



Trabalho Final

O objetivo deste trabalho é exercitar os conceitos de Programação Orientada a Objetos, considerando abstração, encapsulamento, herança e polimorfismo. **A temática do trabalho será livre**, ou seja, será permitido que os grupos apresentem propostas de trabalho que deverão ser aprovadas pelo professor. Além disso, serão sugeridos temas pré-aprovados, como por exemplo, o jogo da explosão ou um jogo RPG.

Os **grupos** que optarem pelas **propostas de temas pré-aprovados**, devem **seguir os requisitos** da especificação referente a **temática** escolhida, considerando os **anexos**.

Para os **grupos** que **optarem** por **propor um tema**, será **necessário** me **encaminhar** por **e-mail** um **diagrama de classes inicial** e **uma breve descrição** do que vocês desejam implementar. Se necessário, podemos marcar uma reunião na quarta (16/11) ou quinta (17/11) para tratar disso pessoalmente ou por videoconferência. **Para a apresentação final, SERÃO CONSIDERADOS APENAS AS PROPOSTAS QUE FOREM APROVADAS PREVIAMENTE, até o dia 18/11.**

Dos Requisitos Gerais

1- Implementação das classes e funcionalidades da especificação. (Total de 9 pontos para grupos / Total de 10 pontos para implementação individual)

- No caso das propostas pré-aprovadas: implementar a totalidade de classes e funcionalidades previstas na especificação.
- Demais propostas: implementar a totalidade dos requisitos aprovados pelo professor.
- O uso de arquivos/serialização NÃO é obrigatório.
- Critérios de avaliação:
 - (3 pontos) Habilidade em estruturar programas pela decomposição da tarefa em subtarefas, utilizando subprogramação para implementá-las.
 - (3 - 3,5 pontos) Emprego dos conceitos de POO: abstração, encapsulamento, herança e polimorfismo.
 - (3 - 3,5 pontos) Atendimento aos requisitos mínimos da especificação do problema.

2- Implementação de interface gráfica. (+1,5 pontos - opcional)

- Além do material disponível no e-aula, a demonstração de uso de interface gráfica será feita nas aulas do dia 22/11. O exemplo utilizado será o do jogo RPG.



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos - 2022/1
Prof.: Felipe de Souza Marques (felipem@inf.ufpel.edu.br)

3- Implementação Multiplayer com API fornecida pelo professor. (+1,5 pontos - opcional)

- A documentação da API, bem como o código fonte estará disponível no e-aula.

Observação: pode-se perceber que a soma dos pontos de cada requisito pode chegar a 13 pontos. Os pontos excedentes serão somados à nota da primeira prova, que se limita ao máximo de 10 pontos.

Dos Grupos

O trabalho poderá ser feito de forma **individual** ou em **DUPLAS**. **Serão aceitos** trabalhos em **TRIOS**, desde que o(s) requisito(s) 2 e/ou 3 sejam implementados.

Da Entrega do Trabalho

O trabalho deve ser entregue via e-aula até o dia 01/12/2022 às 12h. Os trabalhos devem ser apresentados para o professor até às 18:00 do dia 02/12/2022, com agendamento prévio.

Um representante do grupo deve submeter os seguintes arquivos para completar a entrega do trabalho:

- Diagrama de Classes: um arquivo pdf contendo o diagrama de classes final do projeto;
- Readme: Arquivo texto contendo as instruções básicas para o uso correto do programa enviado;
- Arquivos fonte: para simplificar, pode-se compactar o diretório do projeto em um arquivo ZIP.

Observação: este conjunto de arquivos deve ser compactado no formato ZIP em um único arquivo enviado para entrega via e-aula.

Código de Honra

O trabalho deve ser implementado na sua totalidade, sem uso de código de outros (colegas ou não). O trabalho enviado deve representar um esforço honesto em resolver o problema - isto é, não é algo "pela metade", que não implementa funcionalidades essenciais. Violações a esta conduta serão penalizadas e o violador não só terá nota nula neste trabalho, mas também não terá direito a enviar os próximos trabalhos, ficando portanto sem parte da nota da disciplina. Os plágios estão sujeitos a sanções administrativas pelo colegiado do curso.



