

Projeto RPG

**Emanuel Felipe , Gabriel Rodrigues, Isabella Saueressig Ricci, Raul Carneiro,
Ricardo Lewkowicz**

¹ Instituto de ensino superior de Brasília (IESB)

{nedel,flavio}@inf.ufrgs.br, R.Bordini@durham.ac.uk, jomi@inf.furb.br

Abstract. *This paper discusses the development of an application that presents a Role-Playing videogame as its main topic. Furthermore, this paper will also show the systems presented in the application, the methods used for calculating data, as well as the demonstration of all the tests made throughout the project, which also made a big impact on data balancing.*

Resumo. *Esse artigo discorre sobre o desenvolvimento de um aplicativo que apresenta um jogo do gênero RPG como seu principal tópico. Nele também serão abordados os sistemas presentes no aplicativo, os métodos utilizados para o cálculo de dados, assim como a demonstração dos testes feitos ao longo do projeto, que influenciaram na questão do balanceamento de dados.*

1. Introdução

O conceito de *Role Playing Game* (RPG) [SITOWSKI, 2015], pro português traduzido como “jogo narrativo”, é um gênero de jogo que vem de partidas puramente abstratas como o *Dungeons & Dragons* [GYGAX e ARNESON 1974], onde um grupo de jogadores se une e interpretam situações fictícias e fantásticas sem apoio material algum, objetivando a criatividade e interpretação de cada pessoa envolvida com o intuito de “entretenimento construtivo”.

A evolução do gênero de jogos RPG promoveu produtos cada vez mais rentáveis ao mercado, com uma proposta que aproxima o jogador das obras de ficção medieval-fantástica populares, como por exemplo a saga *Senhor dos Anéis* [J. R. R. TOLKIEN 1954], antes mesmo de sua adaptação às telas de cinema.

A invenção dos computadores pessoais, inaugurada pela Apple, foi um avanço considerável não somente à tecnologia, mas também ao mundo todo e às empresas de jogos que rapidamente se consolidaram em torno da possibilidade de virtualmente renderizar o conteúdo que existia somente na imaginação dos jogadores de RPG com jogos como *The Legend of Zelda* [MIYAMOTO e TEZUKA 1986], *Dragon Quest* [HORII 1986] e a série de jogos *Ultima* [GARRIOTT 1980].

Esse mercado emergente, então, se tornou cada vez maior e com o avanço da Internet foi possível conectar todos esses jogadores em uma mesma partida, inaugurando o conceito de Massive Multiplayer Online (MMO) RPG, traduzido como Jogo de RPG em Massa para Multijogadores, com jogos como *Ultima Online* [GARRIOTT 1997], *Runescape* [GOWER 1998] e *World of Warcraft* [PARDO 2004].

2. Objetivo Geral

Nosso projeto é tomar a proposta base que o gênero RPG oferece, com seu mundo fantástico e histórias épicas, evoluindo o conceito para os dispositivos móveis e as inovações que o jogo embarcado em um celular ou tablet pode conter como meio de promover espaço neste mercado e proporcionar entretenimento ao público gamer, visando o incentivo ao isolamento social, tendo em mente, durante toda a construção do aplicativo, a pandemia COVID-19 [DIOGO, 2021], hodierna e seus riscos associados a aglomeração social.

3. Objetivo Específico

O trabalho será concluído mediante as seguintes etapas:

- 3.1. Construção da interface do aplicativo via Android Studio;
- 3.2. Criação de personagens e suas características via Kotlin em Android Studio;
- 3.3. Modelo de batalha entre personagem de jogador e monstro criado pelo programa como meio de evolução do personagem;
 - 3.3.1. Conceito de “Masmorra”, espaço onde o combate será promovido de forma mais desafiadora e recompensante;

3.4. Geolocalização do jogador como meio de promover atividades ímpares em posições geográficas diferentes;

3.5. Integração da aplicação com banco de dados (sistema de gerenciamento de banco de dados a ser definido);

3.6. Tutoriais e sugestões ao jogador em forma de Chatbot.

4. Referencial Teórico

4.1. Requisições

A integração entre o usuário e o servidor está sendo feita por requisições GET e POST via API. Atualmente contamos com as seguintes requisições:

