



Relatório do Trabalho Prático de Sistemas Operacionais

Parte 2

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

Sumário

1. Introdução	2
2. Descrição do Jogo	2
3. Descrição dos Conceitos que Foram Aplicados	2
4. Tema do Jogo	3
5. Como jogar	4
6. Bibliografia	7

1. Introdução

O trabalho foi proposto com o objetivo de aprofundar os conhecimentos do aluno nos assuntos abordados na disciplina de Sistemas Operacionais, mais especificamente Threads, Semáforos e Gerência de Memória.

A ideia seria a implementação de um jogo que utilizasse esses conceitos, e o tema do jogo seria de escolha livre do grupo.

2. Descrição do Jogo

O jogo desenvolvido pelo grupo foi do gênero Tower Defense. Esse estilo de jogo possui, por definição:

- Uma torre, que deve ser defendida de qualquer ameaça;
- Inimigos, que tentarão destruir a torre;
- Armas, que serão usadas para destruir os inimigos.

3. Descrição dos conceitos que foram aplicados

A implementação dos conceitos vistos em aula pedidos no trabalho foi feita da seguinte forma:

- Cada elemento do jogo funciona como uma thread diferente: inimigos, armas, a torre, a interface gráfica e uma thread principal que controla o funcionamento e a lógica do jogo.
- Cada *thread* possui também seu próprio semáforo mutex. Por exemplo, os inimigos têm um semáforo que controla a vida, dano e velocidade, e assim por adiante; existe também um semáforo principal que cuida da função de *pause* do jogo.

- O jogo foi totalmente implementado na linguagem Java, portanto executa na Java Virtual Machine. A Java Virtual Machine é subdividida em 3 segmentos principais: heap (que armazena os objetos Java), non-heap (que armazena as classes carregadas e outros metadados) e JVM Code (que armazena o bytecode Java). Além disso, os objetos são subdivididos em relação a sua expectativa de vida:
 - Eden: objetos recém criados;
 - Survivor 1 e Survivor 2: objetos sobreviventes ao Garbage Collection realizado no Eden (quando este fica sobrecarregado);
 - Old Generation: sobreviventes ao Garbage Collection realizado no Survivor 1 e Survivor 2;
 - PermGen: objetos mais antigos.

4. Tema do jogo

O grupo adotou como tema para o jogo 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 reprovar o jogador na disciplina (destruir a torre “5 bola”). As armas são objetos que serão usados para destruí-los e conseguir a aprovação na disciplina (como vídeo aula e professores conceituados).

5. Como jogar

Para iniciar o jogo, o usuário deve:

- ❑ Em alguns sistemas operacionais, clicar duas vezes no arquivo Calculus.jar é o bastante para executar o jogo.
- ❑ Em outros sistemas operacionais, é necessário iniciar a Java Virtual Machine via terminal através do comando `java -jar Calculus.jar`. Note que, o arquivo Calculus.jar deve estar acompanhado do código-fonte e recursos do jogo (dentro do diretório “src”). Além disso, o ambiente de execução (*workspace*) deve ser o mesmo em que o diretório “src” se encontra.
- ❑ Além do arquivo *.jar, pode-se compilar (*javac*) separadamente cada uma das classes e executar a classe principal *Main* (*java*).
- ❑ É possível executar a aplicação também através de uma IDE (ambiente de desenvolvimento integrado) como Eclipse ou IntelliJ IDEA.

O jogo começa apresentando a seguinte tela:

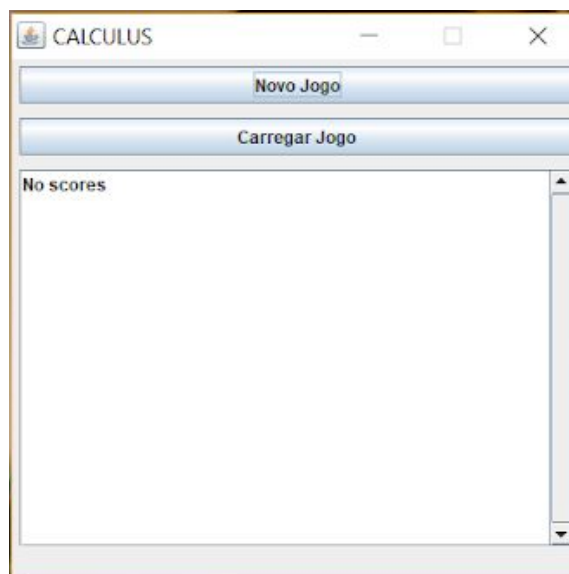


Figura 1 - Tela inicial do jogo.

