

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO

Confecção de um jogo envolvendo *threads* e semáforos

Sistemas Operacionais I

Profa. Dra. Kalinka Regina Lucas Jaquie Castelo Branco

2º semestre 2021

Nome: Gustavo Henrique Brunelli

NUSP: 11801053

Nome: Matheus Henrique de Cerqueira Pinto

NUSP: 11911104

Nome: Pedro Lucas de Moliner de Castro

NUSP: 11795784

Entrega 2: **SuperDino** - jogo utilizando *threads* e semáforos

1 Proposta

O projeto proposto consiste no desenvolvimento de um jogo análogo ao do navegador Google Chrome, Dino, comumente conhecido como “**Jogo do Dinossauro**”. O grupo entende que, por ser uma aplicação popular, os mecanismos do jogo, mesmo alterados, serão de fácil assimilação e que, portanto, o jogo poderá ser potencialmente utilizado.

Na aplicação originalmente proposta, diferentemente da implementação padrão, seriam utilizados, como recursos extras, conceitos para presença de dois personagens simultâneos, em contextos separados, e pontuação relacionada ao acúmulo de moedas. Entretanto, devido à complexidade da construção da proposta para dois jogadores, a aplicação desenvolvida foi simplificada para a reconstrução do jogo Dino usando os recursos gráficos do terminal, através de bibliotecas específicas, além dos conceitos de *threads* e *semáforos*, requisitos obrigatórios no desenvolvimento da aplicação.

O conceito de *threads* foi utilizado para separar a leitura das entradas do teclado da atualização dos objetos e da tela e os *semáforos* foram aplicados na forma de *semáforos binários*, também chamados de *mutex*, para exclusão mútua nas duas regiões críticas introduzidas.

2 Implementação

2.1 Embasamento técnico

Para posicionamento do personagem, geração de obstáculos e *background*, foi utilizado como base um tutorial ([clique aqui para acessá-lo](#)) que consistia na produção de um jogo utilizando a biblioteca **ncurses** e a linguagem **C++**. De uma proposta

também de construção de Dino em terminal ([clique aqui para acessá-la](#)), foram embasados alguns *sprites*. Para geração das artes textuais em ASCII, foi utilizado o site PatorJK ([clique aqui para acessá-lo](#)).

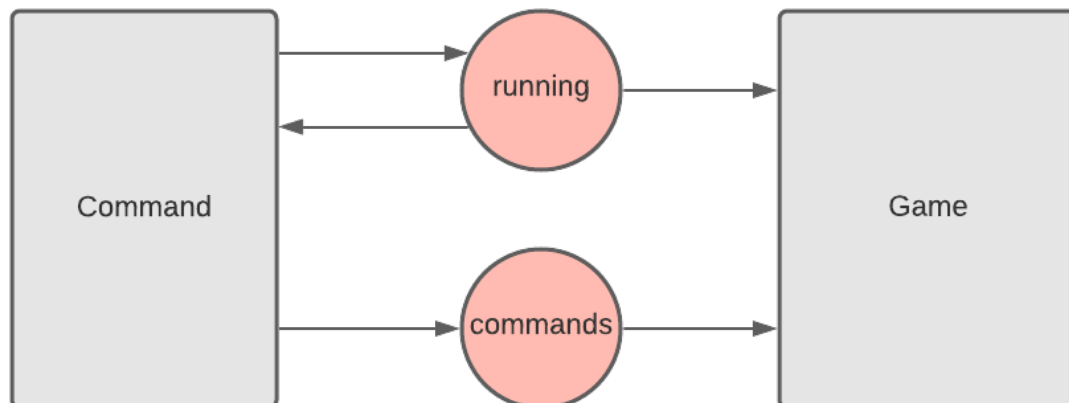
2.2 Construção

O código foi dividido em duas *threads*: **Command** é um captador de comandos do teclado que envia uma mensagem cada vez que uma tecla correspondente a uma ação é pressionada e **Game** é responsável pelo *loop* principal do jogo, onde há atualização da posição dos objetos, checagem de colisão e renderização da tela.

É importante ressaltar que a *thread* **Game** tem uma limitação de execução em 60Hz, sendo essa uma taxa de atualização comum em diversos monitores que possibilita a geração randômica e a aplicação da física sem uma preocupação com taxas altíssimas de atualização por segundo.