- GET (“/”) - Requisição que gera a lista das demais requisições disponíveis em nossa API.
- POST (“/jogadores/criarjogador”) - Envia-se uma classe, nome e elemento ao servidor para a criação de um novo personagem.
- GET (“/jogadores”) - Disponibiliza uma lista de todos os jogadores cadastrados e seus atributos.
- GET (“/monstros”) - Disponibiliza uma lista de todos os monstros cadastrados e seus atributos.
- POST (“/batalha/PERSONAGEMX”) - Envia-se um personagem para uma batalha, a função de batalha então monta um monstro que possui atributos calibrados com o nível do jogador.
- POST (“/loja/PERSONAGEMX/OPÇÃOY”) - Requisita-se a compra de um item OPÇÃOY ao inventário de PERSONAGEMX, caso o personagem possua moedas o suficiente.
- POST (“/inventario/PERSONAGEMX”) - Recebe-se o inventário do PERSONAGEMX.

4.2. Sistema de Personagens

O aplicativo apresenta um sistema de personagens, na qual é possibilitado ao usuário a criação de um personagem, que será utilizado na aplicação, com o limite de apenas três personagens por usuário. Cada personagem apresenta 5 atributos diferentes:

- Vida: define os pontos de vida do personagem;
- Mana: define os pontos de magia e a força mágica do personagem;
- Ataque: define a força de ataque do personagem;
- Defesa: define a defesa básica do personagem;
- Velocidade: define a velocidade básica do personagem.

Os personagens são divididos em três classes diferentes: Cavaleiro, Arqueiro e Mago, na qual o usuário é responsável por definir a classe do personagem atual. Cada

classe apresenta características e atributos iniciais diferentes, que por consequência, introduzem vantagens e desvantagens a cada uma.

Nesse caso, o Cavaleiro apresenta o atributo Ataque elevado, porém, o atributo Mana reduzido. Já o Arqueiro apresenta a Velocidade elevada e a Vida baixa. Por fim, o Mago possui a Defesa reduzida, com a Mana elevada.

O personagem apresenta a opção de selecionar um Ídolo, que consiste em um item especial que possibilita a mudança do elemento do personagem. Esse item também fornece o aumento do poder de ataque de magias que apresentam o mesmo elemento do Ídolo atual.

Ao fim de cada batalha, o personagem recebe uma quantidade de experiência, na qual se acumula até alcançar o máximo, possibilitando um aumento no nível do personagem. Esse aumento proporciona um acréscimo nos pontos de seus atributos, assim como fornece uma certa quantidade de Pontos de Sabedoria, que são utilizados para desbloquear o uso de magias mais fortes que as atuais, tornando o personagem mais poderoso a cada nível.

Após a criação do personagem, é possível acessar um menu que apresenta todas as características do personagem, como os atributos atuais, a classe, as magias disponíveis e a quantidade necessária de experiência para alcançar o próximo nível.

4.3. Sistema de Inventário

O aplicativo apresenta um sistema de inventário, na qual são armazenados itens que auxiliam o jogador nas batalhas. Os itens são divididos em dois tipos diferentes: Elixir e Upgrades. Os itens do tipo “Elixir” são utilizados para fornecer um acréscimo temporário nos pontos de um atributo específico do personagem nas Batalhas, sendo que existe um tipo específico de Elixir para cada atributo. Os itens do tipo “Upgrades” são utilizados para fornecer um aumento temporário do poder de ataque de uma magia durante as Batalhas, possibilitando um dano maior contra os inimigos.

Dentro do inventário, os itens são armazenados em locais diferentes, baseado no tipo de item. Cada item apresenta uma descrição própria, explicando a sua função e o efeito que ela proporciona ao personagem.

4.4. Sistema de Batalha

O aplicativo apresenta um sistema de batalha, na qual o personagem participa de batalhas contra os inimigos, visando o aperfeiçoamento de seus atributos e habilidades em cada luta. As batalhas seguem o sistema de batalha em turno, na qual é definida a ordem em que o personagem é permitido efetuar uma ação. Nesse caso, a ordem é

definida a partir da comparação entre os atributos de Velocidade do personagem e do adversário. Caso o jogador apresentar um valor mais alto nesse atributo, ele inicia a luta. Caso contrário, o inimigo é quem inicia.