Anexo I - Especificação do Jogo da Explosão

A especificação deste jogo é inspirada no jogo Explosão da Estrela (para referência: https://www.youtube.com/watch?v=w_X-K2ISHQM).

Considerando os requisitos gerais, **não é obrigatório** a implementação de interface gráfica e/ou uma versão **multiplayer**, para jogar em rede.

Requisitos desta especificação

1. Deverão ser implementadas as regras do jogo original, que estão disponíveis em <https://estrela.vteximg.com.br/arquivos/Manual-Jogo-Explosao.pdf>. Deve-se considerar que o jogo poderá ter até quatro jogadores e cada jogador terá um turno para realizar sua jogada. A mesma dinâmica pode ser empregada para a versão multiplayer.
2. Adicionalmente, nossa versão do jogo contará com a adição de *power ups*, que podem beneficiar o jogador que os obtêm, seja para conferir uma habilidade adicional ou para atrapalhar os adversários. Os *power ups*, modelados pela classe *Item* são divididos em duas categorias: passivos e instantâneos.

Itens passivos: os itens passivos servem para proteger os jogadores de algum infortúnio ou atrapalhar adversários quando eles cruzam o caminho do jogador. Considera-se que eles se tornam ativos no momento em que o jogador os adquire. Assim que ele é utilizado, ele deve ser excluído do inventário do jogador.

- **Asas de Ícaro:** ao explodir a ponte em que o jogador está, as asas são ativadas. Elas permitem que o jogador voar até a próxima torre.
- **Escudo:** o escudo previne qualquer tipo de infortúnio causado por um item ao jogador que o detém.
- **Empurrar:** empurra o adversário que estiver em uma casa (pedaço da ponte) que o jogador passar. Caso haja mais de um adversário na mesma casa, o jogador escolherá qual será empurrado.

Itens instantâneos: os itens instantâneos podem ser utilizados para atingir algum adversário ou para um benefício temporário. Eles só podem ser utilizados durante o turno do jogador que o possui. Uma vez utilizado, ele deve ser excluído do inventário.

- **Imobilizar:** um jogador adversário fica uma rodada sem jogar.
- **Velocidade:** duplica o valor tirado no dado.
- **Sabotar:** sabota a explosão, ou seja, evita que um adversário acione o mecanismo de explosão.



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos - 2022/1
Prof.: Felipe de Souza Marques (felipem@inf.ufpel.edu.br)

3. Deve ser considerado que o dado do jogo tem 6 faces, sendo três com os números de 1 a 3, indicando quantas casas o jogador deve andar, duas bombas indicando que o mecanismo de explosão deve ser acionado e uma com um ponto de interrogação. Ao tirar o ponto de interrogação, o jogador receberá um item aleatoriamente, dentre as 6 possibilidades. O jogador pode carregar apenas um item passivo e um instantâneo. Caso ele já esteja carregando algum item o mesmo será substituído pelo último item escolhido aleatoriamente, considerando o seu respectivo tipo.
4. Todas as classes do diagrama da Figura 1 devem ser implementadas de acordo com o que foi especificado, considerando as relações entre as classes, atributos e métodos. O diagrama pode estar implementado de forma parcial. Implemente os métodos que achar necessário para contemplar todas as funcionalidades do jogo.



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos - 2022/1
Prof.: Felipe de Souza Marques (felipem@inf.ufpel.edu.br)

