

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO E ENGENHARIA DE  
COMPUTAÇÃO

FELIPE FISCHER COMERLATO  
LEONARDO EICH

**Grupo Equipe 7**  
**Projeto Space Invaders Utilizando Lua**

Relatório apresentado como requisito parcial para  
a obtenção de conceito na Disciplina de Modelos  
de Linguagens de Programação

Prof. Dr. Lucas Mello Schnorr  
Orientador

Porto Alegre  
2018

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>3</b>
<b>1.1 O Problema.....</b>	<b>3</b>
<b>2 VISÃO GERAL DA LINGUAGEM .....</b>	<b>4</b>
<b>2.1 Modelo Funcional .....</b>	<b>4</b>
<b>2.2 Modelo Orientado a Objetos.....</b>	<b>4</b>
<b>3 RECURSOS FUNCIONAIS .....</b>	<b>5</b>
<b>3.1 Elementos Imutáveis e Funções Puras .....</b>	<b>5</b>
<b>3.2 Funções Anônimas .....</b>	<b>5</b>
<b>3.3 Currying.....</b>	<b>5</b>
<b>3.4 Pattern Matching .....</b>	<b>5</b>
<b>3.5 Funções de Ordem Superior .....</b>	<b>5</b>
<b>3.6 Funções de Ordem Maior Fornecidas Pela Linguagem .....</b>	<b>5</b>
<b>3.7 Funções como Elementos de Primeira Ordem .....</b>	<b>5</b>
<b>3.8 Recursão.....</b>	<b>5</b>
<b>4 RECURSOS DE ORIENTAÇÃO À OBJETOS .....</b>	<b>6</b>
<b>4.1 Classes .....</b>	<b>6</b>
<b>4.2 Encapsulamento e Proteção dos Atributos .....</b>	<b>6</b>
<b>4.3 Construtores .....</b>	<b>6</b>
<b>4.4 Destrutores.....</b>	<b>6</b>
<b>4.5 Espaços de Nomes Diferenciados.....</b>	<b>6</b>
<b>4.6 Mecanismos de Herança.....</b>	<b>6</b>
<b>4.7 Polimorfismo por Inclusão .....</b>	<b>6</b>
<b>4.8 Polimorfismo Paramétrico .....</b>	<b>6</b>
<b>4.9 Polimorfismo por Sobrecarga .....</b>	<b>6</b>
<b>4.10 Delegates .....</b>	<b>6</b>
<b>5 CONCLUSÃO FINAL.....</b>	<b>7</b>
<b>REFERÊNCIAS.....</b>	<b>8</b>

## 1 INTRODUÇÃO

Este trabalho tem como objetivo o estudo de uma linguagem de programação moderna com características híbridas contextualizando os conceitos vistos em aula ao longo do semestre e, por fim, analisar e avaliar diferentes linguagens de programação, seguindo os critérios vistos em aula.

A tarefa principal do trabalho consiste em experimentar e comparar as características e funcionalidades orientadas a objeto e funcionais da linguagem de programação escolhida. De posse de uma linguagem, é necessário escolher um problema a ser solucionado com ela. O problema será, então, implementado duas vezes na mesma linguagem: uma delas usando somente Orientação a Objetos e a outra usando somente características funcionais.

O desenvolvimento do trabalho encontra-se em <[https://github.com/felipefcomerlato/mlp\\_equipe7\\_2018-2](https://github.com/felipefcomerlato/mlp_equipe7_2018-2)>.

### 1.1 O Problema

O problema a ser resolvido neste trabalho é o desenvolvimento do jogo Space Invaders. O jogo expõe o jogador como uma espaçonave que deve destruir as espaçonaves inimigas que querem invadir o planeta do jogador. Na medida que elas avançam na tela (de cima para baixo), o jogador guia sua espaçonave horizontalmente e efetua disparos para destruir todas as ondas de inimigos que se seguem, como visto na figura 1.1

Figura 1.1: Tela do jogo Space Invaders



## 2 VISÃO GERAL DA LINGUAGEM

A linguagem escolhida para o trabalho foi **Lua**. Esta é uma linguagem de alto nível, leve, poderosa e multiparadigma: pode ser considerada de *scripting*, imperativa, funcional ou orientada a objetos. Um dos pontos fortes da linguagem é sua semântica extensível, à qual podem ser adicionadas novas funcionalidades sem alterar suas características originais. Uma das características principais da linguagem é sua única estrutura de dados: tabelas - que podem ser usadas para representar arrays comuns, sequências, tabelas de símbolos, conjuntos, registros, grafos, árvores, etc. (MANUAL..., 2018)

### 2.1 Modelo Funcional

Lua oferece suporte ao modelo de programação funcional mas nos proporciona algumas dificuldades, como a de copiar tabelas, por exemplo, já que tal estrutura de dados atribuída a uma variável é apenas uma referência. Tal dificuldade contraria uma das principais características do próprio paradigma funcional: a de usar funções puras, criando novas estruturas ao invés de alterar uma estrutura original. Por outro lado, Lua nos permite representar funções anônimas através de blocos de código chamados de *trechos*, característico no modelo funcional. (MANUAL..., 2018)

### 2.2 Modelo Orientado a Objetos

O suporte para a programação no paradigma orientado a objetos utilizando Lua é muito vasto. É possível encontrar inúmeros tutoriais para a solução de diversos problemas na linguagem fazendo uso deste paradigma. Como não vimos diversos pré-requisitos necessários para a implementação deste trabalho, não podemos analisar com mais propriedade o desempenho da linguagem Lua neste modelo de programação.

## **3 RECURSOS FUNCIONAIS**

### **3.1 Elementos Imutáveis e Funções Puras**

### **3.2 Funções Anônimas**

### **3.3 Currying**

### **3.4 Pattern Matching**

### **3.5 Funções de Ordem Superior**

### **3.6 Funções de Ordem Maior Fornecidas Pela Linguagem**

### **3.7 Funções como Elementos de Primeira Ordem**

### **3.8 Recursão**

## **4 RECURSOS DE ORIENTAÇÃO À OBJETOS**

### **4.1 Classes**

### **4.2 Encapsulamento e Proteção dos Atributos**

### **4.3 Construtores**

### **4.4 Destrutores**

### **4.5 Espaços de Nomes Diferenciados**

### **4.6 Mecanismos de Herança**

### **4.7 Polimorfismo por Inclusão**

### **4.8 Polimorfismo Paramétrico**

### **4.9 Polimorfismo por Sobrecarga**

### **4.10 Delegates**

## **5 CONCLUSÃO FINAL**

## REFERÊNCIAS

MANUAL de Referência de Lua 5.2. 2018. Available from Internet: <<https://www.lua.org/manual/5.2/pt/manual.html#2.6%E2%80%93Co-rotinas>>.