Departamento de Ciência da Computação - UFJF

DCC019 - Linguagens de Programação

Relatório do 2º Trabalho Prático

Oromar Voit de Rezende - 201765122C

Janeiro / 2022

Sumário

1 Introdução

O projeto consiste no desenvolvimento de uma aplicação, a ser utilizada no terminal, do jogo Nim. A aplicação terá dois níveis de dificuldade, explicados a seguir:

1.1 Modo Fácil

No modo fácil, o jogador começa, ou seja, é o primeiro a tomar uma ação.

Nas jogadas do computador, um número aleatório de palitos é retirado de uma linha aleatória. O computador faz apenas lances válidos, ou seja, ao menos um palido será retirado de uma linha.

1.2 Modo Difícil

No modo de jogo difícil, quem começa é o computador. Isso é importante, pois nesta formulação do jogo, existe uma estratégia vencedora para o jogador que não começa.

Ou seja, caso o usuário adote uma estratégia vencedora ele irá ganhar o jogo. Caso o usuário erre ao menos um dos lances, o computador irá ganhar o jogo.

2 Desenvolvimento

O programa foi desenvolvido utilizando uma estratégia incremental. Como se trata de um jogo, o mais importante são as interações com o usuário.

A primeira etapa foi implementar um sistema de menu. Na interface inicial, um título aparece para um usuário e um conjunto de opções enumeradas. O usuário pode digitar a opção escolhida. Porém, nesta versão inicial, existe apenas a opção de sair.

Nos próximos passos, uma pequena interação era adicionada e o código refatorado. Alguns exemplos de interação são:

- Tratamento de opção inválida
- Adição de modo que o jogador possa jogar consigo mesmo
- Tratamento de erros nas jogadas
- Adição de jogadas aleatórias

- Garantia de que as jogadas aleatórias serão válidas
- ...

2.1 Decisões de projeto

Dados os requisitos iniciais, apenas uma decisão deve ser explicitada aqui: **o que fazer quando a estratégia vencedora não estiver disponível para o computador?**

Como a estratégia vencedora começa de posse do usuário, este pode tomar decisões corretas e mantê-la consigo. Dessa maneira, não será possível ao computador realizar a melhor jogada, pois qualquer uma delas leva, empiricamente, à derrota.

Então qual jogada deve ser feita? A decisão aqui é tentar adiar, ao máximo, a vitória do usuário, na esperança de que este cometa um erro. A estratégia adotada nestes casos é retirar o mínimo de palitos (apenas uma unidade) da linha que contém mais palitos disponíveis.

2.2 Representação do Jogo

O estado do jogo é representado por uma lista de 4 posições. Cada posição é uma *linha* e o valor de cada posição é o número de *palitos* naquela *linha*.

3 Executando a Aplicação

O programa foi testado utilizando o Linux Mint 20.2 Uma, e o compilador/interpretador utilizado foi The Glorious Glasgow Haskell Compilation System, version 8.6.5.

3.1 Utilizando o compilador

Para compilar o programa, basta executar o seguinte comando na pasta que contém o código:

Listing 1. Compilar o programa

```
$ ghc Main.hs
```

Após isso, um executável de nome **Main** será criado. Basta executá-lo para utilizar o programa.

3.2 Utilizando o interpretador

Para interpretar o programa através do **ghci** basta carregar o arquivo *Main.hs* e executar a função *main*.

Um exemplo pode ser:

Listing 2. Interpretar o programa

```
$ ghci Main.hs
GHCi, version 8.6.5: http://www.haskell.org/ghc/  :? for help
[1 of 3] Compiling Game             ( Game.hs, interpreted )
[2 of 3] Compiling Menu             ( Menu.hs, interpreted )
[3 of 3] Compiling Main             ( Main.hs, interpreted )
Ok, three modules loaded.
*Main> main
```