



# **Trabalho Prático de Sistemas Operacionais**

## **Parte 1**

*Carolina Arenas Okawa (Nº 10258876)*  
*Luiz Miguel Di Mano Saraiva (Nº 10425420)*  
*Matheus Carvalho Raimundo (Nº 10369014)*

*Orientadora: Prof. Kalinka Castelo Branco*

USP - São Carlos

Novembro/2018

## **Índice Analítico**

1. Considerações Iniciais	2
2. Descritivo do Jogo	2
3. Tema do Jogo	3
4. Níveis e Jogabilidade	4
5. Bibliografia	4

# 1. Considerações iniciais

O objetivo desse documento é descrever o jogo a ser implementado e como serão usados os conceitos de sistemas operacionais em seu desenvolvimento.

## 2. Descritivo do Jogo

O jogo que será desenvolvido pelo grupo será de Tower Defense. Esse estilo de jogatina possui, por definição:

- uma torre, que deve ser defendida de qualquer ameaça;
- inimigos, que tentarão destruir a torre;
- ferramentas e armas, que serão usadas para destruir os inimigos.

Ao definir este projeto, chegamos a conclusão de que cada inimigo e cada ferramenta/arma pode ser uma *thread* diferente. Isso porque eles trabalham de forma independente uns dos outros. Além disso, há outra *thread* para a interface gráfica e para a atualização desta. A priori, temos intenção de inserir áudio (música e efeitos sonoros) no jogo, o que classificaria mais *threads* em execução.

Os inimigos podem atacar simultaneamente a torre. A partir daí, será necessário um **semáforo** que controla a resistência restante da torre (*health*). Além disso, as armas/ferramentas podem atacar simultaneamente um mesmo inimigo, e isso exigiria outro semáforo para controlar a vida restante do inimigo (no caso, um semáforo para cada inimigo).

O jogador recebe pontos para cada inimigo derrotado. Estes pontos podem ser usados para comprar ferramentas/armas melhores, visto que a cada nível os inimigos se tornam mais resistentes e se apresentam em maior número. O jogo também permitirá que o usuário salve e restaure a jogatina, além de manter uma tabela com as melhores pontuações (*scoreboard*).

A intenção é de se usar a linguagem de programação Java, e executar o código em uma Java Virtual Machine. O compilador do Java não disponibiliza muitas ferramentas em relação a memória utilizada no programa, contudo, sabe-se que a Máquina Virtual do Java é

separada em 3 segmentos: heap (que armazena os objetos Java), non-heap (que armazena as classes carregadas e outros metadados) e JVM Code (que armazena o bytecode Java). É possível controlar o tamanho do segmento de HEAP da Java Virtual Machine ao executar um programa usando as variáveis *-Xms* (memória disponível para alocação inicialmente) e *-Xmx* (máximo de memória disponível para alocação). Além disso, com *-Xss* é possível definir o tamanho da *stack* de cada *thread* da Java Virtual Machine.

Exemplo via terminal: `java -Xss1024k -Xms128m -Xmx1g [programa]`

### 3. Tema do Jogo

O grupo adotará como tema para o jogo de Tower Defense uma disciplina relacionada ao curso de computação: cálculo.

A torre representará a aprovação na disciplina (“5 bola”). Os inimigos são símbolos e elementos da matemática, que tentarão de tudo para reprovar o jogador na disciplina (destruir a torre “5 bola”). As armas/ferramentas são objetos que serão usados para destruir os símbolos matemáticos e conseguir manter a aprovação na disciplina.

A cada nível, a dificuldade aumenta e símbolos matemáticos mais complexos tomarão conta da tela. Se no nível anterior você usou como ferramenta uma aula no YouTube, por exemplo, talvez no próximo seja necessário algo a mais para manter a torre segura (aprovação na disciplina).

Este é um tema um tanto complexo, mas com um tom humorístico que representa a dificuldade dos alunos de computação em disciplinas relacionadas à cálculo. De fato, não é um tema definitivo e se definido necessário, podemos alterar o tema do jogo a posteriori. Contudo, a jogatina continuará sendo do estilo Tower Defense.

## 4. Níveis e Jogabilidade

Cada nível representa um semestre da faculdade. Obviamente a dificuldade aumenta à medida que se avançam os níveis. Ao concluir todos os níveis o jogador se forma e sua pontuação é inserida no *scoreboard*.

Dentro de cada nível (semestre) existem 5 hordas, que são as provas: P1, P2, P3, P4 e P5. A cada prova, a dificuldade e o número de questões (inimigos) aumenta.

A interação com o jogo acontece com o uso do mouse na própria interface gráfica.

## 5. Bibliografia

- Memory Segments:  
<<http://www.cs.uwm.edu/classes/cs315/Bacon/Lecture/HTML/ch10s04.html>>  
Acesso em 28 de outubro de 2018.
- JVM Memory Structure:  
<<https://www.yourkit.com/docs/kb/sizes.jsp>>  
Acesso em 29 de outubro de 2018.
- What are the Xms and Xmx parameters when starting JVMs?  
<<https://stackoverflow.com/questions/14763079/what-are-the-xms-and-xmx-parameters-when-starting-jvms>>  
Acesso em 29 de outubro de 2018.
- JVM Memory Segments and JIT Compiler  
<<https://stackoverflow.com/questions/35379383/jvm-memory-segments-and-jit-compiler>>  
Acesso em 29 de outubro de 2018.