Nela, o jogador pode ver uma *leaderboard*, ordenada primeiramente pelo *score*, e depois pelo tempo de duração do jogo em ordem crescente. Além disso, existem dois botões, um para iniciar um novo jogo e outro para carregar um arquivo com um jogo já existente.

Ao clicar em “Novo Jogo”, é exibida uma tela pedindo o nome do jogador (para posteriormente associá-lo a sua pontuação) e, então, o jogo inicia.

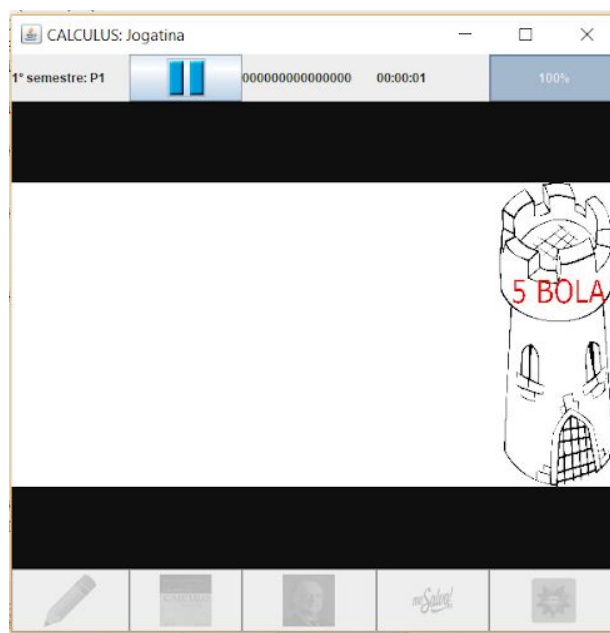


Figura 2 - Tela do jogo.

Nesse momento, o jogador consegue ver, no canto superior esquerdo, o semestre e a prova atual. Ao seu lado, há um botão de *pause*, seu *score*, o tempo de jogo e a vida da torre.

O objetivo do jogo é defender a torre, posicionada na direita, que simboliza a aprovação na disciplina. Os inimigos (elementos relacionados à disciplina de cálculo, como limites, derivadas e integrais) aparecem do lado esquerdo, se movimentam em direção à torre para atacá-la e diminuir sua vida. O dano que cada um deles causa e a velocidade com que se move varia de acordo com o semestre e prova correspondentes.

Na barra inferior estão as armas que o jogador pode usar para defender a torre. Alguns instantes após o início, elas começam a ser liberadas. As armas causam danos diferentes e estão ordenadas em ordem crescente de dano. São elas:

- Lápis (arma básica que causa um dano mínimo);
- Guidorizzi (livro de cálculo);
- Claudio Possani (professor da Univesp);
- Me Salva! (canal de video aulas);
- Wolfram (ferramenta matemática).

Para utilizar uma arma, basta clicar nela com o *mouse*. Cada vez que é utilizada, diminui a vida dos inimigos e gera pontos para o jogador, porém depois fica bloqueada por um determinado tempo, exigindo do jogador uma estratégia para usá-las.

