



INTRODUÇÃO A PROGRAMAÇÃO – JOGOS DIGITAIS

PROVA 2 – 2023.1

DATA: 19/06/23

Professor: Murilo Boratto

Aluno(a):

INSTRUÇÕES GERAIS

I. Nesta prova você encontrará **7** (sete) páginas numeradas sequencialmente, contendo **1** (uma) questão valorada, correspondentes à disciplina: **INTRODUÇÃO À PROGRAMAÇÃO**.

II. A codificação desta prova consiste na extensão do código proposto no Laboratório 2, chamado UoU e apresentado em sala de aula.

III. A **questão 1** têm o caráter prático e deve ser implementado na linguagem C# e enviado por e-mail muriloboratto@uneb.br até a 14:00 horas do dia 21/06/2023 (quarta-feira), com o seguinte Assunto: Questão Prova 2 de IP – IP20231 <nome do(s) discente(s)>. Por exemplo:

Assunto: Questão Prova 2 de IP – IP20231 Murilo Boratto e Marivaldo da Silva

IV. Deve ser anexado ao email o código em um arquivo **ÚNICO** com extensão .cs com os primeiros nomes dos componentes. Por exemplo:

UoU-Murilo-Marivaldo.cs

Boa Prova!



INTRODUÇÃO A PROGRAMAÇÃO – JOGOS DIGITAIS

PROVA 2 – 2023.1

DATA: 19/06/23

Professor: Murilo Boratto

Aluno(a):

PARTE 1 (Valor: 2,0):

1. Realize uma **sobrecarga** do **método ReduzirVida** na **classe Heroi**, de forma que ele receba um valor de dano e subtraia esse valor dos pontos de vida do herói.
2. Adicione um novo **atributo** à **classe Herói** chamado **UTHeroi** (unidades de tempo). Esse atributo será utilizado para controlar a quantidade de ações queum herói pode realizar a cada ciclo de jogo.
3. Ajuste o **construtor da classe Heroi** para iniciar o **atributo UTHeroi** com valor 7.
4. Crie dois **métodos**, um para adicionar um valor ao **atributo UT (AumentarUT)** e. outro para subtrair um valor do **atributo UTHeroi (ReduzirUT)**.
5. Crie uma nova **classe chamada Magia**. Uma magia possui um **nome**, um **custo de unidades de tempo (CustoUT)** e um **valor de dano**.
6. Adicione um método à **classe Magia**, chamado **Lancar()**, que retorne o valor de dano causado.
7. Crie um **atributo** do tipo **Magia** na **classe Heroi**.
8. Atualize o construtor das **classes HeroiAlianca** e **HeroiHorda** para criar uma. nova magia para o herói de acordo com a tabela abaixo:

Classe	Magia	Custo de UT	Dano
HeroiAlianca	Força Rutilante	4 unidades	20 pts de vida
HeroiHorda	Caminho de Chamas	6 unidades	30 pts de vida



INTRODUÇÃO A PROGRAMAÇÃO – JOGOS DIGITAIS

PROVA 2 – 2023.1

DATA: 19/06/23

Professor: Murilo Boratto

Aluno(a):

9. Refatore o **método LancarMagia()** na classe **Heroi** para garantir os seguintes comportamentos:
 - a. Receber como parâmetro um objeto do tipo **Heroi** que será o alvo do ataque;

Lembre-se que **refatorar um código** em computação quer dizer: ajustar o código existente para que ele fique mais completo, mais seguro, mais rápido e/ou mais fácil de ser mantido.



UNEB
Universidade do
Estado da Bahia

INTRODUÇÃO A PROGRAMAÇÃO – JOGOS DIGITAIS

PROVA 2 – 2023.1

DATA: 19/06/23

Professor: Murilo Boratto

Aluno(a):

PARTE 2 (Valor: 2,0):

10. Crie uma nova **classe** chamada **Arma**. Uma arma possui um nome, um custo de unidades de tempo (**CustoUT**) e um valor de dano.
11. Adicione um método à **classe Arma**, chamado **Atacar()**, que retorne o valor de dano causado.
12. Crie um **atributo** do tipo **ArmaHeroi** na **classe Heroi**.
13. Atualize o **construtor** das **classes HeroiAlianca** e **HeroiHorda** para criar uma nova arma para o herói de acordo com a tabela abaixo:

Classe	Arma	Custo de UT	Dano
HeroiAlianca	Espada	2 unidades	10 pts de vida
HeroiHorda	Machado	4 unidades	20 pts de vida

14. Refatore o **método AtacarComArma()** na **classe Heroi** para garantir os seguintes comportamentos:
 - a. Receber como parâmetro um **objeto** do **tipo Heroi** que será o alvo do ataque;
 - b. Se o atacante tiver saldo suficiente para atacar com a arma, o **método Atacar()** do objeto arma deve ser invocado e o dano atribuído ao alvo. Caso contrário, nada deve ser feito;
 - c. Exibir uma mensagem informando o nome do alvo, o nome da arma, a quantidade de dano causado (zero, caso não tenha havido ataque) e a quantidade de vida restante do alvo.



