

Dance Dance Revolution

Jéssica Santos Borges e Leticia Meurer

*Universidade Federal de Santa Catarina (UFSC)

Resumo—Este relatório tem como objetivo apresentar a formação de um jogo em terminal utilizando a linguagem C de programação. Nele estão contidos os problemas encontrados para execução e os métodos utilizados para que fosse possível formar uma adaptação de Dance Dance Revolution.

Palavras-chave—Programação, linguagem C, jogo de Arcade.

Abstract—This report aims to present the formation of a game in terminal using the C programming language. In it are contained the problems encountered for execution and the methods used so that it was possible to form an adaptation of Dance Dance Revolution.

Index Terms—Programming, C language, Arcade game.

I. INTRODUÇÃO

ESTE projeto tem como intuito realizar uma atividade prática que utilize os conhecimentos adquiridos durante as aulas para a implementação de um jogo em terminal. Assim, utilizou-se do sistema operacional Ubuntu e a linguagem C.

Teve-se como inspiração o antigo jogo de Arcade lançado em 1998 no Japão - Dance Dance Revolution (Figura 1), no qual, os jogadores devem acertar as setas em sequência conforme as mesmas aparecem na tela. Existem algumas variações deste jogo, no entanto, todos seguem a mesma ideia.



Figura 1. Dance Dance Revolution (1998) - Fonte: Emuparadise (2019).

A elaboração do relatório tem por objetivo discutir a realização do jogo para um único usuário. Primeiramente realizou-se uma verificação de todos os pontos que precisavam ser melhorados como o layout inicial, a distribuição das setas e a inclusão de tempo. Após isso, realizou-se um levantamento de problemas, tendo como o maior deles o ajuste da música de fundo.

A. Problemas

Nesta subseção serão apresentados alguns problemas encontrados para elaboração do jogo como um todo e suas possíveis resoluções.

1) *Música*: um dos problemas encontrados foi a implementação de música no decorrer da atividade, o qual gerou uma série de erros. Primeiramente a música tocava inteira e só iniciava-se o jogo quando a música chegava ao fim, depois o jogo começava e a música só se iniciava quando o mesmo terminava. Uma das maiores dificuldades foram encontrar soluções simples e eficientes para o ajuste nos problemas.

Para que a música e jogo ocorressem ao mesmo tempo, foi adicionado o símbolo "&" a função `play %s(nome da música.extensão) -q &"` o que permitiu que ambos funcionassem corretamente porém, a música continua a tocar mesmo com o término do jogo, criando assim um novo erro.

O problema por fim foi solucionado pela adaptação da função `kill`, a `kill`, que tem por finalidade interromper a parte desejada do código, o que neste caso, utilizou-se para encerrar a música juntamente com o terminar do jogo.

No decorrer do trabalho serão apresentados a metodologia utilizada para criação do jogo, seguida pelos resultados encontrados durante a execução, a conclusão e o apêndice, onde encontra-se o código de criação do jogo.

II. METODOLOGIA

O jogo utiliza quatro bibliotecas distintas e diversas funções e ponteiros localizados dentro da biblioteca `funcoes_jogo`. Nela, encontram-se todos os códigos utilizados.

Localizada dentro da pasta `code`, utiliza-se o código `main` para inserir duas funções. A `startmenu`, que tem como finalidade abrir a tela inicial do jogo, proporcionar a cor preta no fundo de tela, a cor vermelha empregada na alteração de cor das setas, e as cores azul, verde e branca na mudança dos emojis apresentados conforme os movimentos são executados corretamente. A função `gameview` tem como principal utilidade a formação do código, as strings (setas), emojis, ponteiros, `if`, `while` e o chamamento das demais funções utilizadas no decorrer da criação do jogo.

Existem algumas funções aplicadas e chamadas pela função descrita anteriormente, como a `play_sound` que emprega o `play` para a inserção de música de fundo, o qual está situado dentro da pasta `SOX` que serve como ferramenta para a linha de comando. A função `lose_winview` que tem por objetivo auxiliar na melhor funcionalidade do jogo, permitindo que ao chegar em determinada pontuação o usuário vença ou no caso de o mesmo errar a coordenada, ele perca.

A maior biblioteca utilizada para execução do jogo é a `ncurses` que possibilita o uso de cores, a introdução de janelas

e o uso do teclado para efetuar os movimentos necessários. A biblioteca **stdio** que serve para entrada e saída do teclado. A **stdlib** tem como finalidade acessar pastas e mover diretórios internos. E, utilizando a biblioteca **time**, pôde-se acrescentar tempo para as jogadas ocorrerem.

O gerenciamento lógico do jogo consiste principalmente em setas e funções que realizam a diferenciação de cores das mesmas para o vermelho e são representadas pelas teclas W, S, A e D. O jogador tem um tempo máximo de até 5s para realizar uma próxima atividade, assim, se o mesmo se ausentar por mais tempo o jogo se encerra e uma nova partida precisa ser iniciada.