Nas batalhas, o personagem pode efetuar quatro tipos de ações diferentes: Ações Básicas, Magias, Inventário e Fugir. A opção Ações Básicas é dividida em: Atacar e Defender.

- A opção Ações Básicas é dividida em: Atacar, Defender e Ataque da Classe
 - Atacar: possibilita o personagem de efetuar um ataque básico no adversário, na qual não retira muitos pontos de vida do monstro;
 - Defender: coloca o personagem em uma posição de defesa, possibilitando uma redução na força do ataque causado pelo adversário;
- Magias: é o principal método de ataque do personagem, na qual apresenta uma lista de diversas magias ao dispor do personagem. Dependendo do tipo utilizado, o personagem pode causar um dano alto no inimigo;
- Inventário: possibilita o uso de itens presentes no inventário do personagem para trazer vantagens a ele na batalha, como aumentar seus pontos de atributos ou aumentar a poder de ataque das magias;
- Fugir: permite o personagem escapar da luta, o que possibilita uma forma de evitar a derrota em uma luta.

O sistema de batalhas segue o método do “Pedra - Papel - Tesoura”[**BOSSCHE, 2010**], em que cada elemento do combate apresenta uma fraqueza e força. Nesse caso, esses elementos são representados em seis tipos diferentes: Fogo, Água, Natureza, Luz e Escuridão. O método segue da seguinte forma:

- Fogo é forte contra Natureza;
- Natureza é forte contra Água;
- Água é forte contra Fogo.

Esses três elementos seguem a teoria do método da "Pedra - Papel - Tesoura". Os outros dois elementos, Luz e Escuridão, são considerados elementos neutros, em que ambos não apresentam vantagens ou desvantagens contra os três elementos principais, porém, são capazes de neutralizar uns aos outros. Ou seja:

- Luz é forte contra Escuridão;
- Escuridão é forte contra Luz.

Em uma batalha, é possível fazer uso das fraquezas do adversário para facilitar a vitória do jogador. Inicialmente, ao lutar contra um monstro desconhecido, o jogador deve descobrir a sua fraqueza, visando uma vantagem durante a luta. No entanto, o inimigo também é capaz de utilizar da fraqueza do jogador para causar um dano maior. Em situações em que o jogador utiliza uma magia que apresenta o mesmo elemento do

inimigo, o dano do ataque é reduzido, incentivando o jogador a continuar buscando pela magia mais forte contra o inimigo em questão.

Quando os pontos de vida do adversário chegam a zero, o jogador vence a batalha, recebendo pontos de experiência, assim como algumas recompensas como: dinheiro ou Elixirs. No entanto, caso os pontos de vida do personagem cheguem a zero, o monstro vence a batalha, trazendo uma penalidade ao jogador, como a perda de uma certa quantidade do dinheiro atual.

4.5. Sistema de Monstros

O aplicativo apresenta um sistema de monstros, na qual são a representação de obstáculos que o jogador deve enfrentar durante o jogo. Esses monstros, assim como jogadores, são considerados personagens, que por consequência, apresentam algumas características similares às de um jogador comum. Nesse caso, esses inimigos apresentam elementos e magias pré-definidas, assim como atributos que são definidos baseados no nível atual do jogador.

Assim como os jogadores, os monstros também podem utilizar do conceito da Pedra-Papel-Tesoura para conseguir uma certa vantagem na batalha, porém, eles também sofrem as penalidades quando usam magias do mesmo elemento do jogador, ou de elementos que apresentam fraqueza contra a do jogador.

4.6. Masmorras

O aplicativo apresenta um sistema de Masmorras, que consistem de locais onde o jogador poderá entrar em batalhas contra uma certa quantidade de inimigos, até alcançar o chefe da Masmorra em questão, que consiste em um inimigo mais forte que o monstro comum, apresentando atributos maiores que o normal. Nesse tipo de batalha, o jogador deve apresentar estratégias mais elaboradas para garantir a vitória. Cada Masmorra apresenta uma temática específica, assim como o grau de dificuldade dela, sendo ele representado pelo nível recomendado para enfrentá-la.