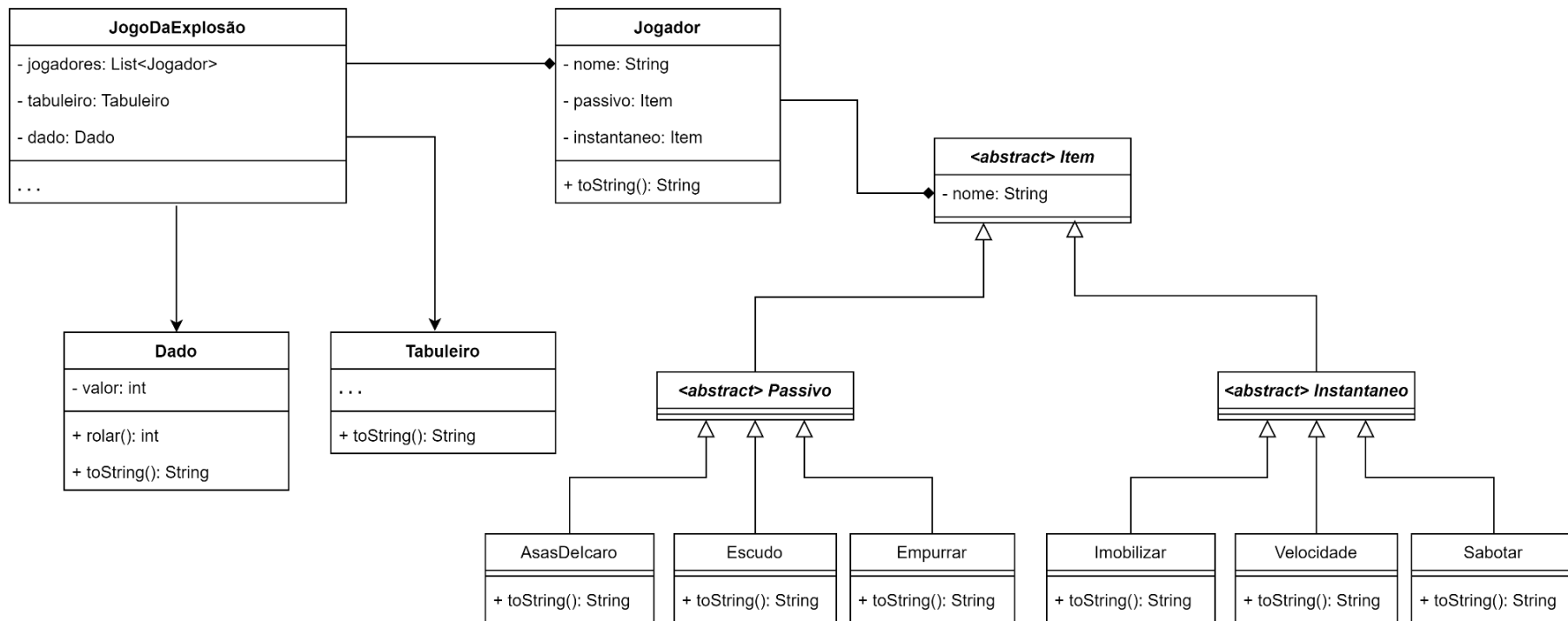


Figura 1 - Diagrama de Classes do Jogo Explosão



Dicas para a modelagem

1. Inicie a implementação considerando as regras originais. O projeto pode ser feito de forma incremental. Após realizar a implementação da versão original, pode-se introduzir a versão com os itens de *power up*.
2. O tabuleiro pode ser visto como uma matriz, de acordo com a Figura 2. Isso facilita o controle para implementar a dinâmica do jogo e a construção de uma interface gráfica. Cada célula da matriz pode ser implementada por um *JPanel*, que mostra uma imagem, correspondente ao cenário do jogo. As torres ficam nos cantos do tabuleiro. As células em amarelo representam as casas em que as pontes podem ser construídas. A parte em cinza representa os demais elementos do jogo, que servem apenas para acrescentar elementos visuais no tabuleiro.

Torre 1					Torre 2
Torre 4					Torre 3

Figura 2 - Representação do tabuleiro do jogo.



Anexo II - Especificação do Jogo RPG

Introdução

A ideia do jogo World-of-Zuul é simples e o projeto inicial está disponível no e-aula. Até o momento temos a possibilidade de criar um mapa de salas, inserindo itens e personagens em cada uma delas. A partir deste cenário inicial, um personagem herói pode se deslocar no mapa para coletar itens valiosos e combater inimigos.

Agora é o momento de você exercitar sua criatividade e criar a sua história para o jogo! Com base em um conjunto de regras mínimas pré-estabelecidas, você poderá criar a sua história e aumentar o conjunto de regras, incluindo novos procedimentos, novos itens, novos tipos de personagem, criar túneis secretos de uma sala para outra mais distante, etc.

