

Universidade Federal de Lavras

Departamento de Computação Aplicada

Trabalho Prático

O trabalho a ser desenvolvido corresponde a um jogo implementado na linguagem Java. Ele deve ser um jogo estilo *Adventure* baseado na implementação do jogo *World of Zuul* a ser apresentado na aula sobre “Design de Classes: Coesão e Acoplamento” e discutido no livro do Barnes e Kolling (da bibliografia da disciplina). Desse modo, **a sua implementação deverá ser realizada a partir do código inicial do jogo *World of Zuul* fornecido pelo professor** no Campus Virtual, ou seja, o jogo proposto pelo seu grupo deve ter como estrutura base **um jogador que se movimenta por diferentes ambientes** a partir das direções escolhidas pelo mesmo em cada momento. Vale observar que a versão inicial do jogo disponibilizada no Campus Virtual possui diversas falhas relacionadas ao design de classes, como por exemplo, alto acoplamento, baixa coesão de métodos, duplicação de código e outras. Sendo assim, a primeira tarefa antes de começarem a utilizar o código fornecido pelo professor é fazer todas as alterações apresentadas no Capítulo 7 (4ª edição) do livro do Barnes e Kolling, a saber:

- Remover duplicação de código.
- Remover atributos públicos.
- Fazer uso da classe `HashMap` na classe `Ambiente`.
- Criar método `getDescricaoLonga` para melhorar o design baseado em responsabilidade.
- Remover o acoplamento implícito e criar o comando observar no jogo.

Somente após a correção do código fornecido o grupo deve começar as alterações no mesmo para que ele possa implementar o jogo que está sendo proposto.

A implementação inicial fornecida do jogo permite apenas que o jogador possa ficar trocando de ambiente. Ou seja, você terá que expandir esse código para criar um jogo que seja realmente interessante e tenha um objetivo bem definido. Para tornar o seu jogo mais interessante e desafiador, você pode fazer alterações tais como:

- Permitir que os ambientes contenham itens que possam ser usados pelo jogador e/ou entidades que interajam com o jogador.
- Incluir novos comandos no jogo, como por exemplo, **pegar** item, **usar** item, **largar** item, **destruir** fantasma etc.
- Fazer com que as saídas dos ambientes possam ficar temporariamente bloqueadas ao longo do jogo e/ou que o jogador precise atender alguma condição para poder acessar algumas saídas.

Cada grupo deverá implementar uma proposta diferente de expansão desse jogo. A proposta do jogo a ser implementado por cada grupo deverá ser documentada e entregue pelo Campus Virtual. Nessa proposta, cada grupo deverá definir o objetivo do jogo, o mapa do mesmo e todos os demais detalhes que possibilitem o entendimento sobre como o jogo funcionará. O **Anexo I** deste documento apresenta **um exemplo desse documento de proposta** que deve ser produzido por cada grupo.

Requisitos Adicionais

- 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 (variáveis polimórficas e polimorfismo de método).
- O trabalho deverá ter uma interface gráfica como a apresentada na Figura 2 do Anexo I.
- O trabalho deverá ter um bom *Design* de Classes.
- Deve ser entregue junto com o trabalho o diagrama de classes UML do mesmo.
- Todo o código deve fazer uso de comentários (padrão Javadoc) e os comentários devem estar em português.
- Também serão avaliados a legibilidade do código, organização e uso de comentários.

Pontuação e Entrega

Conforme previsto no Plano de Curso, este trabalho vale 24% da nota do curso.

Cada grupo deverá entregar um documento (*PropostaInicialTrabalhoPratico_Grupo@.pdf* – onde @ é o número do grupo no Campus Virtual) que definirá a proposta do jogo a ser implementado. Esse documento deve ser entregue via Campus Virtual **até 23h50 do dia 28/07/2022**.

