

Hero of Goahnd - Relembrando o Desenvolvimento de RPGs Clássicos

João Pedro Peres, Martha Pires da Rocha, Rodrigo T. Marques, Ronaldo P. de Oliveira Jr. e Alison R. Panisson
 Centro de Engenharias
 Centro de Desenvolvimento Tecnológico
 Universidade Federal de Pelotas – Pelotas, Brasil
 Emails: {jpperes, martha.rocha, rodrigo.torres, ronaldo.pereira, alison.panisson}@ufpel.edu.br

Resumo—Com o avanço tecnológico dos últimos anos (placas gráficas, poder de processamento, capacidade de memória, inteligência artificial) a indústria de games é repleta de jogos muito atraentes "graficamente", com boa jogabilidade, tempo de resposta, dentre outros aspectos. Vivenciando um cenário em que os usuários estão cada vez mais exigentes, aumentando a competitividade entre as empresas de jogos nesse ramo.

Com o intuito de resgatar os clássicos jogos que marcaram a infância de várias pessoas nas décadas de 80 e 90, originando certa nostalgia de *games old school*, desenvolvemos um jogo RPG (*Role-playing game*) clássico com enfoque lúdico e imaginativo. Nesse desenvolvimento, utilizamos a linguagem de programação C na sua versão básica, tanto para a criação dos algoritmos que implementam a lógica do jogo, como também para projetar interfaces gráficas visualmente simples, porém com uma complexidade estrutural, resultando em um ambiente de boa jogabilidade, onde o principal intuito é atrair o usuário que opta por jogá-lo.

Keywords—Jogos Clássicos, Role-Playing Game (RPG); Linguagem de Programação C.

I. INTRODUÇÃO

Ao voltarmos alguns anos atrás na época do Atari [5] e compararmos aos jogos desenvolvidos na atualidade, há uma enorme diferença nos gráficos e na jogabilidade, os quais na época eram muito rudimentares. Jogos em que o jogador se movimenta entre salas e cumpre objetivos ficaram muito famosos naquela época, como, por exemplo: Adventure (1979) [2] e Swordquest (1980) [3].

O sucesso destes jogos foram uma inspiração para a criação de outros, os quais foram sendo aprimorados em questão de jogabilidade e gráficos, até chegar ao que conhecemos hoje. Desta forma, podemos dizer que eles foram os jogos originários.

Inspirados por esses jogos clássicos, decidimos desenvolver um jogo de RPG [4] estilo "*old school*". E já que decidimos desenvolver um jogo nesse estilo, por que não utilizar uma linguagem clássica que foi utilizada como base para a criação de diversas outras linguagens de programação? Sim, utilizamos o C na sua versão básica para desenvolver esse projeto!

Nesse artigo curto, apresentamos o jogo nomeado *Hero of Goahnd*. Um jogo do tipo RPG desenvolvido utilizando a linguagem de programação C na sua versão básica, inspirado pelos jogos clássicos do Atari. No decorrer desse artigo,

na seção II, descrevemos o jogo desenvolvido, incluindo a história por trás do mesmo: classes de personagens e monstros, os diferentes cenários modelados, detalhes de jogabilidade e mapa, batalhas, itens e drops, inventário e como atraímos o interesse do jogador através de uma técnica de evolução do personagem. Após, na seção III, descrevemos uma história de jogo, ou seja, jogamos o jogo desenvolvido, evoluindo o personagem, até alcançar a vitória do jogo. Neste processo, apresentamos as interfaces clássicas desenvolvidas e a evolução do personagem no jogo. Por último, na seção IV, descrevemos brevemente alguns detalhes da implementação e dos desafios de implementar o jogo. Seguido de algumas colocações finais na seção V.