Requisitos Mínimos

O mapa do jogo deve conter pelo menos 9 salas. Algumas destas salas podem estar completamente vazias, servindo apenas de corredor de acesso a outras salas. As demais, devem conter itens e/ou inimigos de acordo com a história do jogo.

Ao menos um personagem principal deve existir: o nosso herói. Além dos atributos já existentes no projeto desenvolvido em aula, ele deve considerar um atributo para guardar um número de “moedas” (que pode ser um valor inteiro). Cada conjunto de 1000 moedas ocupa uma unidade de peso no inventário do herói. Valores intermediários serão sempre arredondados para o próximo valor inteiro, ou seja, se o herói estiver carregando 1200 moedas, elas estarão ocupando 2 unidades de peso no inventário do herói. Também deve ser permitido que o herói se equipe com um ou mais equipamentos que garantam características especiais. Por exemplo, armas que aumentam o dano causado em combate, ou ainda, um escudo e/ou armadura que previne dano em combate.

Deve haver no mínimo dois tipos de inimigos: comuns e chefes. Todos os inimigos devem ter um atributo que informa a quantidade de moedas que ele carrega. Ao eliminar um inimigo, ele deve ser removido da sala e as moedas que ele carregava devem ser adicionadas ao inventário de itens da sala. O herói poderá coletar as moedas do inimigo derrotado a partir do mapa de itens da sala. Os inimigos do tipo chefe podem carregar um ou mais itens que concedem habilidades a quem os carrega/equipa. Por exemplo, um item que aumenta o dano causado de um ponto de energia para dois, ou ainda, um item que aumenta a energia em X pontos (existem várias outras possibilidades). Ao eliminar o inimigo chefe, todos os itens devem ser movidos para o inventário da sala. O herói pode se apossar dos itens a partir do mapa de itens da sala.



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos - 2022/1
Prof.: Felipe de Souza Marques (felipem@inf.ufpel.edu.br)

Alterar o sistema de luta implementado durante as aulas práticas, de modo que se considere pontos de danos e multiplicadores a partir de itens que os personagens carregam. Para um sistema mais sofisticado é interessante considerar que os personagens possuem pontos de ataque e defesa. Toda vez que se profere um ataque, existe uma probabilidade de se acertar o golpe. Exemplos podem ser encontrados em: <https://centrorpg.com/index.php?topic=13160.0> ou <http://rpg-swordartonlinepgrpg.weebly.com/sistema-de-danodefesa.html>.

Deve-se prever itens de diferentes tipos:

- **Instantâneo:** por exemplo, um alimento contém uma determinada quantidade de energia que, quando comidos pelo herói, converte a energia do alimento em energia extra para o herói (observe o limite de energia do herói). Este tipo de item pode ser carregado pelo herói (no inventário) e utilizado quando for necessário. O exemplo do alimento é um dos itens obrigatórios que devem constar no jogo.
- **Permanentes:** por exemplo, enquanto o herói carregar um item do tipo permanente, ele aumenta um determinado atributo do herói: energia máxima, carga de peso máximo ou quantidade de dano causado em cada ataque realizado. Neste caso, esta seria uma habilidade passiva. Armas, escudos, elmos, anéis, etc são considerados itens permanentes. Itens permanentes devem ter um atributo de durabilidade. Toda vez que o item é utilizado ou o herói sofre algum dano, os itens que estão equipados também sofrem perda de pontos de durabilidade. Estes pontos poderão ser restaurados a partir do uso de outros itens ou outros personagens com habilidades para consertar itens.

Seja criativo!!! A partir destes requisitos, proponha o seu jogo! Para aqueles que quiserem se aventurar em uma interface gráfica mais elaborada (isso não é um requisito mínimo), recomendo o uso do framework slick2D (<http://slick.ninjacave.com>). Ele permite a criação de mapas, uso de imagens para personagens, sons etc. Tem vários tutoriais disponíveis na internet. Outro framework bastante interessante e mais completo é o LibGDX - <https://libgdx.badlogicgames.com/>.