5. Metodologia (vou escrever quando voltar)

Figure and table captions should be centered if less than one line (Figure 1), otherwise justified and indented by 0.8cm on both margins, as shown in Figure 2. The caption font must be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

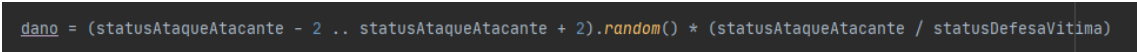
5.1 Fórmulas

No sistema de Batalha, diversas variáveis, que apresentam valores diferentes, são utilizadas para conduzir fatores diferentes na luta. Variáveis como o dano do atacante, os pontos de vida do alvo e a velocidade do atacante são alguns exemplos de variáveis utilizadas para o fluxo da batalha. No entanto, para que seja possível utilizá-las, é necessário o uso de fórmulas que vão transformar essas variáveis em valores balanceados para o equilíbrio da luta.

Nesse caso, fórmulas como o Cálculo do Dano Mágico e do Aumento de Nível são algumas reformuladas do jogo Pokémon, com alguns ajustes feitos a partir dos testes feitos no projeto, visando o balanceamento.

5.1.1 Cálculo do Dano Básico

Para calcular o valor do dano básico do atacante na batalha, é necessário fazer o uso de uma fórmula que utiliza-se do valor atual do atributo Ataque do atacante e do valor atual do atributo de Defesa do alvo. A fórmula está representada na imagem a seguir:



```
dano = (statusAtaqueAtacante - 2 .. statusAtaqueAtacante + 2).random() * (statusAtaqueAtacante / statusDefesaVitima)
```

Figura 1. Exemplo da Fórmula de Cálculo do Dano Básico

A fórmula funciona da seguinte maneira: primeiramente, é definido um valor mínimo e máximo do atributo Ataque do atacante. O valor mínimo consiste no resultado da subtração do Ataque atual por dois, enquanto o valor máximo consiste no resultado da soma do Ataque atual por dois. Após a definição do valor mínimo e máximo, a definição do valor do Ataque será feita a partir da escolha aleatória entre os números presentes situados entre o valor mínimo e o valor máximo. Logo após a escolha do número, esse valor é multiplicado pelo resultado da divisão entre o atributo Ataque atual do atacante e o atributo Defesa atual do alvo. Por fim, o resultado da multiplicação define o valor do dano que o alvo vai receber na batalha.

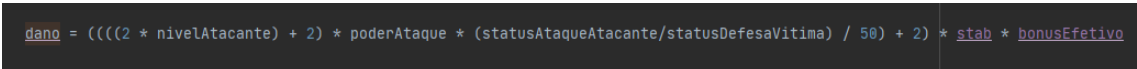
Em uma situação exemplo, o atacante apresenta um Ataque com valor de 45 e o alvo apresenta uma Defesa com valor de 50. Nesse caso, é feita a definição do Ataque mínimo e máximo, sendo respectivamente 43 e 47, e logo após, é definido um valor aleatório, que nesse caso, será o número 46. Em seguida, é feita a divisão entre o Ataque do atacante e a Defesa do alvo, resultando no valor 0,9. Com isso, é feita a multiplicação entre o resultado da divisão com o valor do ataque, gerando o valor 41,4, que será o valor do dano que o alvo receberá na batalha.

5.1.2 Cálculo do Dano Mágico

Para calcular o valor do dano mágico do atacante na batalha, é necessário fazer o uso de uma fórmula que utiliza-se de diversos atributos. Entre eles, temos: o nível atual do atacante, o valor do dano básico da magia utilizada, o valor da Mana do atacante, o valor

da Defesa do alvo, o elemento da magia utilizada, o elemento do atacante, e por último, o elemento do alvo.

Tirando os atributos anteriores, também será utilizado ferramentas que proporcionarão um acréscimo ou decréscimo no valor do dano, dependendo da situação atual do atacante. Nesse caso, essas ferramentas são: STAB, que consiste no bônus utilizado quando o atacante faz o uso de uma magia que apresenta o elemento similar ao do atacante, e o Efetivo, que consiste no bônus utilizado quando o atacante utiliza uma magia que apresenta um elemento forte contra o elemento do alvo. A fórmula está representada na imagem a seguir:



```
dano = (((2 * nivelAtacante) + 2) * poderAtaque * (statusAtaqueAtacante/statusDefesaVitima) / 50) + 2) * stab * bonusEfetivo
```

