de execução, através da interface: **Creature.stop.**

Para rotação da criatura passos similares foram seguidos.