“RPG em javascript: Exploração de Dungeons”

O objetivo do desafio é criar um jogo baseado em texto, onde o usuário é guiado ao longo de uma história e poderá tomar decisões escolhendo entre múltiplas alternativas.

- **Interface**

Para exibir os textos e receber as decisões do usuário, o **algoritmo deve utilizar a função alert() e prompt()**, não será necessário criar nenhuma interface de usuário em HTML ou CSS.

-
  - Para tomada de decisão, o prompt deve dizer as opções (A, B, C) e logo depois

    registrar a ação baseado no que o usuário escreveu. Ex:
    A) continuar caminhando
    B) esconder
  - Para escolher os atributos do jogador (explicação abaixo), o usuário deve escrever números inteiros, e deve ser verificado se os valores estão corretos, caso não estejam, deve ser mostrado um alerta dizendo que foi inserido um valor incorreto, caso estejam corretos, deve ser registrado os valores no objeto do jogador(player)

- **Fluxo do jogo**

No início do jogo, será necessário registrar o nome do jogador e os atributos, sendo estes HP (vida atual) e DANO no começo do jogo.

-
  - Pedir o nome do usuário e registrar no objeto player
  - O usuário tem 20 pontos de atributos no começo do jogo, e tem que dizer quantos

    pontos ele quer em vida e quantos em ataque
  - Ao distribuir os pontos, o jogo deve calcular o HP e dano da seguinte forma:

    *HP: pontos em vida * 100*

*DANO: pontos em ataque * 10*

- Isso deve ser guardado num objeto player, exemplo:

{
nome: Henrique,

- *dano: 100,*

HP: 1000 }

- Conforme o usuário receber bônus de atributos(poções ou itens) ou perder vida, esse objeto deve ser modificado de acordo

-
- Após isso, o algoritmo deve começar a exibir a história em ordem, conforme o diagrama no link a seguir:
  Link para o diagrama de decisõesLinks to an external site.

- ***Você pode adicionar recursos porém tente seguir a mesma lógica de jogo***

- Quando o diagrama exibir <nome do jogador> ou <nome do inimigo> significa que você deve obter a informação a partir do objeto daquele inimigo ou do próprio player.

  Exemplo: Parabéns, você venceu a batalha contra <nome do inimigo> (colocar o nome do monstro), a equipe comemora pelo grande feito.

- **Fluxo de batalha**

(mais detalhes no diagrama)

-
- Os inimigos devem ter atributos predefinidos como nome, HP e DANO, esses atributos serão usados na batalha contra o player.

-
- Para iniciar a batalha, você pode criar uma estrutura de repetição que irá pedir pro usuário escolher o ataque e logo irá jogar um dado de 1 a 6 (usar função Math.random) e calcular se falhou ou funcionou baseado nas probabilidades da seguinte tabela:
  Ataque simples: o ataque sempre vai funcionar

independente do dado, dano = DANO do player
Ataque combo: precisa cair o dado acima de 2 para atacar, dano = DANO do player * 1.5
Ataque especial: precisa cair o dado acima de 3 para atacar, dano = DANO do player * 2

Os passos irão se repetir até que algum dos jogadores tenha o HP <= 0 (menor ou igual a zero).

- **O fator sorte**
    -
    - Ao longo do diagrama, você verá algumas decisões que levarão o player a ter uma chance de receber um item ou não, nesses casos, você deve incluir no algoritmo um Math.random() e tentar aplicar a porcentagem descrita no desafio. Por exemplo:

//número aleatório com 50% de chance de cair 0, e 50% de cair 1

```
const resultado = Math.round(Math.random());
```

    -
    - A mesma lógica deve ser usada para jogar os dados de 1 - 6, porém usando o Math.random multiplicado por 5 e depois somando +1:

```
const resultado = Math.round(Math.random()
```