

Universidade Federal de Lavras
Departamento de Ciência da Computação
GCC178 – Práticas de Programação Orientada a Objetos

Trabalho Prático

Objetivo do Trabalho

O principal objetivo deste trabalho é praticar os conceitos aprendidos na disciplina. Espera-se com isso que os alunos possam revisar os conteúdos vistos em sala e entendê-los melhor.

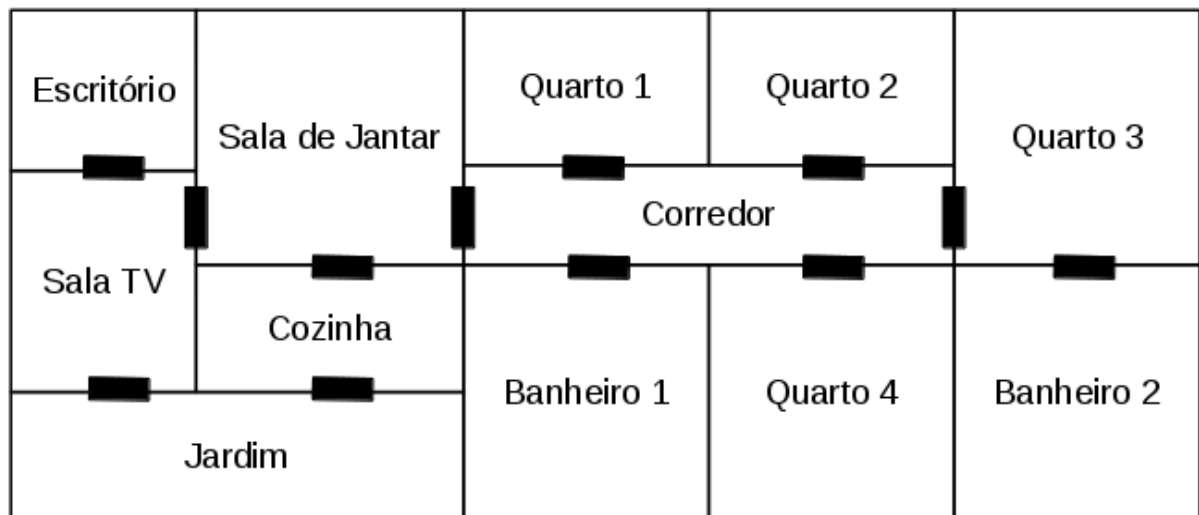
Um segundo objetivo é manter os alunos praticando a programação orientada a objetos, pois o aprendizado da disciplina só acontece ao colocar em prática os conceitos vistos em sala de aula.

Proposta do Trabalho

O trabalho a ser desenvolvido corresponde a um jogo implementado na linguagem Java. Ele deve ser um jogo estilo *Adventure* (texto) baseado na implementação do jogo *World of Zuul* apresentado na aula sobre “Design de Classes” e discutido no livro do Barnes e Kolling (da bibliografia da disciplina).

Nesse jogo, o jogador é um caçador de tesouro em uma casa abandonada e mal assombrada. O jogador deve se movimentar entre os diferentes ambientes da casa por meio da abertura de portas em busca de um tesouro que está enterrado embaixo de algum dos ambientes da mesma. Ao chegar no ambiente que o caçador julga estar o tesouro, ele deve usar a única carga explosiva que carrega em sua mochila para abrir um buraco no solo de modo a recuperar o tesouro. O jogador vence o jogo somente se conseguir recuperar o tesouro.

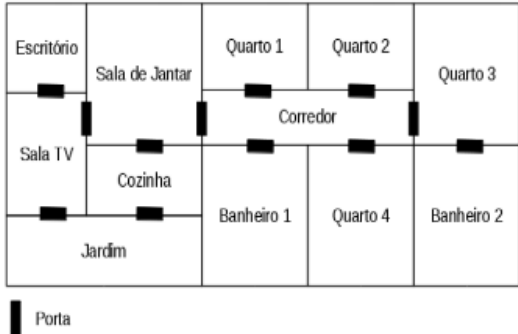
O mapa da casa é mostrado na Figura 1. Cada ambiente, além das portas para os demais ambientes, pode conter itens (tesouro, chave mestra e dica). Nesse jogo, há um tesouro enterrado no ambiente **A**, três dicas com o texto “O tesouro não está no(a) **X**”, uma dica com o texto “O tesouro está próximo ao(à) **Y**” e uma chave mestra que possui uma durabilidade **Z** (número de portas que ela pode abrir). Como a casa é mal assombrada, às vezes, algumas portas não abrem quando o jogador tenta passar pelas mesmas. Os valores de **X's**, **Y** e **Z** (inteiro entre 1 e 12) devem ser definidos aleatoriamente no início do jogo.