Para a entrega final do trabalho deve haver uma pasta raiz contendo todos os códigos do trabalho (**somente arquivos .java**) e um arquivo **PDF** com o diagrama de classes UML. Essa pasta deve ter o nome *TrabalhoPratico_Grupo@*, onde @ corresponde à identificação (número) do grupo no Campus Virtual. A pasta deve ser **compactada (.zip)** **em um único arquivo** com o mesmo nome da pasta (ex: *TrabalhoPratico_Grupo@.zip*) e esse arquivo deve ser enviado pelo Campus Virtual **até 23h50 do dia 01/09/2022**.

Os alunos poderão utilizar qualquer editor e/ou IDE para o desenvolvimento do trabalho prático e não podem utilizar qualquer biblioteca ou API além do padrão Java (versão 8 ou superior). Além disso, **a implementação deverá compilar e executar corretamente a partir de um terminal**.

ATENÇÃO: O grupo que não entregar o trabalho prático pelo Campus Virtual (a proposta inicial até 28/07/2022 e a implementação com diagrama de classe até 01/09/2022) ficará com nota zero e não participará da etapa de entrevista.

ANEXO I - Exemplo de Proposta de Trabalho

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. O jogador sempre começa o jogo na sala de TV.

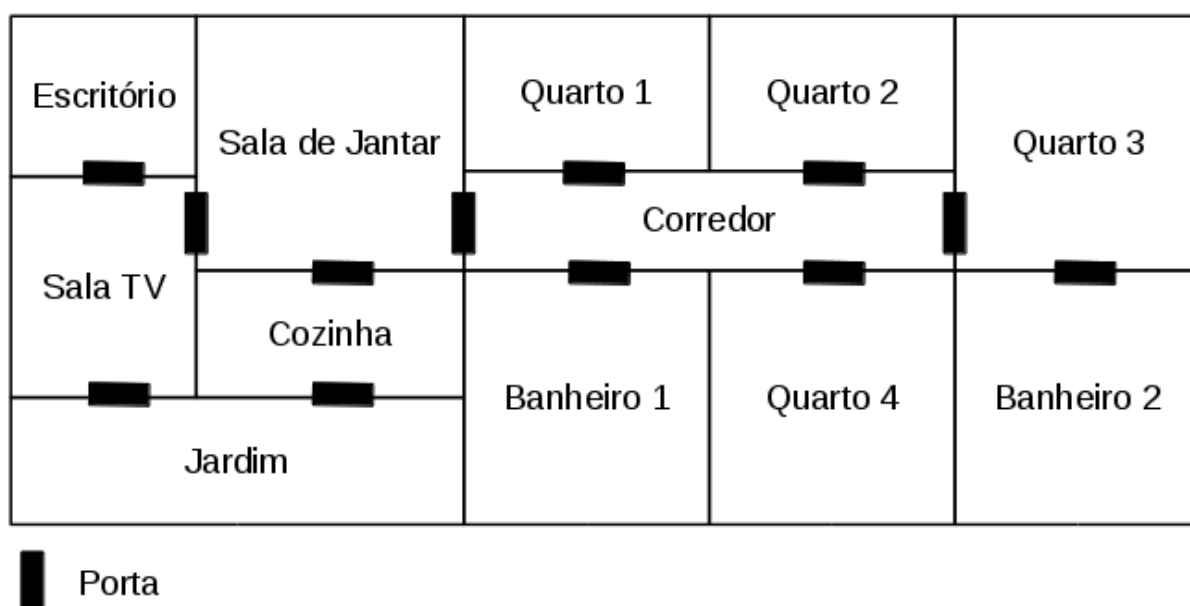


Figura 1 - Mapa do jogo

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.

Por fim, o jogo terá a interface gráfica apresentada na Figura 2.

Número de Tentativas Restantes: 35 Durabilidade da chave mestra: 8	<p>Porta</p>	Dicas encontradas: O tesouro não está no escritório. O tesouro não está no quarto 1.
<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>		

Figura 2 - Interface gráfica de usuário