Figura 1. Exemplo da Fórmula de Cálculo do Dano Mágico

A fórmula funciona da seguinte maneira: primeiramente, é feito uma multiplicação do nível atual do atacante por dois, na qual o resultado da mesma é somado por mais dois. Logo após, o valor resultante é multiplicado pelo valor do dano básico da magia utilizada, e em seguida, o resultado dessa operação é multiplicado pelo resultado da divisão entre o valores da Mana atual do atacante e da Defesa atual do alvo. Com isso, o resultado dessa multiplicação é dividido por cinquenta, e logo após, é feita uma soma por dois. Por fim, o resultado dessa operação toda passa por uma multiplicação entre os bônus de batalha, resultando no valor do dano do ataque mágico.

Dependendo dos elementos da magia, do atacante e do alvo, os bônus de batalha podem passar por algumas alterações nos seus valores. Nesse caso, como explicado anteriormente, o bônus do STAB apenas estará presente caso o atacante apresentar o mesmo elemento da magia utilizada. Já em relação ao Efetivo, quando a magia utilizada apresenta o elemento que apresenta uma vantagem contra o elemento do alvo, o bônus contribuirá para o aumento do valor do dano mágico. Caso a magia apresenta o mesmo elemento que o monstro, o bônus não é aplicado ao valor final. E em situações em que a magia apresenta o elemento que possui uma desvantagem contra o elemento do alvo, o dano do ataque sofre um decréscimo, diminuindo o valor final do dano.

Em uma situação exemplo, o atacante apresenta uma Mana com valor de 50, o elemento Fogo e um nível com valor 20, o alvo apresenta uma Defesa com valor de 45 e o elemento Natureza, a magia utilizada apresenta um dano de 10 e o elemento Fogo. Nesse caso, é feita a multiplicação do nível do atacante por 2, e logo após, a soma do resultado por 2, resultando no valor 42. Com isso, esse valor é multiplicado pelo valor do dano da magia, resultando no valor 420, e logo após, é feita a multiplicação com o valor resultante da divisão entre a Mana e Defesa do atacante e do alvo, respectivamente, sendo 1,11, e por fim: 466,2.

Esse resultado é dividido por 50, resultando em 9,32, na qual é somado por 2, transformando em 11,32. Por fim, seguindo as condições de bônus de ataque, o valor final é multiplicado 4, resultando no valor 45,28, representando o dano final do ataque mágico.

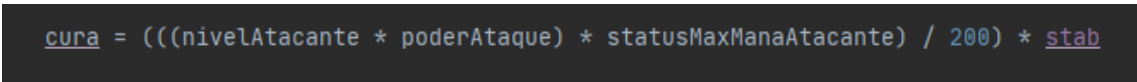
5.1.3 Cálculo do Acréscimo da Defesa

Quando o personagem escolhe a opção de Defender na batalha, sua Defesa atual sofre um pequeno acréscimo temporário, visando reduzir o dano gerado pelo inimigo. Nesse caso, o valor atual da Defesa é somado por 20, retornando ao valor anterior no fim do turno.

5.1.4 Cálculo da Cura

Para calcular o valor da cura, é necessário fazer o uso de uma fórmula que utiliza-se do nível atual do atacante, do valor do dano gerado pela cura e do valor atual do atributo Ataque do atacante. A fórmula também utilizará o STAB, que consiste no bônus de batalha mencionado anteriormente. Nesse caso, o STAB vai seguir o mesmo princípio da menção anterior, proporcionando apenas o efeito do bônus caso o atacante apresentar o mesmo elemento da magia Cura.

A fórmula está representada na imagem a seguir:



```
cura = (((nívelAtacante * poderAtaque) * statusMaxManaAtacante) / 200) * stab
```

Figura 1. Exemplo da Fórmula de Cálculo do Cura

A fórmula funciona da seguinte maneira: primeiramente, é feita a multiplicação entre o nível do atacante e o valor do dano da Cura. Logo após, o resultado dessa operação é multiplicado pelo valor da Mana do atacante, sendo feita a divisão entre esse resultado por 200, em seguida. Por fim, é feita a multiplicação do bônus da batalha, gerando no valor final da cura.

