



Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

Disciplina de Programação Orientada a Objeto (SSC0103)

Documentação do Trabalho Final de Programação Orientada a Objeto

*Jogo Educacional para Ensino Introdutório de Comandos da CLI Bash
(Linus Quest)*

*Prof. Dr. Marcio Delamaro
Estagiário PAE Misael Junior*

**São Carlos
Data de entrega: 24 de julho de 2020**

1. INTRODUÇÃO

1.1. SOBRE O PROJETO

O projeto final consiste na Implementação de um jogo que simula um terminal customizado e seguro para o ensino de comandos e manipulação de arquivos pelo CLI Bash. O jogador é imerso em uma narrativa em que o aprendizado dos principais comandos é necessário para avançar.

O programa foi concebido com a linguagem Python 3, interpretado e testado no SO Linux Ubuntu e Windows 10 em WSL.

1.2. O JOGO LINUS QUEST

Linus Quest simula um ambiente de terminal portanto todas interações são realizadas por meio de uma interface textual. Personagens fictícios guiarão o jogador por diversos níveis que testarão seus conhecimentos adquiridos até o dado ponto de forma lúdica.

Inicialmente, o usuário tem de criar um nome de usuário e uma senha para se comunicar com Linus o Pinguim, um protagonista da narrativa. Então, ele se disponibilizará para ajudar, guiando o início do caminho.

No mundo sem interface gráfica, o usuário parte em uma missão para recuperá-la. Desvia dos mares e florestas, cruzando a Ponte da Miragem, passando pelas Cavernas da Criptografia e pelo Deserto da Desordem e chega ao Castelo das Trevas, lar de Delamaro.

Ao finalmente derrotar o antagonista e ter a oportunidade de reconquistar a interface gráfica o jogador deverá ponderar como seria possível fazer muitas coisas poderosas mesmo sem a UI. Então poderá decidir se irá abandonar seu amigo Linus e voltar para interface gráfica ou aderir ao novo mundo do terminal.

```
Capítulo 1 - A Tragédia e o Pinguim
Linus: Não, de novo não...
Linus: O Delamaro Mestre Hacker do Mal precisa parar com essas travessuras.
Linus: Lá vamos nós então, consegue me ouvir err.. ler? (Não consigo te ouvir, responda
teclando "segue", depois pressionando ENTER. Se não me entendeu, tecle "repete" e ENTER.)
user@user:root$ segue
Linus: Perfeito, esse teclado vai ser nossa única forma de comunicação por enquanto, tente
não perdê-lo.
user@user:root$ segue
Linus: Deve ter várias perguntas. Por que o céu é azul? Por que a minha tela está preta?
Como amarro uma gravata?
user@user:root$ _
```

2. OBJETIVOS

O objetivo principal seria criar um meio efetivo de compartilhar nossos conhecimentos sobre o uso de interfaces textuais.

Aprendemos portanto a utilizar a plataforma GitHub e a linguagem Python (no editor Jupyter) para programar uma simulação rudimentar de um terminal Bash genérico.

Para finalizar, compusemos uma narrativa e injetamos-a em falas sucintas e digeríveis para guiar o jogador por níveis criados com o terminal mencionado de base.

3. DESCRIÇÃO DO SOFTWARE

3.1. FUNCIONAMENTO DO JOGO / IMPLEMENTAÇÃO

Ao iniciar o programa, o jogador seria apenas responsável por acompanhar a narrativa por meio do comando fabricado “segue” (ou “repete”, em caso de desentendimento), memorizando e entendendo o contexto de uso dos comandos mostrados, a fim de posteriormente aplicá-los para completar os desafios.

Todos os comandos (com exceção de “segue” e “repete”) são reais e listados na documentação oficial da Bash Shell. Apesar disso, dentro do jogo, seria custoso demais realmente implementar todos os comandos ensinados para uso livre (essencialmente reprogramar uma nova shell), portanto optamos por apenas fazer isso para alguns mais ubíquos E.g. “cd”, “ls”, “touch”, “mkdir”.

Toda a programação foi feita em Python 3 por conta da opção de desenvolvimento orientado a objetos, porém com uma estrutura mais tolerante a erros do programador e mais eficiente (no critério tempo de implementação) para o programador.

3.2. CLASSES IMPLEMENTADAS

O código fonte original se encontra no arquivo Pasta.py, contendo as seguintes classes:

- Pasta: cria e administra objetos que representam diretórios (manipulados livremente pelo jogador).
- File: cria e administra objetos que representam arquivos (manipulados livremente pelo jogador).
- Environment: administra o estado do jogo (com relação à disposição dos arquivos, diretórios e informações miscelâneas sobre o jogador e a narrativa) dependendo do nível (maior parte da funcionalidade do jogo reside nas funções desta).

4. CONCLUSÃO

O andamento e resultado do projeto permaneceram dentro dos parâmetros previstos. Além de consolidar conceitos imprescindíveis de programação orientada a objetos expostas na disciplina, adquirimos inevitavelmente habilidades de extrema importância no mercado de trabalho, como produtividade em equipe e experiência com novas linguagens e frameworks de programação.

Ao longo do desenvolvimento surgiram boas ideias que teriam espaço numa possível futura versão do Linux Quest (por conta de limitações de tempo). Entre elas a implementação de bifurcações na narrativa, em oposição a um esquema essencialmente linear; ou ainda um aperfeiçoamento do ambiente simulado de terminal para suportar uma gama mais ampla de comandos genéricos.

As contribuições de cada participante foram fundamentais para o término do trabalho de forma ideal e dentro do prazo.

5. REFERÊNCIAS

- [1] <https://docs.python.org/3/>
- [2] <https://jupyter.org/documentation>
- [3] <https://www.gnu.org/software/bash/manual/bash.html>
- [4] <https://www.emwelsh.com/blog/write-good-video-game-story>