II. HERO OF GOAHND

A história do jogo se passa em um mundo onde diferentes reinos viviam inicialmente em harmonia. Esse cenário mudou com a chegada ao poder, no reino de Ultigart, de um rei impiedoso e ganancioso, chamado Hendark, o qual, com o tempo, conquistou os demais reinos. Neste jogo, com o objetivo de fortalecer seu personagem, um jogador deve adquirir itens e nível de experiência suficiente para entrar na Fortaleza Urzak e derrotar o rei Hendark.

A. Classes

O jogador poderá escolher entre quatro classes equilibradas para entrar em sua aventura: Mago, Guerreiro, Arqueiro e Druida, sendo duas classes de dano mágico e duas de dano físico. As classes de dano mágico (Mago e Druida) possuem os mesmos atributos totais que as classes de dano físico (Guerreiro e Arqueiro), diferindo apenas na distribuição destes atributos.

Mesmo que hajam diferentes classes, o jogador não está limitado a usar armas específicas de sua classe, tendo a opção de utilizar qualquer equipamento.

B. Cenários

Ao iniciar sua aventura, o jogador pode se deparar com algumas das cinquenta estruturas geradas aleatoriamente pelo mapa, sendo estas distribuídas em cinco tipos diferentes comuns e uma especial (Fortaleza Urzak), onde se encontra o objetivo do jogo: o *boss* final.

As cinco estruturas comuns são: Bosque, Castelo, Caverna, Pântano e Ruínas, onde cada uma possui seus monstros específicos.

C. Monstros

Durante a sua jornada, o jogador pode encontrar trinta e seis diferentes monstros classificados como: monstros de ataque físico, mágicos e mistos. Foram designados seis tipos de monstros para cada tipo de estrutura, sendo cinco considerados comuns e apenas um como *boss* da estrutura. Ademais, os monstros possuem um nível máximo para garantir que o jogador, após uma longa trajetória, possa alcançar a vitória.

D. Jogabilidade

Ao inicializar um novo jogo, o personagem se encontra sozinho no mapa, apenas com a Fortaleza Urzak à vista. Para que encontre novas estruturas, o jogador deve se locomover pelo mapa a procura delas, visto que somente são reveladas quando o jogador se aproxima. Grande parte da dinâmica RPG, isto é, regras e sistema de jogo, é integrada pela aleatoriedade de resposta às decisões tomadas pelo jogador. Desta maneira, o programa atua como o *Game Master* da partida e os monstros atuam como *non-player characters* (NPC's) [4]. A progressão dos atributos e vida dá-se de forma exponencial, assim, atraindo ainda mais o jogador que tem a sensação de evolução constante.

E. Mapa

O mapa foi desenvolvido com um tamanho de 25x75 posições de uma matriz e, conforme o deslocamento pelo mesmo, o jogador tem 5% de chance de ser surpreendido por monstros (comuns) aleatórios e, a cada movimento realizado, há um acréscimo na fome do personagem. No caso de a fome chegar a 100%, a vida começa a sofrer decréscimos ao movimentar-se até que o personagem faça o consumo de algum alimento.

Os ambientes do jogo possuem tamanho de 15x40 posições de uma matriz, contendo um *boss* cada ambiente. Ao movimentar-se pelo ambiente, há 12.5% de chance de ser surpreendido por um monstro aleatório específico desta localidade. O objetivo dentro da estrutura é derrotar o *boss* daquele ambiente, evitando assim o *spawn* de mais monstros no mesmo cenário.

Durante a visualização do mapa, existem opções para movimentar-se, abrir inventário, salvar jogo e salvar e sair do jogo. Estas duas últimas não são disponíveis enquanto o jogador estiver dentro de um cenário.

F. Batalhas

No quesito de batalhas, o jogador tem duas opções ao encontrar um monstro: fugir ou lutar. Se selecionar a opção de lutar, o personagem ataca o monstro com 20% de chance de errar o ataque. Caso acerte, tem 20% de probabilidade de

causar dano crítico ao monstro (o mesmo vale para a vez do oponente). Ao selecionar a opção de fugir, o monstro realiza três ataques sem a possibilidade de o jogador revidar, tendo apenas a chance de 20% de esquivar.