Em uma situação exemplo, o atacante apresenta uma Mana com valor de 50, o elemento Luz e um nível com valor 20. A magia Cura apresenta um dano de valor 30. Nesse caso, é feita a multiplicação entre o nível e o dano da Cura, resultando no valor 600. Em seguida, esse valor é multiplicado pela Mana do atacante, gerando no valor 30.000, que logo após, é dividido por 200, resultando em 150. Por apresentar o mesmo elemento da magia, o resultado final recebe o bônus STAB, sendo multiplicado por 2, resultando em 300 de dano de cura, proporcionando um acréscimo de 300 nos pontos de vida do atacante.

5.1.5 Cálculo do Aumento de Nível

Todo personagem jogador possui um atributo numérico inteiro Nível que qualifica seu avanço no jogo, jogadores com nível maior possuem personagens mais fortes que jogadores de nível menor. Para se evoluir o nível do personagem precisa-se antes atingir uma meta de Pontos de Experiência, atributo este que mede o progresso do personagem entre seu nível atual e o próximo nível.

O jogador recebe pontos de experiência ao vencer combates, o cálculo dos pontos de experiência recebidos é igual a $NM * 10 + (NM * RANDOM)$ onde NM é o Nível do Monstro e RANDOM é um valor aleatório no intervalo aberto de 1 a 10. Esses pontos de experiência vão sendo somados a cada batalha vencida até atingirem uma meta que é calculada por $(NP * 100) * (NP * 0.1)$ onde NP é o Nível do Personagem. Desta forma nós conseguimos manter um avanço de nível com dificuldade incremental, de forma que seja mais difícil atingir um nível mais alto e, assim, valorizando jogadores mais esforçados ou veteranos.

Quando o personagem aumenta o seu nível, os valores atuais dos seus atributos passam por uma fórmula para determinar os acréscimos que terão no novo nível. A fórmula apresenta apenas dois componentes para o seu cálculo: o valor base do atributo passando pelo acréscimo e o novo nível conquistado pelo personagem. No entanto, a classe do personagem também apresenta uma influência na fórmula em questão, influenciando no valor adicionado ao final da operação.

A fórmula está representada na imagem a seguir:

```
this.maxVida += ((2 * statusBaseVida) * nivel)/100 + nivel + 8
this.maxMana += ((2 * statusBaseMana) * nivel)/100 + nivel + 10
this.maxAtaque += ((2 * statusBaseAtaque) * nivel)/100 + nivel + 10
this.maxDefesa += ((2 * statusBaseDefesa) * nivel)/100 + nivel + 10
this.maxVelocidade += ((2 * statusBaseVelocidade) * nivel)/100 + nivel + 12
```

Figura 1. Exemplo da Fórmula de Cálculo do Aumento de Nível

Como explicado anteriormente, a fórmula é utilizada para cada atributo do personagem, com elas apresentando diferenças no atributo utilizado e no último valor adicionado na operação. Nesse caso, a fórmula em questão é utilizada para os personagens que apresentam a classe Arqueiro. Cada classe possui uma proficiência, ou vantagem, em um determinado atributo, assim como uma desvantagem em um outro atributo. Essa característica é demonstrada por meio da adição final da fórmula, na qual caso o personagem apresentar essa vantagem em um atributo, o valor padrão desse componente, que é o número 10, sofre um acréscimo de dois números, resultando no valor 12. Caso contrário, o valor sofre um decréscimo de dois números, resultando no valor 8. O valor padrão é apenas utilizado quando o personagem não apresenta uma vantagem e nem uma desvantagem no atributo em questão.

Em relação ao Arqueiro, essas características estão presentes no atributo Velocidade, na qual ele apresenta uma vantagem, e no atributo Vida, na qual ele apresenta uma desvantagem. Já em relação ao valor base do atributo, ele consiste no valor padrão que cada classe possui no nível 1, sendo ele utilizado para determinar as mudanças dos atributos no próximo nível.

A fórmula funciona da seguinte maneira: o valor base do atributo é multiplicado por 2, e seu resultado é multiplicado pelo novo nível do personagem. Logo após, o resultado dessa multiplicação é dividido por 100, com o valor resultante sendo somado pelo novo nível e, por fim, pelo valor final, variando dependendo da classe do

personagem, como explicado anteriormente. O valor resultante da operação é definido como o novo valor do atributo, tornando o personagem mais forte que antes.