Ao atingir determinada pontuação durante uma sequência de acertos na partida, o jogador automaticamente é transferido para um nível superior e passa a receber pontuações maiores, o que é gerenciado pelo ponteiro ***prt_nivel**.

Para que haja um vencedor ou um perdedor durante a partida adicionou-se o ponteiro ***prt_pontos**, que possibilita as diferentes pontuações serem atribuídas para cada movimento feito pelo usuário, assim, seu *score* varia de acordo com seus acertos, erros e mudanças de níveis.

III. RESULTADOS

O template inicial é formado pelas explicações de como o jogo funciona, diversos caracteres formando o nome do mesmo e o play, como pode ser observado na figura 2.

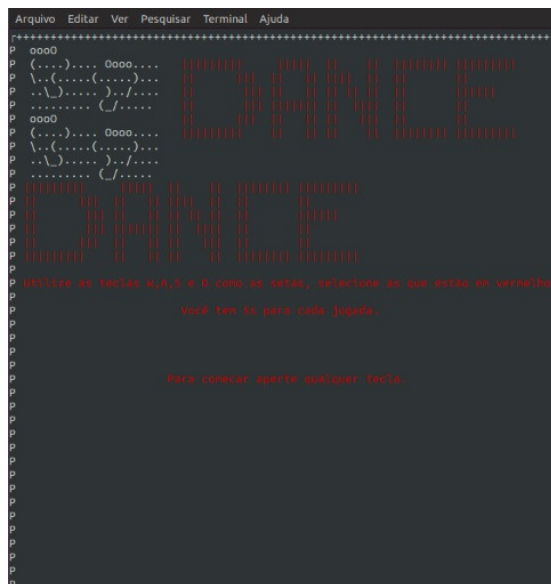


Figura 2. Template inicial.

A parte gráfica do jogo é composta por setas: uma para cima (W), uma para baixo (S), uma para esquerda (A) e outra para direita (D) alocadas em uma única coluna, como pode ser observado na figura 3.

O objetivo era colocar todas em forma de teclado, alinhando a seta direita com a esquerda, o que não pode ser alcançado devido a um movimento incompreendido da string, o qual movia apenas o início da mesma e o restante ficava no mesmo lugar.

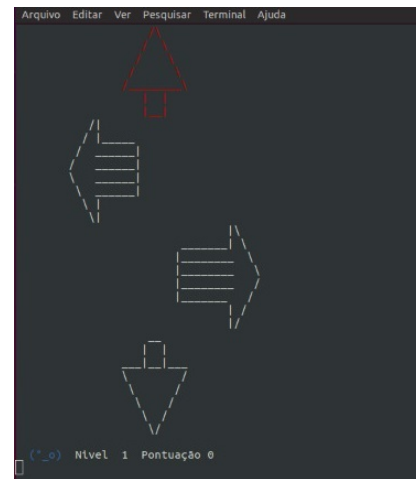


Figura 3. Formação gráfica.

Os resultados esperados após a formação do jogo são de aproveitamento dos usuários e entendimento de todas as posições atribuídas. Juntamente com o funcionamento de todas as partes ao mesmo tempo, como as setas trocando de cor para identificação, a música tocando em tempo correto e as pontuações geradas adequadamente.

IV. CONCLUSÃO

Conforme o exposto, pode-se concluir que o objetivo principal do jogo, de ter todas as peças funcionando corretamente e ao mesmo tempo, foram alcançados. Com o desenvolvimento deste projeto foi possível ter um maior entendimento da linguagem C e colocar em prática todo o assunto verificado em teoria na sala de aula. Para projetos futuros, o que não foi possível implementar neste, teríamos um personagem executando os movimentos ao invés de flechas de indicações para o movimento.

APÊNDICE A MÚSICAS UTILIZADAS

Up in my jam.
<<https://www.audiolibrary.com.co/kubbi/up-in-my-jam-all-of-a-sudden>. Acesso em: Junho de 2019.>
Family crownd celebration.
<<https://www.youtube.com/audiolibrary/soundeffects?ar=3>. Acesso em: Junho de 2019.>
Leszek_Szary
<https://freesound.org/people/Leszek_Szary/sounds/171672/. Acesso em: Junho de 2019.>

REFERÊNCIAS

- [1] dos Santos, L. B., *Utilizando a biblioteca NCURSES*, Rio de Janeiro: Viva o Linux, 2003. pp. 4.
- [2] Pinheiro, F. A. C., 2012. *Elementos de programação em C*. Porto Alegre: Bookman, 2012. 978-85-407-0202-8.
- [3] da Silva, W. B. 2019. In: *CPF*, Curitiba. <<https://github.com/wyllianbs/CPF>>.