



Relatório do Trabalho de PEOO

Ciência da Computação – 2º Semestre

João Pedro de Moura Góes – 1538631, Francisco Alexandro Araújo Souza e Filho – 1538708, Gustavo Ângelo Cavalcante da Costa – 1538722

joao.goes@aluno.uece.br, gustavo.angelo@aluno.uece.br,
francisco.alexandro@aluno.uece.br

Resumo. O presente relatório refere-se ao trabalho final da matéria de Programação Estruturada e Orientada a Objeto. Este trabalho consiste na criação de um programa que simule um jogo de gênero battle royale com elementos de RPG usando da Programação Orientada a Objeto aprendida em aula.

Desenvolvimento

Funcionalidades Básicas

O programa começa dispondo um menu que utiliza da estrutura Switch Case para registrar a escolha do usuário, há também um bloco Try/Catch no caso do valor digitado pelo participante não seja um número inteiro.

Classe Personagem

Com a necessidade da criação de 3 tipos de personagem, os quais possuirão vantagens e desvantagens próprias junto com duas habilidades especiais, surgiu a ideia da criação da Classe Abstrata Personagem, que seria a classe a qual os 3 tipos de personagem (Guerreiro, Arqueiro e Feiticeiro) herdariam. A escolha de fazer a Classe Personagem como classe abstrata, pois não seria necessário instanciar a classe Personagem, ao invés dela, suas classes filhas são as quais serão instanciadas.

Classe Guerreiro

Atributos Base:

Vida: 56

Escudo: 32

Ataque: 30

Defesa: 6

Mana: 0

PowerUP: 20%

Este tipo de personagem possui atributos base de vida e escudo, elevados se comparado aos tipos Arqueiro e Feiticeiro.

Suas duas habilidades especiais são:

+ Melhorar Shield: Esta habilidade dá um aumento de 10 pontos no atributo Escudo do personagem podendo ser acumulada.

+ Custo de Mana: Nulo

+ DUAL-BLADES - ELUCIDATOR: O Guerreiro usa da mana para invocar uma segunda espada mágica e causar um dano baseado no dobro de seu ataque no alvo.

+ Custo de Mana: 50

Classe Arqueiro

Atributos Base:

Vida: 34

Escudo: 22

Ataque: 32

Defesa: 8

Mana: 0

PowerUP: 30%

Este tipo de personagem possui um atributo base de ataque elevado se comparado aos tipos Guerreiro e Feiticeiro.

Suas duas habilidades especiais são:

+ Aumentar Vida: Esta habilidade dá um aumento de 15 pontos no atributo Vida do personagem podendo ser acumulada.

+ Custo de Mana: nulo

+ Turbilhão de Flechas Celestiais: Esta habilidade faz com que chuvas centenas de flechas mágicas no alvo causando o dobro do atributo de ataque em dano.

+ Custo de Mana: 50

Classe Feiticeiro

Atributos Base:

Vida: 36

Escudo: 26

Ataque: 26

Defesa: 10

Mana: 50

PowerUP: 50%

Este tipo de personagem possui atributos base de mana, defesa e power-up, elevados se comparado aos tipos Guerreiro e Arqueiro.

Suas duas habilidades especiais são:

+ Aumentar Mana: Esta habilidade dá um aumento de 10 pontos no atributo Escudo do personagem podendo ser acumulada.

+ Custo de Mana: nulo

+ BAKURETSU MAHOU – EXPLOOOOSION: Esta habilidade causa uma explosão no alvo, essa explosão causa o dobro do dano de ataque atual do Feiticeiro.

+ Custo de Mana: 50

Menu

1 - Iniciar Partida

Para que uma partida tenha início, é necessário que existam no mínimo dois ou mais personagens, e que, dos existentes, no mínimo dois sejam selecionados.

Quando uma partida tem início, é definida aleatoriamente uma ordem para os turnos, essa ordem permanece até o fim do jogo. Com o objetivo de definir a ordem aleatoriamente importamos o `java.util.Random` e o utilizamos para criar uma ordem aleatória na qual os competidores agirão.

Para que uma partida acabe é necessário que sobre apenas um competidor vivo no estádio, o ganhador então ganhará uma benção que restaurará seus atributos atuais a seus valores base e perderá todos os seus itens no inventário.

2 - Criar Personagens

Ao criar um personagem é necessário primeiro escolher seu tipo, Guerreiro, Arqueiro ou Feiticeiro, e então definir um nome para esse personagem. Para criar o personagem se utiliza um bloco Switch/Case que de acordo com o input do usuário irá definir o tipo de classe que será instanciada (Guerreiro, Arqueiro e Feiticeiro). O construtor recebe apenas o nome e o objeto criado vai para um ArrayList do tipo Personagem.

3 - Editar Personagens

Edita o nome de um personagem escolhido entre os competidores disponíveis, se o nome já existir dentre os competidores a ação não é realizada.