INTRODUÇÃO A PROGRAMAÇÃO – JOGOS DIGITAIS

PROVA 2 – 2023.1

DATA: 19/06/23

Professor: Murilo Boratto

Aluno(a):

PARTE 3 (Valor:6,0):

Os exercícios seguintes serão todos realizados no Programa Principal (Main). A ideia básica é simular uma batalha entre as 2 fações (Horda e Aliança), um para um:

15. Crie um **Herói da Aliança** e um **Herói da Horda**, ambos com **80 pontos** de vida.
16. Crie um **“Game Loop”** que alterne entre um ataque da Aliança e um ataque da Horda. A cada ataque, o jogo deverá realizar as seguintes ações:
 - a. Atribuir 3 UTs para o atacante.
 - b. Exibir o saldo de UTs do atacante.
 - c. Aguardar o ataque: o jogador deverá digitar a tecla “M” para magia ou a tecla “A” para arma.
 - d. A depender da escolha do jogador, chamar os métodos **LancarMagia()** ou **AtacarComArma()** do atacante, passando o alvo como argumento.
 - e. Exibir os pontos de vida restantes do alvo.
 - f. Se o ataque for suficiente para retirar toda a vida do alvo, o jogo deverá ser encerrado com as seguintes ações:
 - i. Exibir uma mensagem informando o nome do herói vencedor e a quantidade restante de vida dele.
 - ii. Exibir uma mensagem de **Game Over**.

O conceito de "Game Loop" (laço de jogo) se refere à estrutura básica de um jogo em que o programa é executado repetidamente para atualizar o estado do jogo. Em C#, você pode implementar um Game Loop de várias maneiras, mas a abordagem comum envolve o uso de um laço principal que é executado continuamente enquanto o jogo está em execução.



UNEB
Universidade do
Estado da Bahia

INTRODUÇÃO A PROGRAMAÇÃO – JOGOS DIGITAIS

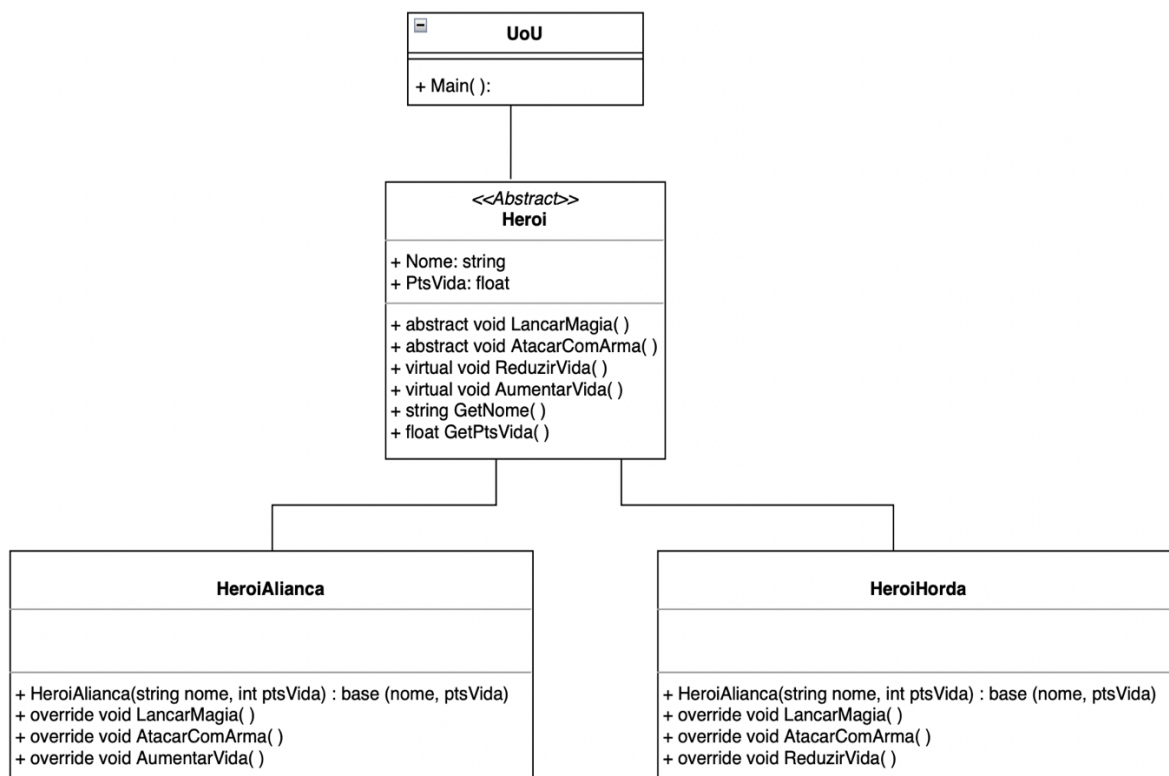
PROVA 2 – 2023.1

DATA: 19/06/23

Professor: Murilo Boratto

Aluno(a):

ANEXO I: UML DO JOGO UoU original





UNEB
Universidade do
Estado da Bahia

INTRODUÇÃO A PROGRAMAÇÃO – JOGOS DIGITAIS

PROVA 2 – 2023.1

DATA: 19/06/23

Professor: Murilo Boratto

Aluno(a):

ANEXO II: Exemplo de um Game Loop

```
while (game.Status == UoU.RUNNING)
{
    String ataque = String.Empty;

    //Ataque Alianca
    Console.WriteLine ("Ataque da Aliança:");
    Console.WriteLine ("Digite M para Magia e A para arma...");
    ataque = Console.ReadLine ();

    if (ataque.Equals ("M"))
        alianca.LancarMagia (horda);
    else if (ataque.Equals ("A"))
        alianca.AtacarComArma (horda);

    if (horda.GetPtsVida() <= 0) {
        game.Status = UoU.GAMEOVER;
        Console.WriteLine ("Game Over!");
        break;
    }

    //Ataque Horda
    Console.WriteLine ("Ataque da Horda:");
    Console.WriteLine ("Digite M para Magia e A para arma...");
    ataque = Console.ReadLine ();

    if (ataque.Equals ("M"))
        horda.LancarMagia (ali);
    else if (ataque.Equals ("A"))
        horda.AtacarComArma (alianca);

    if (alianca.GetPtsVida() <= 0) {
        game.Status = UoU.GAMEOVER;
        Console.WriteLine ("Game Over!");
        break;
    }
}
```