G. Itens/Drops

Ao derrotar um monstro, é criada uma chance de 50% de ocorrer o *drop* de um item, podendo ser uma arma, armadura, poção de vida, comida ou mochila. No total, o jogo conta com quarenta e cinco itens, divididos em cinco tipos de raridade: comum, incomum, raro, lendário e ancestral. Cada nível de raridade de item possui quatro armas, duas armaduras, uma poção de vida, uma comida e uma mochila.

H. Inventário

Ao acessar o inventário, o jogador pode equipar armas e armaduras, consumir poções e comidas, descartar itens e visualizar informações sobre o item desejado. Ademais, o inventário possui um limite de armazenagem (inicialmente dez) que pode ser incrementado ao encontrar mochilas maiores nos drops.

I. Experiência/Nível

Ao derrotar um monstro, são adquiridos pontos de experiência. O personagem começa no nível 1, necessitando de 100 pontos de experiência para subir de nível. O nível influencia diretamente na vida e atributos de batalha do personagem, assim como nos níveis dos oponentes e nos pontos de experiência necessários para o próximo nível. Os níveis consecutivos necessitam 20% mais pontos em relação ao nível anterior para serem concluídos, e a cada vez que o jogador eleva um nível do personagem, seus atributos aumentam em 5% em relação ao anterior.

III. UMA HISTÓRIA DE EXECUÇÃO DO JOGO

Nessa seção descreveremos uma narrativa de execução do jogo, juntamente com as telas de interfaces mais relevantes durante essa execução.

O herói chamado Dimitri dará entrada no mundo de Goahnd para tentar combater o Rei Hendark e devolver a paz a Goahnd. Dimitri, ao andar para o leste, encontra um castelo no qual contém monstros. Como Dimitri não teme suas obrigações, decide entrar no castelo e enfrentar os monstros que lá se encontram (Fig. 1).



Figura 1. Mapa principal do jogo.

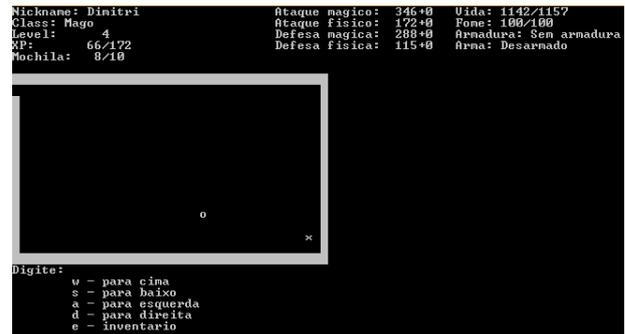


Figura 4. Movimentação em cenários internos.

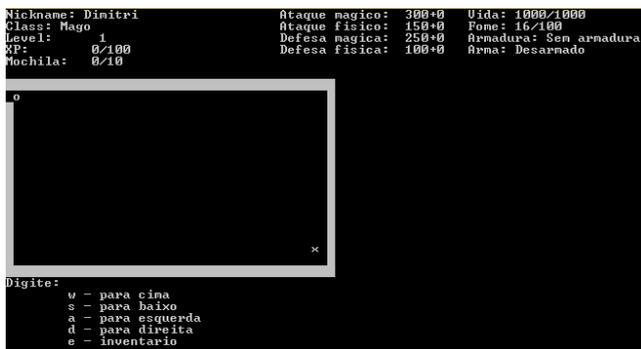


Figura 2. Cenários internos.

