

Universidade Federal de Juiz de Fora

Departamento de Ciência da Computação

Linguagem de Programação

CriptoGame

Daniel Machado Barbosa Delgado – 201835013

Gabriel Bronte Cardoso – 201835002

Professor: Leonardo Vieira dos Santos Reis

Juiz de Fora

Julho - 2022

Conteúdo

1	Organização do projeto	1
2	Principais funções	1
3	Estratégia de implementação	2
4	Como executar	2

1 Organização do projeto

Fizemos a separação do código em quatro arquivos:

- Main.hs - Arquivo principal responsável por iniciar o jogo e interagir com os outros arquivos.
- CompActions.hs - Responsável por realizar os cálculos para garantia do funcionamento do jogo.
- PrintParse.hs - Responsável por analisar e validar a interação com o usuário
- Types.hs - Responsável por armazenar os novos tipos criados e seus dados.

2 Principais funções

Quando o jogo começar as principais funções acessadas pelo sistema serão:

- getRandomCodes - Função responsável por retornar o código secreto.
- parseCodeGuess - Função responsável por analisar uma string que consiste em letras separadas por espaço que representam os códigos.
- getFeedback - Função responsável por comparar um palpite com o código secreto e retornar o feedback com quantidade de acertos completos e parciais.
- calcGame - Função responsável por retornar uma 3-tupla com a quantidade de dígitos corretos nas posições corretas no palpite digitado pelo decifrador, o palpite e o código secreto sem os elementos dos acertos completos
- winnerCondition - Função responsável por retornar se o palpite corresponde ao código secreto.

3 Estratégia de implementação

Em um primeiro momento, implementamos a função para gerar um número aleatório para que seja possível criar a chave secreta. Posteriormente definimos novos tipos de dados para podermos realizarmos o mapeamento da entrada do usuário e o cada um dos códigos da chave secreta. Para calcular a pontuação do jogador, utilizamos a notação head-tail para verificar se o elemento que está em ambas as head (do palpite e da chave secreta) são iguais. Em caso afirmativo incrementaremos 1 em nosso contador e retiramos esse elemento do palpite e código-chave e prosseguimos com o processo de recursão. Ao final deste processo, teremos um contador que resultará nos acertos completos, o palpite e o código-chave sem os acertos completos. Por fim, o resultado dos acertos parciais será a interseção do palpite e código-chave gerado anteriormente. Dessa maneira, conseguimos dar o feedback para o usuário logo após cada tentativa. Consideramos que o jogo acaba assim que todos os elementos são completamente correspondentes.

4 Como executar

Para executar o programa basta seguir os seguintes passos:

- É necessário ter o cabal instalado no seu computador.
- Abra a pasta do projeto e instale as dependências do Cabal executando `cabal install` no terminal
- Navegue até a pasta `src` e execute o compilador `ghci`
- Com o compilador ligado carregue o arquivo `Main.hs`
- Para começar a jogar basta executar a função `main` no terminal
- Use as entradas com o seguinte padrão: `1 2 3 4`

Obs: é preciso ter o `ghci` e o `cabal` instalado no computador.