Relatório Trabalho Prático III

Christoffer de Paula Oliveira, Paulo Henrique Tobias, Millas Násser

UFSJ- Universidade Federal de São João Del Rei

Sumário

Introdução

Este documento tem como objetivo relatar o que ajudou e o que atrapalhou na implementação das novas funcionalidades requeridas no Trabalho Prático III assim como a percepção do grupo sobre a evolução do código do Trabalho Prático II.

1 Corredores

A princípio, para termos um código mais limpo e sem repetições, foi criada a classe *Local*. Salas, assim como corredores, estendem de Local. Com isso, as classes *Corredor* e *Sala* ficaram com implementações de apenas suas diferenças, como por exemplo o tamanho das salas, informações de objetos dentro da sala e troll das cavernas. Estas são funcionalidades que os corredores não necessitam utilizar.

Houve também uma utilidade dessa classe para adicionar os novos trolls pois há trolls guerreiros em ambas as classes. Para termos uma organização melhor antes de adaptar os corredores, as portas passaram ser únicas, isto é, cada porta tem sua Sala e seu Corredor. Apenas uma porta liga dois Locais, ao contrário do Trabalho prático II que tínhamos duas portas para essa tarefa.

Por último, a classe *Mapa* passou a ter uma lista de Corredores.

1.1 Nossa perspetiva sobre essa evolução

Esta foi provavelmente a mudança mais drástica feita, pois algumas classes tiveram que ser modificadas para acomodar esta mudança. Entretanto, devido ao código bem modularizado, mesmo estas alterações foram feitas de maneira rápida e não afetaram o funcionamento do resto do programa.

2 Trolls

A antiga classe *Troll* passou a ser uma super classe para os dois novos tipos de troll. A partir disso, assim como nas classes que estendem local, as classes *TrollGuerreiro* e *TrollCaverna* ficaram com implementação de apenas suas diferenças. Os Trolls Guerreiros ficam responsáveis apenas por atacar o jogador, enquanto os trolls das cavernas ficam responsáveis por proteger os amontoados de ouro e diamantes.

2.1 Nossa perspetiva sobre essa evolução

Devido ao conceito de herança, houve poucas modificações no código, já que as novas classes se comportam como a antiga. Os trolls ainda possuem o mesmo comportamento, porém, este foi dividido para os Trolls da Caverna e os Trolls Guerreiros. Nenhuma funcionalidade realmente nova teve que ser feita.

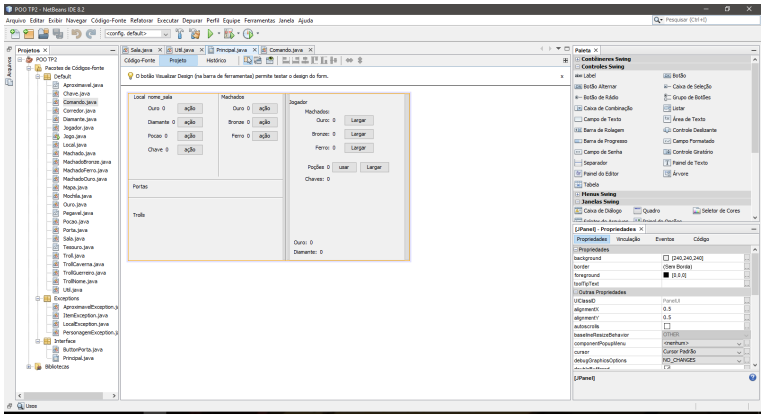

Figura 1. NetBeans GUI Builder

3 Interface Gráfica

Esta foi a parte mais problemática da nova versão. Grande parte da dificuldade se deu pelo fato de que nenhum dos integrantes do grupo tinha algum tipo de experiência em fazer programas gráficos. Adaptar-se aos conceitos do *Swing* não foi tarefa fácil.

Com este obstáculo em mente, optamos por utilizar a ferramenta de criação de interfaces gráficas do *NetBeans*.

O próximo passo foi implementar e conectar os eventos da interface com o jogo. Para isso, transformamos a antiga classe *Console* em *Comando* e fizemos algumas pequenas alterações nela. Assim, quando um botão é clicado, ele envia um sinal em formato de *String* para a função *comando*, que processa as ações do jogador da mesma maneira que a versão do segundo trabalho.

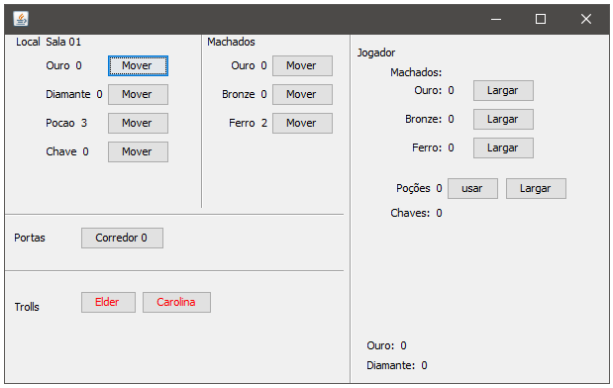

Figura 2. Interface Gráfica

4 Conclusão

Este trabalho ajudou a reforçar a importância dos conceitos de POO. Melhorar um código que estava bem modularizado se mostrou uma tarefa muito mais fácil do que melhorar o código do primeiro trabalho, em que a maior parte do tempo foi gasta consertando erros e reescrevendo e implementando funções. Neste, podemos começar diretamente implementando as novas funcionalidades sem perder tempo corrigindo erros passados.

Uma rápida evolução do projeto, indica que o projeto feito anteriormente estava seguindo os conceitos básicos de forma ideal, que foram continuados por este trabalho.