Em uma situação exemplo, o personagem aumenta de nível para o 30, e logo em seguida, passa pela fórmula do aumento de nível. Por ser da classe Arqueiro, ele apresenta como valores bases: 5, 6, 6, 6 e 7, para os atributos Vida, Mana, Ataque, Defesa e Velocidade, respectivamente. Com isso, os valores são introduzidos a fórmula, com a Vida apresentando 41 pontos, a Mana, o Ataque e a Defesa apresentando 44 pontos, e por fim, a Velocidade apresentando 46 pontos.

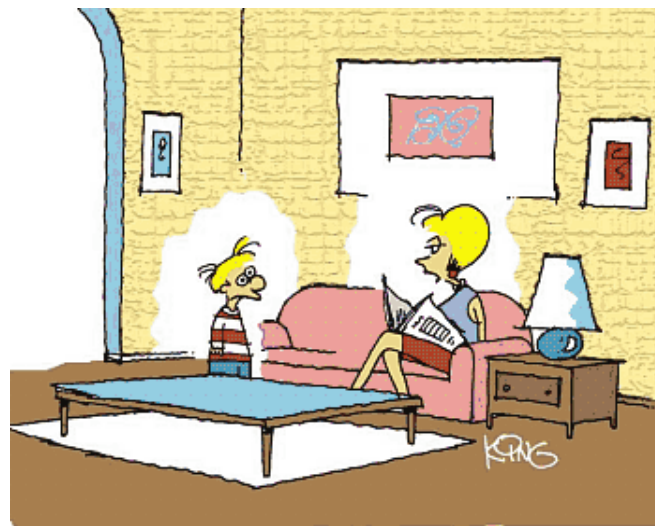
Essa fórmula é também utilizada para determinar os valores dos atributos dos monstros, com a diferença que o nível será determinado pelo nível da Masmorra atual.

5.1.6 Cálculo do Ganho da Experiência (Todo seu Ricardao achuaachuah)

5.2 Balanceamento (Vou fazer dps de começarmos os testes)

5.3 Testes (Vou fazer quando fizermos os testes)

5.4 Banco de Dados (comparação entre os Banco de Dados)



*"No, you weren't downloaded.
Your were born."*

Figure 1. A typical figure

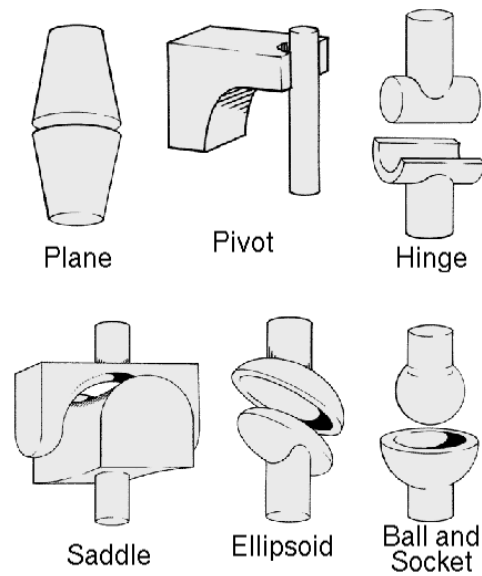


Figure 2. This figure is an example of a figure caption taking more than one line and justified considering margins mentioned in Section 5.

— In tables, try to avoid the use of colored or shaded backgrounds, and avoid thick, doubled, or unnecessary framing lines. When reporting empirical data, do not use more decimal digits than warranted by their precision and reproducibility. Table caption must be placed before the table (see Table 1) and the font used must also be Helvetica, 10 point, boldface, with 6 points of space before and after each caption.

Table 1. Variables to be considered on the evaluation of interaction techniques

	Chessboard top view	Chessboard perspective view
Selection with side movements	6.02 ± 5.22	7.01±6.84
Selection with in- depth movements	6.29±4.99	12.22±11.33
Manipulation with side movements	4.66± 4.94	3.47±2.20
Manipulation with in- depth movements	5.71 ±4.55	5.37 ±3.28

6. Resultados Obtidos

All images and illustrations should be in black and white, or gray tones, excepting for the papers that will be electronically available (on CD-ROMs, internet, etc.). The image resolution on paper should be about 600 dpi for black and white images, and 150-300 dpi for grayscale images. Do not include images with excessive resolution, as they may take hours to print, without any visible difference in the result.

