

**Trabalho: Implementação de um jogo em Haskell**

**Jogo escolhido: Virtual Tabletop RPG**

**Aluno: Artur Melchiori Cerri**

O jogo escolhido foi um RPG de mesa virtual simplificado, baseado no jogo Dungeons & Dragons. A ideia foi de implementar estruturas para representação da ficha de personagem, personagens não jogáveis (inimigos), itens, e função para rolagem de dados, assim como adicionar modificadores às estas, o que é a base de qualquer RPG de mesa. Com a ficha de personagem criada, o jogador pode interagir com a história pelo terminal, escolhendo opções de caminhos a seguir e de qual maneira (através de combate, subterfúgio, persuasão etc.).

Não cheguei a concluir o trabalho para botar o jogo em prática, mas as funções principais e a introdução do jogo foram feitas. Segue abaixo informações relevantes.

## **1. Personagens, Raças, Classes, Itens e Inimigos**

Existem milhares de maneiras de se criar e customizar um personagem em um RPG de mesa, então fiz uma implementação de maneira simplificada, e com o número de raças e classes reduzidos para teste.

**Personagem, Raça, Classe, Item e NPC** (inimigos) são tipos definidos com esses nomes no arquivo vttrpg.hs. Algumas raças, classes e NPCs foram pré definidos dentro do arquivo como exemplo.

## **2. Rolagem de Dados (base do jogo)**

RPGs de mesa geralmente fazem uso de dados com diversos números de faces (lados).

A função **rollDie** recebe o número de lados do dado a ser rolado e gera um número aleatório de 1 até o número de lados.

Uma rolagem de dados é comparada a uma classe de dificuldade da ação a ser realizada. Se essa rolagem for maior ou igual à dificuldade, a ação foi bem sucedida, ou falhou caso contrário.

Mas o resultado quase nunca é baseado na rolagem pura do dado. Para isso existem modificadores de habilidades, representados na função **modifier**.

Digamos que o jogador quer realizar uma tentativa de passar despercebidamente por um inimigo. Para isso, ele estaria realizando uma ação de subterfúgio, que se baseia em **destreza**, então sua rolagem receberia o bônus referente à destreza do personagem. Se o personagem do jogador tem 16 de destreza, por exemplo, o modificador é 3, assim seria somado +3 à sua rolagem, e então feita a verificação do resultado. No caso de combates, a dificuldade de um ataque equivale à **CA** (Classe de Armadura) do inimigo).

Como neste jogo simplificado o jogador não distribui os pontos de habilidade como seria o usual e tudo é baseado nos estereótipos criados, a rolagem com modificador se baseia na classe do personagem, conforme implementado na função **rollPlusModifier**.

## **3. Conclusão**

Como dito anteriormente, o trabalho não foi concluído apesar de ter todas as funções necessárias para a execução. É possível executar o código através da função **main** e realizar a criação do personagem, gerando todos seus atributos baseados na raça e classe de escolha, assim como dar início à história, apesar desta não possuir nenhum caminho além da introdução.