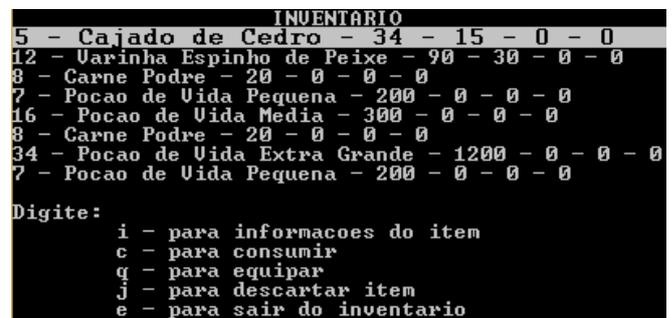


Figura 5. Inventário.

Ao entrar no castelo, Dimitri se depara com um boss, que está bem longe dele. Para chegar até ele, deve passar por um caminho cheio de monstros que tendem a atacá-lo quando menos ele espera (Fig. 2).



Figura 3. Exemplo de batalha.

Oh, não! Dimitri foi surpreendido por um Cérbero! O herói não fraqueja e decide enfrentar seu oponente (Fig 3).

Dimitri conseguiu vencer a luta, mas ainda deverá conseguir mais itens e experiência para enfrentar o chefe do Castelo (Fig 4).

Após algumas batalhas, o herói adquiriu nível, itens, um aumento na fome e redução da vida. Para isso, deve acessar o inventário para conferir os itens e utilizar os consumíveis que estão à sua disposição (Fig 5).

Percebe-se que o personagem melhorou o equipamento e se encontra pronto para mais batalhas! (Fig 6).

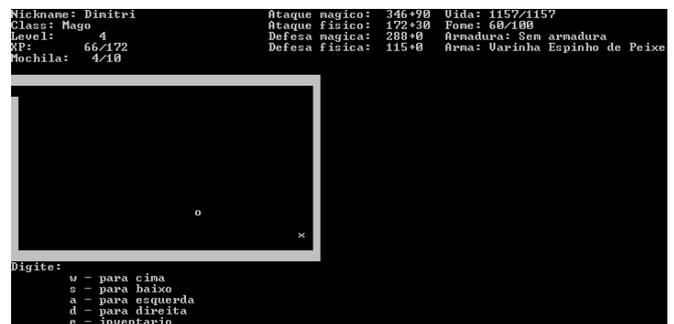


Figura 6. Personagem após utilização de itens do inventário.



Figura 7. Batalha contra o Demônio das Ruínas.

Dimitri decide enfrentar o chefe, se deparando com o Demônio das Ruínas! (Fig 7).

Após vencer o chefe do castelo, os monstros presentes sentem-se ameaçados e decidem fugir por não terem mais um comandante.

Dimitri então parte para outros lugares, aumentando seu nível e obtendo outros itens, com o objetivo de conseguir entrar na Fortaleza Urzak.

Após um longo caminho repleto de batalhas, Dimitri decide entrar na Fortaleza Urzak, mesmo ainda não sendo capaz de derrotar o Rei Hendark. .

Conforme acumulava uma sucessão de vitórias, o herói conquistou nível suficiente para enfrentar Hendark. Um pouco assustado, mas com a cabeça erguida, decide enfrentá-lo. Foi uma batalha sangrenta, mas Dimitri consegue escapar de forma vitoriosa.

IV. ESTRUTURA DO PROGRAMA

O jogo Hero of Goahnd, como previamente constatado, foi desenvolvido utilizando a linguagem de programação C na sua versão básica. Para isto, foram utilizadas 4 estruturas que permitem armazenar informações: do personagem, dos monstros, dos itens e da mochila.

A mochila foi implementada a partir de uma lista duplamente encadeada, sendo delimitada por uma variável global que é atualizada toda vez que o jogador encontrar uma nova mochila de maior capacidade que a atual.

Para o gráfico do mapa, dos cenários e do personagem foram utilizados caracteres da tabela ASCII.

Para o jogo em geral, foram desenvolvidas trinta e seis funções diferentes, cada uma com um propósito específico, como, por exemplo: andar pelo mapa, batalhas, gerar estruturas, salvar dados, entre outros.

