Especificações do Código

Classes:

**Personagem:**

**Mob:**

A classe é uma interface para os mobs do jogo, com metodos virtual getters e setters para ataque maximo, ataque minimo, vida, defesa e nome.

**Npc:**

A classe é uma sub-classe de Mob, tem como atributos *float* life, max\_attack, min\_attack, defense e uma *string* para name. Possui metodos setters e getters para todos os atributos e implementa todos os metodos de Mob. Possui um construtor vazio e um que recebe todos uma *string* e 4 valores *float*.

**Boss:**

A classe é uma sub-classe de Mob, tem como atributos *life, max\_attack, min\_attack, defense* que são valores *float, name* que é uma *string. Possui também um* atributo *skill* que é da classe Habilidade. Os metodos de Mob são implementados e possui outros dois metodos que retornam valores relacionados a *skill e mais uma para settear*.

**Arma:**

**Armadura:**

**Inventário:**

**Habilidade:**

A classe tem como atributos *damage* e *spend*, valores *int*, e *name,* uma string. Tem como metodos setters e getters dos atributos. Habilidades são passadas para Personagem e Boss para serem usados nas batalhas.

Funções

**Funções Gerais:**

**Batalha:**

Biblioteca dedicada a fazer a batalha entre o Jogador e Mobs

* **int batalha(Personagem&, Mob &)**: A função batalha recebe dois parâmetros, um “Personagem” (protagonista) e um “Mob” (inimigo) e é a responsável pelo confronto entre o personagem e o inimigo. Nela, o protagonista recebe e inflige dano, podendo derrotar o oponente e receber uma recompensa, ser derrotado e perder o jogo, ou fugir, pagando uma certa quantia pela misericórdia. A função retorna um valor inteiro entre 0 e 2, informando o resultado da partida (0 = perdeu, 1 = ganhou, 2 = fugiu).
* **int true\_damage(int, double)**: A função true\_damage recebe dois parâmetros, um inteiro (ataque) e um double (defesa). Ela é responsável por calcular o dano real de um ataque, seja do protagonista ou do inimigo. Em outras palavras, ela desconta uma porcentagem do dano de ataque a partir dos pontos de defesa de quem o recebe multiplicado por uma taxa constante (4√x) e retorna o resultado.
* **void menu2(Personagem&, Mob&)**: A função é dedicada a parte de habilidades na batalha. A função recebe um Personagem e um Mob, mostra as habilidades do Personagem com o metodo *display\_skill(),* pede a entrada da habilidade e chama a função *atacar\_hab()* com a opção selecionada. A função testa se *atacar\_hab()* vai funcionar e então testa se o Mob passado é um Boss, caso seja, é chamado a função *atacar\_hab\_boss()* e caso seja um Npc é chamada a função *atacar\_mob().*
* **void atacar(Personagem&, Mob&)**: A função recebe dois parâmetros, um “Personagem” (protagonista) e um “Mob” (inimigo), e é responsável por realizar o ataque do herói no oponente. Para isto, o procedimento busca o valor de ataque do personagem, a defesa do Mob, e chama a função “true\_damage” para calcular o dano real. Após este passo, o método desconta da vida do oponente o valor que foi retornado desta função.
* **void atacar\_mob(Personagem&, Mob&)**: A função recebe dois parâmetros, um “Personagem” (protagonista) e um “Mob” (inimigo), e é responsável por realizar o ataque do oponente no herói. Para isto, o procedimento busca o valor de ataque do Mob, a defesa do personagem, e chama a função “true\_damage” para calcular o dano real. Após este passo, o método desconta da vida do protagonista o valor que foi retornado desta função.
* **int atacar\_hab(Personagem&, Mob&, Habilidade)**:A função recebe parametros de 3 classes (Personagem, Mob e Habilidade), primeiro testa se o Personagem tem stamina suficiente para usar a habilidade passada e o valor de retorno da função é baseado nisso. Caso tenha stamina a função chama *true\_damage()* e desconta o dano da vida ddo Npc e a stamina do Personagem.
* **void atacar\_hab\_boss(Personagem&, Boss&)**:A função recebe Personagem que sera atacado e o Boss que será o atacante. A função chama true\_damage() para calcular o dano e desconta da vida do Personagem.

**Treino:**

**Jogo:**

**Menu:**

**Loja:**