A localização do tesouro (**A**), das dicas e da chave mestra devem ser aleatoriamente definidos no início do jogo e armazenados em um arquivo texto. Além disso, cada porta da casa pode estar no estado “funcionando corretamente” ou “emperrada”, sendo que esse estado deve ser aleatoriamente redefinido a cada vez que o caçador tenta abrir uma porta. Ou seja, ao iniciar o jogo e após cada tentativa do caçador de abrir uma porta, o estado das portas da casa deve ser aleatoriamente redefinido. A quantidade de tentativas de abertura de portas define a duração máxima do jogo. Esse valor deve ser um número inteiro entre 20 e 50 aleatoriamente definido no início do jogo e apresentado para o jogador. Ao longo do jogo, a única situação que não altera o número de tentativas de abertura de portas realizadas pelo jogador é quando o mesmo faz uso da chave mestra. Nesse caso, apenas decrementa-se uma unidade na durabilidade da chave mestra e a porta será aberta independentemente do estado da mesma. Se o caçador não recuperar o tesouro antes de acabarem suas tentativas de abertura de porta ou se utilizar sua carga explosiva no ambiente errado (onde não existe tesouro), o jogador perde o jogo e o mesmo termina.

Além disso o trabalho deve atender os seguintes requisitos:

- A versão do *World of Zull* disponibilizada no Campus Virtual possui diversas falhas relacionadas ao Design de Classes. O trabalho deverá estar com todas essas falhas corrigidas.
- Você deve considerar que no futuro alguém poderá desejar alterar as passagens entre os ambientes incluindo novas portas.
- O jogador sempre começa o jogo na sala de TV.
- A geração da dica com o texto “O tesouro está próximo ao(à) **Y**” deve ser realizada escolhendo-se aleatoriamente um ambiente (vizinho) que tenha uma porta de passagem para o ambiente onde o tesouro se localiza.
- O jogo deverá ter uma interface gráfica como a apresentada na Figura 2.

<p>Número de Tentativas Restantes:</p> <p>35</p> <p>Durabilidade da chave mestra:</p> <p>8</p>		<p>Dicas encontradas:</p> <p>O tesouro não está no escritório.</p> <p>O tesouro não está no quarto 1.</p>
<p>Bem-vindo ao Caça ao Tesouro! Caça ao Tesouro é um novo jogo de aventura, incrivelmente bacana. Digite 'ajuda' se <u>voce</u> precisar de ajuda.</p> <p><u>Voce</u> esta na Sala de TV da casa mal assombrada. <u>Saídas</u>: Escritório, Sala de Jantar, Jardim ></p>		

Requisitos Não-Funcionais

- O trabalho deverá usar corretamente os conceitos de Orientação a Objetos.
- O trabalho deverá fazer uso de composição e/ou agregação.
- O trabalho deverá fazer uso de herança.
- O trabalho deverá fazer uso de polimorfismo (variável polimórfica e polimorfismo de método).
- O trabalho deverá ter um bom *Design* de Classes.
- O código deverá estar comentado para permitir a geração da documentação Javadoc do mesmo.
- Deve ser entregue junto com o trabalho o Diagrama de Classes UML da modelagem do sistema.
- Serão avaliados a legibilidade do código, organização e uso de comentários.

Pontuação e Entrega

Conforme previsto no Plano de Curso, este trabalho vale 25% da nota do período.

Para a entrega final do trabalho deve haver uma pasta raiz contendo todos os códigos do trabalho e um arquivo **PDF** com o diagrama de classes UML. Essa pasta deve ter o nome “*TrabalhoPratico_Grupo@_Turma#A*”, onde @ corresponde à identificação do grupo e # ao número da turma (10 ou 14). A pasta deve ser **compactada** (.zip) **em um único arquivo** com o mesmo nome da pasta (ex: “*TrabalhoPratico_Grupo@_Turma#A .zip*”) e esse arquivo deve ser enviado **até dia 05/12/2019, às 23h50**, via atividade que será aberta para essa finalidade no **Campus Virtual**.

Os alunos podem escolher o ambiente de desenvolvimento (IDE) que preferirem e não podem utilizar qualquer biblioteca ou API além do padrão Java (versão 8 ou superior).