4 - Mostrar Competidores

Dispõe uma lista com os competidores atualmente presentes, os que estão salvos no arquivo `save.txt`, mas que não foram carregados ainda não são mostrados.

5 - Remover Competidores

Recebe o input do usuário para saber o nome do competidor a ser removido, então a função irá procurar por alguém com o mesmo nome, se não encontrar irá mostrar que não existe competidor com esse nome e retorna um ArrayList de competidores do tipo Personagem que essa função recebe como argumento. Caso contrário, ela irá procurar o index no ArrayList dos nomes dos competidores, então, utilizando esse index e um método do ArrayList chamado “remove” será removido o index/posição do competidor escolhido.

6 - Salvar Jogo

Salva os dados dos competidores atuais.

7 - Carregar Jogo Salvo

Carrega dados de competidores salvos previamente.

8 – Sair do Jogo

Finaliza o programa, dados salvos permanecem, mas dados não salvos desaparecem.

Funcionalidades nas Batalhas

Ações do Competidor

As ações que o competidor pode realizar são dispostas numa lista enumerada, onde o uso da estrutura Switch Case permite a escolha da ação tomada pelo competidor. Algumas ações gastam o turno ao serem realizadas, enquanto outras não.

1 – Atacar

Ataca um alvo, dano baseado no atributo ataque atual do competidor. Gasta 1 turno.

2 - Defender

Defende um ataque de qualquer inimigo, o dano do ataque é zerado. Gasta 1 turno.

3 - PowerUP

Aumento percentual (atributo powerUP), baseado no tipo do personagem. Essa função a cada rodada irá aumentar uma determinada porcentagem baseada no atributo powerUP o atributo ataque, possibilitando que mais dano seja causado no próximo ataque. O valor do atributo voltará ao seu valor base após o ataque carregado atingir um alvo. Gasta 1 turno.

4 - Habilidade Especial de Melhora

Habilidade especial de cada tipo, ela não gasta mana e tem a função de aumentar certo atributo do competidor, atributo esse sendo definido pelo seu tipo de personagem. Gasta 1 turno.

5 – Mostrar Inventário

Dispõe o inventário do competidor, nesse inventário ficam as poções. Não gasta turno.

6 - Usar item

Se o competidor possuir uma poção em seu inventário ele poderá usa-la, se ele usa-la, gasta 1 turno. Se ele não possuir nenhuma poção não gastará turno.

7 - Ver Status

Tem a função de mostrar o estado atual do competidor. Não gasta turno.

8 - Pular Vez

Se o competidor não quiser tomar nenhuma ação em seu turno ele pode pular sua vez.

9 - Ataque Especial

Habilidade especial, diferente para cada tipo de personagem, apenas desbloqueada se o competidor possuir 50 ou mais de mana. Se usada, o competidor perde 50 de mana e ela possui 50% de eficácia.

Crítico

A cada rodada múltipla de 3, o competidor nessa rodada ganhará um bônus, definido aleatoriamente entre 1 e 10, ao dano de seu ataque. Para calcular aleatoriamente o dano que será adicionado ao ataque foi importado o `java.util.Random` para conseguir um número entre 0.0 e 0.9, então multiplica por 10 e soma 1 para que ficasse um conjunto de números entre 1 e 10.

Sistema de Carregar e Salvar jogo

Para salvar o jogo uma classe `File` cria o arquivo `save.txt`, nele serão inseridos os dados dos competidores, nome e tipo, por meio da classe `FileWriter`. As classes `BufferedWriter` e `BufferedReader` foram utilizadas com o objetivo de otimizar a criação do arquivo `save.txt`.

Para carregar um jogo previamente salvo, utilizamos a classe `FileReader` para a leitura do arquivo seguido pela classe `BufferedReader` para otimizar a leitura do arquivo, usando duas variáveis “nome” e “classe” para ler respectivamente o nome e a classe de cada um dos competidores. Assim que a variável “classe” não encontrar valor a leitura do arquivo é finalizada.

Optamos pelas classes `BufferedWriter` e `BufferedReader` pois acreditamos que elas tenham conseguido atingir uma eficácia mais elevada de escrita e leitura do arquivo.

Sistema de Poções

Uma poção tem 40% de chance de surgir quando um inimigo é morto e se ela surgir irá automaticamente para o competidor responsável pelo abate. Existem poções de mana, de cura, de escudo e de força, cada poção tem uma chance de 80% de aumentar o atributo indicado por seu nome e 20% de chance de diminuir tal atributo.

A interface chamada poções foi criada, pois não seriam usados atributos nas classes que irão implementar essa interface e não precisaria de funcionalidades pré-definidas. Por isso decidimos utilizar uma interface como um guia para as classes das respectivas poções.

Conclusão

A realização deste trabalho foi de grande importância para os membros desta equipe, pois conseguimos aplicar em nosso projeto os conceitos de Orientação a Objeto vistos em aula.