7. Conclusão

~~Bibliographic references must be unambiguous and uniform. We recommend giving the author names references in brackets, e.g. [Knuth 1984], [Boulie and Renault 1991]; or dates in parentheses, e.g. Knuth (1984), Smith and Jones (1999).~~

~~— The references must be listed using 12 point font size, with 6 points of space before each reference. The first line of each reference should not be indented, while the subsequent should be indented by 0.5 cm.~~

8 . Bibliografia

PETERSON, Jon. Website que mostra a linha do tempo do jogo. Dungeons & Dragons, 2021. Disponível em: [History: Forty Years of Adventure | Dungeons & Dragons](#) - Acessado em: 18 de Março de 2021.

DOUGHAN, David. Bibliografia sobre o autor J.R.R Tolkien. The Tolkien Society, 2021. Disponível em: [Biography – The Tolkien Society](#) - Acessado em: 18 de Março de 2021.

WATERS, Darren. Artigo sobre o número de vendas de Dungeons & Dragons. BBC News, 2004. Disponível em: [BBC NEWS | What happened to Dungeons and Dragons?](#) - Acessado em: 18 de Março de 2021.

GADELHA, Julia. Artigo sobre a Evolução dos Computadores. Universidade Federal Fluminense, ???. Disponível em: [A EVOLUÇÃO DOS COMPUTADORES](#) - Acessado em: 18 de Março de 2021.

WELL, Thomas. Artigo sobre a história do Dragon Quest. Thomas Well, 2019. Disponível em: [Enix Origins: The Story Behind Dragon Quest](#) - Acessado em: 18 de Março de 2021.

WEBSTER, Andrew. Timeline da franquia The Legend of Zelda. The Verge, 2016. Disponível em: [30 years of Zelda: a timeline of the legend so far](#) - Acessado em: 18 de Março de 2021.

MOWATT, Todd. Entrevista do Shigeru Miyamoto, criador do The Legend of Zelda. Amazon, 2007. Disponível em: [In the Game: Nintendo's Shigeru Miyamoto](#) - Acessado em: 18 de Março de 2021.

BARTON, Matt. Artigo sobre a história dos jogos RPG para computadores. Gamasutra, 2007. Disponível em: [The History of Computer Role-Playing Games Part 2: The Golden Age \(1985-1993\)](#) - Acessado em: 18 de Março de 2021.

DODSON, Sean. Artigo sobre o desenvolvimento de RuneScape. The Guardian, 2003. Disponível em: [Rune to move | Games](#) - Acessado em: 18 de Março de 2021.

WILLIAMS, Mike. Artigo sobre o desenvolvimento do World of Warcraft. USGamer, 2019. Disponível em: [How World of Warcraft Was Made: The Definitive Inside Story of Nearly 20 Years of Development](#) - Acessado em: 18 de Março de 2021.

DIOGO, Darcianne. Notícia sobre o número de casos e mortes de COVID-19. G1, 2021. Disponível em: [Covid-19: DF registra 1.322 casos em 24 horas e total de mortes chega a 5.206](#) - Acessado em: 18 de Março de 2021.

SITOWSKI, André. Artigo sobre o gênero RPG. Medium, 2015. Disponível em: [RPG — Tudo que você precisa saber sobre esse tal Role Playing Game](#) - Acessado em: 18 de Março de 2021.

BOSSCHE, Andrew. Artigo sobre o uso do método “Pedra-Papel-Tesoura” nos jogos. Gamasutra, 2010. Disponível em: [Analysis: Everything is Rock, Paper, Scissors](#) - Acessado em: 15 de Maio de 2021.

POKEDREAM, Calculadora de Status Base do jogo Pokémon Gold. Pokedream, ???. Disponível em: [Gold/Silver - Stat Calculator - PokeDream](#) - Acessado em: 25 de Maio de 2021.

CHARD, Ben; LAU, Vincent. Artigo sobre a explicação do cálculo de Status do jogo Pokémon Shield. Gamer Guides, 2019. Disponível em: [Understanding Stats - Pokémon 101 - Advanced Trainer Info | Pokémon: Sword & Shield](#) - Acessado em: 25 de Maio de 2021.