Em questão de drop de itens, foi atribuída uma função randômica, a qual disponibiliza chance de 50% de dropar um item. Além disso, os itens possuem diferentes classes: comuns, incomuns, raros, lendários e ancestrais. Dentro dessa chance de dropar um item, há uma chance de dropar uma classe de item, sendo 60% de chance de cair um item comum, 30% um incomum, 6% um raro, 3% um lendário e 1% um ancestral. Ainda mais, dentro dessas classes, há uma chance de cair um item específico: arma (2,5% cada) armadura (5% cada), poção de vida (30%), comida (45%) e mochila (5%).

V. CONCLUSÃO

Na indústria atual dos jogos, a demanda pela qualidade e performance dos mesmos tende a estimular este tão acirrado mercado. A carência de meios gráficos torna um jogo menos atrativo para um mercado mais amplo. Entretanto, ainda há aqueles que são apaixonados pelo jogos clássicos de interfaces modestas que deram origem ao nosso cenário atual de jogos. Assim, o enfoque técnico do jogo tratado nesse trabalho possui cunho estrutural, e não gráfico. Foram desenvolvidas estruturas e sub-estruturas que dão vida ao jogo, de modo a inserir o jogador ao máximo dentro do contexto RPG através de referências tomadas como, por exemplo, a renomada saga “The Lord of Ring” [1]. Assim, também remetemos a jogos clássicos como: Adventure (1979) [2]

e Swordquest (1980) [3] que deram início nos anos 70 e 80 ao que hoje é o grande mercado de jogos e serviram de inspiração para o desenvolvimento desse trabalho.

Desenvolver um jogo exclusivamente na linguagem C e não perder a interação com o usuário e também a demonstração de interesse do mesmo é considerada uma missão difícil, mediante aos jogos que possuem gráficos e jogabilidade mais complexos e sofisticados. Diante disso, inseriu-se o jogador representado (de maneira gráfica) no ambiente do jogo, além de apresentar simulações (2D) de batalha e incluir aspectos vitais, como a fome, com o intuito de assemelhar-se com a realidade.

A combinação da linguagem C com o contexto e o universo RPG torna possível a criação de diversas hipóteses de jogabilidade durante a execução do jogo. Portanto, é este o objetivo do jogo: agregar possibilidades à imaginação, tornando-se um canal de entretenimento ao usuário que estiver disposto a embarcar nesta aventura.

Como resultado, baseado em avaliações preliminares realizadas com os próprios desenvolvedores desse projeto, obtivemos um jogo clássico com enfoque lúdico e imaginativo, com interfaces que são graficamente simples mas possuem estruturas complexas, capaz de prender o interesse do jogador através da evolução do personagem.

Como trabalho futuro, pretendemos realizar testes de jogabilidade com usuários que não participaram do desenvolvimento do jogo, permitindo assim um *feedback* maior sobre as constatações alcançadas no desenvolvimento de Hero of Goahnd.

ACKNOWLEDGMENT

Este trabalho foi desenvolvido pelos alunos da disciplina de Programação de Computadores, oferecida no curso de Engenharia de Controle e Automação pela Universidade Federal de Pelotas (UFPEL).

REFERÊNCIAS

- [1] W. G. Hammond, and C. Scull, *The Lord of the Rings: A Reader's Companion*, HarperCollins, 2005.
- [2] W. Robinett, *The Annotated Adventure*, 1979.
- [3] J. Hardie, *Swordquest Interview*. Disponível em: <http://www.atarihq.com/2678/swordqst.html>. Acesso em: Agosto de 2019.
- [4] J. Kim, “Narrative” or “Tabletop” RPGs. Disponível em: <http://www.darkshire.net/~jhkim/rpg/whatis/tabletop.html>. Acesso em: Agosto de 2019.
- [5] ATARI. In: WIKIPÉDIA: a enciclopédia livre. Wikimedia, 2019. Disponível em: <https://pt.wikipedia.org/wiki/Atari>. Acesso em: Agosto de 2019.