Há entre as *threads* duas regiões críticas: **running** é uma variável booleana que indica se a execução do programa deve continuar, ela recebe requisições de leitura de ambas as linhas de execução, mas só é reescrita pelo método `exit()` convocado pela *thread* **Command** quando `[ESC]` é pressionado, já **commands** é um sistema de mensagens assíncrono implementado com uma fila de comandos que é populada pelo método `send()` pela **Command** e consumida com a chamada `receive()` pela **Game**.

Figura 1: Diagrama das *threads* e regiões críticas



2.3 Capturas de tela

Figura 2: Jogo com personagem ainda vivo

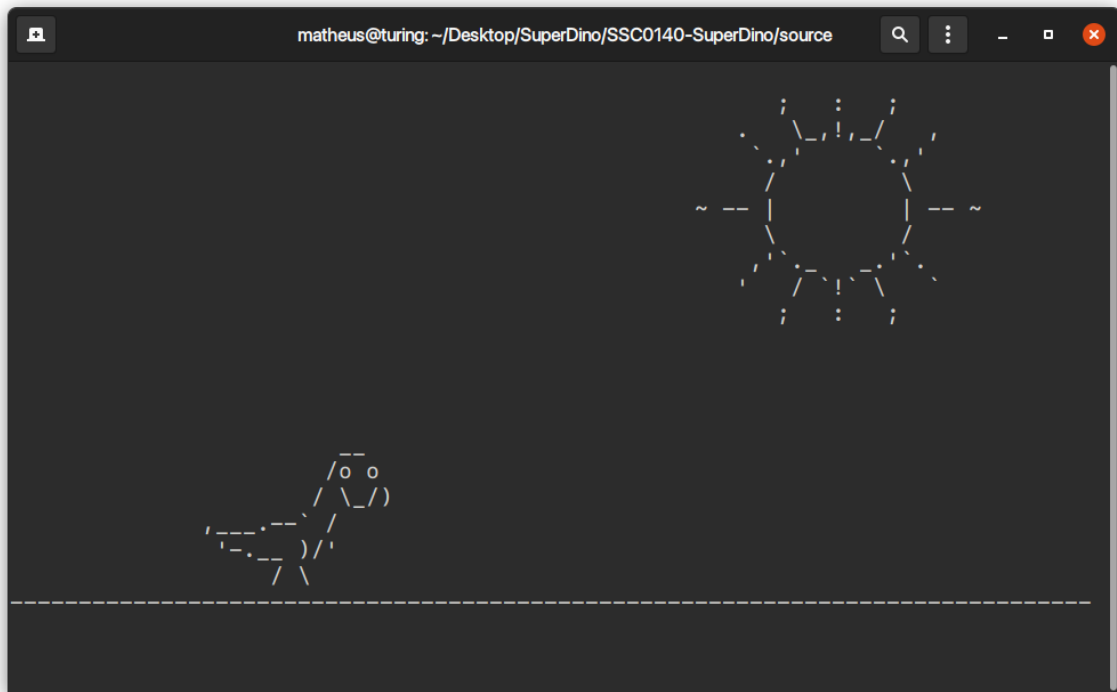


Figura 3: Tela de fim de jogo



3 Manual de uso

3.1 Instalação

O jogo é dedicado aos sistemas operacionais Linux, foi desenvolvido na linguagem **C++** e sua compilação depende dos programas **g++**, **make** e da biblioteca **ncurses**, todas essas dependências podem ser instaladas com a execução dos comandos:





```
sudo apt install g++  
sudo apt install make  
sudo apt install libncurses-dev
```

Para compilar o projeto pela primeira vez, basta entrar no diretório `source` e utilizar a diretiva `make`. Finalmente, para executar o jogo, é necessário chamar o comando `make run`.

```
cd source  
make  
make run
```

3.2 Jogabilidade

O jogo desenvolvido é uma releitura do Dino conhecido por sua implementação no navegador Google Chrome, assim, todos os comandos são parecidos com o original.

-  : Pular
-  : Abaixar
-  : Sair do jogo
-  : Reiniciar o jogo

4 Considerações finais

A construção do jogo permitiu aos autores a utilização prática dos conceitos teóricos estudados durante as aulas de Sistemas Operacionais. Além disso, a proposta de criação de jogo se mostrou desafiadora, uma vez que diversos processos considerados simples precisaram ser implementados manualmente. De forma geral, o grupo considerou efetiva a mescla dos conceitos teóricos, programação e a parte lúdica como requisito de avaliação.