

Columns/Magic Jewelry feito em Haskell

Nome do Aluno: Gustavo Hübner Rehbein

Matrícula: 18100021

Cadeira: TÓPICOS ESPECIAIS EM COMPUTAÇÃO IX - 2020/1

O jogo feito é baseado nos jogos [Columns](#) e [Magic Jewelry \(NES\)](#), com mecânicas semelhantes ao Tetris, o objetivo do jogo é formar combinações de 3 ou mais blocos em uma direção (vertical, horizontal e diagonais). O jogador controla uma peça com 3 blocos de cores aleatórias pelo tabuleiro, também é possível alternar a ordem das cores na peça.

Ao formar combinações, o jogo removerá os blocos que formam as combinações e aplica gravidade no tabuleiro. Novas combinações podem ser formadas ocasionando reações em cadeia. Quando não houver mais combinações, uma nova peça será adicionada ao tabuleiro. Ao avançar de nível uma peça bônus é adicionada, que removerá todos os blocos com a mesma cor daquele que ficar abaixo da peça bônus.



Figura 1 - Aparência gráfica do jogo.

Para executar o jogo, é necessário instalar as seguintes dependências:

- System.Console.ANSI - cabal install ansi-terminal
- System.Random - cabal install random
- Data.List.Unique - cabal install Unique

As demais dependências devem vir com a instalação padrão do Haskell ou como uma dependência dos pacotes instalados acima. Caso ocorra algum erro aqui está uma lista completa das importações:

```
import System.Console.ANSI (setSGR, clearScreen, hideCursor,  
setCursorPosition, showCursor, xterm6LevelRGB)  
import System.Console.ANSI (SGR(Reset, SetPaletteColor),  
ConsoleLayer(Foreground))
```

```
import System.Random (randomRIO)
import System.IO
import Control.Concurrent (threadDelay)
import Data.List.Unique (sortUniq)
import Data.Colour.SRGB (sRGB24)
import Data.Word
```

Para executar o jogo basta executar o arquivo *main.hs* com o GHCi ou GHC, no GHCi a função *main* inicia o jogo. Os controles do jogo são os seguintes:

- **A:** Move a peça para a esquerda.
- **D:** Move a peça para a direita.
- **S:** Move a peça para baixo.
- **Espaço:** Alterna a ordem das cores da peça.

Executando no Windows com GHCi

Devido a uma diferença da função *getChar* do Haskell no Linux e no Windows, o jogo do arquivo *main.hs* não funcionará corretamente no Windows. Onde o *getChar* (usado para obter o *Input* do jogador) só retorna o valor após a tecla Enter ser pressionada, trancando a execução do código caso contrário. Mesmo usando *hReady stdin* para testar se havia algo no buffer, no momento que *getChar* é chamado ele trava a execução do código, mesmo com mais de um caractere digitado, resumindo apenas após a tecla Enter ser pressionada.

Uma versão antiga com interatividade reduzida está presente no arquivo *main-windows.hs*. Para executá-la é necessário instalar outra dependência, que remove a necessidade de apertar Enter entre os comandos, mas trava a execução do código até que algum caractere seja inserido:

- `System.IO.NoBufferingWorkaround` - cabal install no-buffering-workaround

Essa versão usa os seguintes controles:

- **1 .. 6:** Escolhe a coluna para adicionar a peça.
- **Espaço:** Alterna a ordem das cores da peça.

O jogo foi inicialmente desenvolvido no Windows mas devido a essa limitação, foi migrado para o Linux e desenvolvido a versão mais interativa presente no início desse documento.

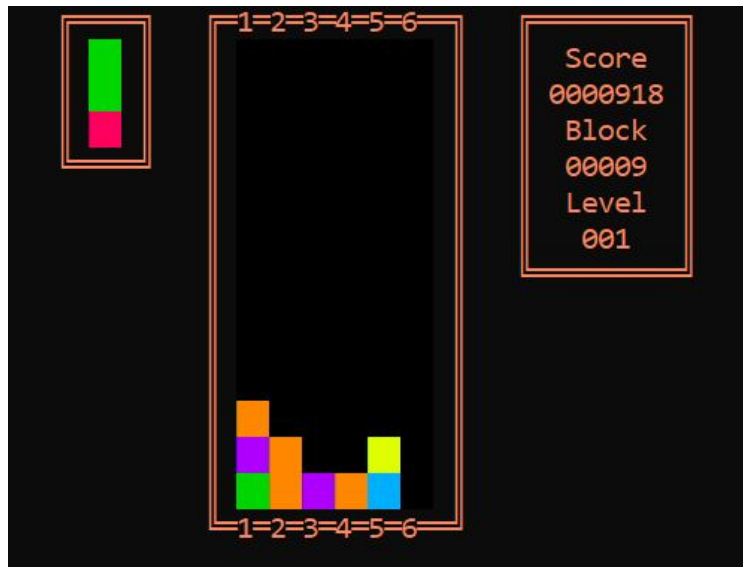


Figura 2 - Versão com menos interatividade no Windows.

Obs: Em alguns terminais a interface é composta por vários “??” devido a incompatibilidade com *Extended ASCII*.

Referências:

Columns (video game), Wikipedia. Disponível em:
[https://en.wikipedia.org/wiki/Columns_\(video_game\)](https://en.wikipedia.org/wiki/Columns_(video_game))

Magic Jewelry, Wikipedia. Disponível em:
https://en.wikipedia.org/wiki/Magic_Jewelry

Magic Jewelry, A guide by Lagoona. Disponível em:
<https://gamefaqs.gamespot.com/nas/935394-magic-jewelry/faqs/48284>

hSetBuffering stdin NoBuffering doesn't work on Windows. Disponível em:
<https://gitlab.haskell.org/ghc/ghc/-/issues/2189>