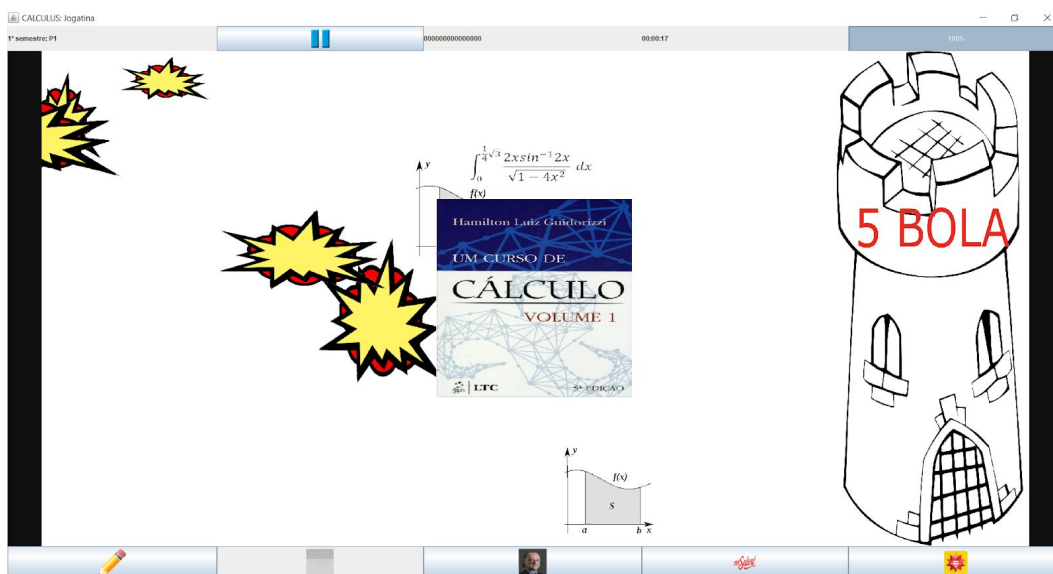


Figura 3 - Tela de uso da arma “Guidorizzi”

Cada vez que os inimigos de uma prova são mortos, outra prova inicia, totalizando cinco provas por semestre. Quando elas terminam, o semestre troca. Existem 5 semestres. Somente após finalizá-los, se ganha o jogo.

Quando o jogo termina, seja porque o jogador perdeu ou ganhou, sua pontuação é salva e mostrada na página inicial.

Durante o jogo, é possível *pausar* e, ao fechar a tela, escolher ou não salvar o jogo. Assim, é mostrada uma tela para escolher um local de salvamento e depois é possível restaurar o jogo por meio do botão “Carregar Jogo” na tela inicial.

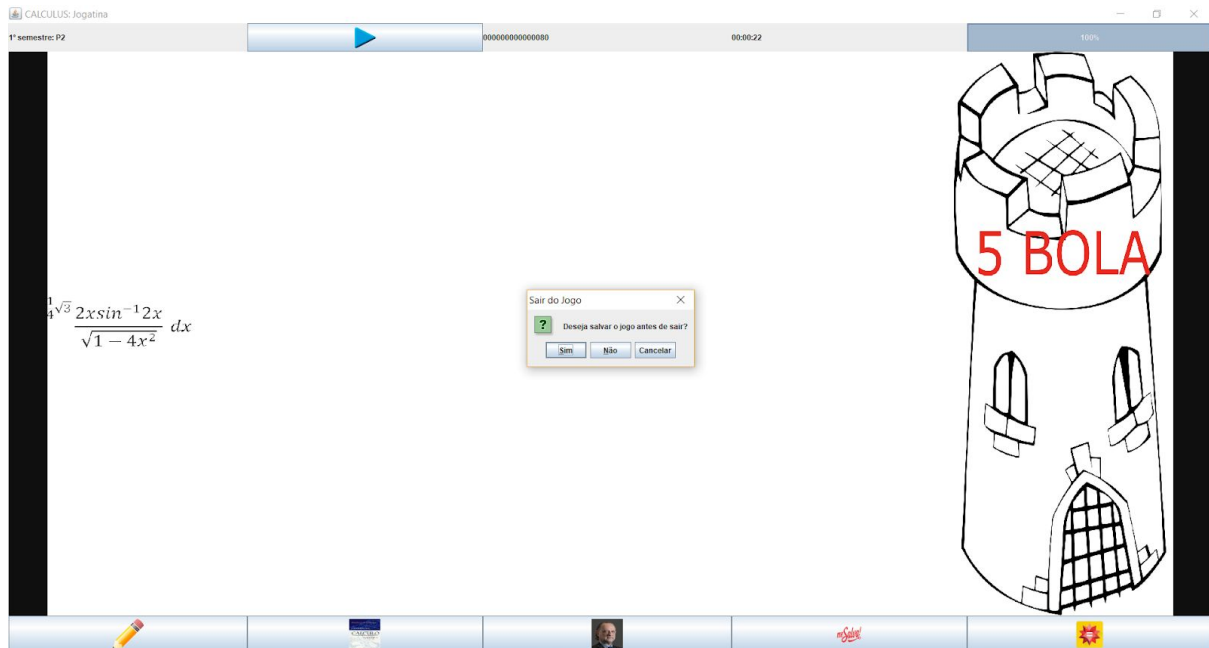


Figura 4 - Tela com a opção de salvar o jogo

6. 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.