

Proposta para o TCC  
MAC499 – Trabalho de Formatura  
Supervisionado  
Desenvolvimento de uma Linguagem de Scripts  
para Jogos

*Aluno: Daniel Barbosa Stein*  
*Supervisor: Flávio Soares Corrêa da Silva*

## Introdução

Em desenvolvimento de jogos, muitas vezes nos deparamos com funcionalidades que precisam ser dinâmicas. Um meio de suportá-las seria criar na própria engine do jogo todas as possibilidades, mas isto se mostra impossível quando as possibilidades são enormes.

A solução é desenvolver uma linguagem de scripts na qual não precisamos mexer no código fonte da engine para adicionar novas funcionalidades, ou configurar as que são dinâmicas.

Com isso, a linguagem de script tem uma parte genérica (como qualquer linguagem, que define variáveis, funções e comandos básicos) e uma parte específica à engine (que define comandos específicos à funcionalidade desejada).

Como exemplo de comandos básicos temos: definição de variáveis, modificação de valor de variáveis, chamada de funções e controle de fluxo (if, compare). De comandos específicos temos como comandos ligados diretamente à engine que a utiliza, como a criação e configuração de um objeto que a engine disponibiliza ou alguma funcionalidade que possa ser configurada.

Normalmente temos várias variações da linguagem (ou mais de uma linguagem) em uma mesma engine para objetivos específicos. Podemos ter uma linguagem A para o scripting central, uma linguagem B para a AI, e uma C para o sistema de partículas.

Dependendo do uso, podemos gerar linguagens sem os comandos básicos (assim a linguagem seria totalmente especializada) como seria o caso de uma linguagem para um sistema de partículas, a qual em muitos casos apenas necessita de uma linguagem para configurar seus parâmetros.

## Objetivos

O objetivo deste trabalho é implementar uma linguagem de script simples interpretada (sem o uso de geradores automáticos como javacc e bison) para ser utilizada em jogos como ferramenta de scripting.

## A Linguagem

A linguagem, nomeada de SimpleScript, terá dois módulos: o Interpretador, que compilará o arquivo source em um bitcode interno, e a SSVM (ScriptScript Virtual Machine) que fará a execução do bitcode gerado pelo Interpretador. O Interpretador compilará o código em tempo real.

Os comandos da linguagem seguirá o formato “comando(parametros);”, parecido com a linguagem C. O “;” será obrigatório como marca de finalização do comando.

As funções terá o formato “[nomedafuncao] (retorno)(parametros)”. Cada parâmetro terá o formato “tipo,[nomedoparametro]” e as variáveis serão definidas como “[nomedavariável]” e terão os seguintes tipos básicos: int, float, string, bool e char.

## **Atividades Realizadas**

Estudo de linguagens de programação simples para jogos, implementação de uma linguagem de teste simplificada.

## **Cronograma das Atividades**

- *Julho*
  - Iniciar implementação da primeira versão da linguagem.
  - Implementar na linguagem:
    - > variáveis (globais)
    - > comandos básicos (if, compare, etc)
- *Agosto*
  - Implementar na linguagem :
    - > blocos de comandos
    - > variáveis locais
    - > chamada de blocos
- *Setembro*
  - Implementar na linguagem:
    - > preprocessor de inclusão de outros arquivo de source
    - > funções com retorno e parâmetros
- *Outubro*
  - Implementar na linguagem:
    - > defines
    - > structures
- *Novembro*
  - Otimização da Linguagem

## **Estrutura da Monografia**

A monografia pretente seguir a seguinte organização:

- I. Introdução
- II. Motivação
- III. Objetivos

- IV. Conceitos e Métodos Utilizados
- V. Atividades Realizadas
- VI. Resultados
- VII. Conclusão
- VIII. Planos Futuros
- IX. Bibliografia