

# MAC0216 - Técnicas de Programação I

## Relatório - Aventura!

Evandro Nakayama Mota - NUSP: 10737230  
Gabriel Brandão de Almeida - NUSP: 10737182  
Leonardo Alves Pereira - NUSP: 10737199  
Prof. Marco Dimas Gubitoso

Novembro de 2018

### 1 Primeira fase

O programa foi implementado em módulos, responsáveis por controlar a tabela de símbolos (hash table) e as listas ligadas. Para inserir elementos na tabela utiliza-se a função de hash que retorna um índice na lista a partir da soma dos valores em ASCII de cada caracter da string 'chave' módulo tam, onde tam é o tamanho da tabela.

O tratamento de colisões na inserção é feito através da lista ligada, assim, cada posição da tabela pode possuir uma lista de elementos.

Vale ressaltar que a função `retiraL`, que remove um elo na lista ligada, foi implementada de forma a retirar o elo seguinte àquele que recebe como argumento.

Para realizar os testes, fizemos um programa de simulação que verifica o funcionamento das funções definidas. Para tanto, foi implementada uma função `criaElemento` que recebe uma string como argumento e retorna um ponteiro para um `Elemento`, os quais são inseridos, buscados e retirados da tabela de símbolos. Todas essas verificações são impressas na tela com mensagens de sucesso ou falha.

### 2 Segunda fase

O *Makefile* está em estruturas/

Os cabeçalhos estão em headers/. Foi utilizado a biblioteca `<string.h>` para alguns arquivos para auxiliar na manipulação de strings.

O arquivo de testes é o `testes.c`. Existe um verbo específico `tentarSenha` que foge do padrão do tipo de função (`func`) que possui como parâmetros (`Elemento *e1`, `Elemento *e2`), pois era necessário passar uma string como parâmetros, portanto, na execução do `./testes` há *warnings* relacionados a isso.

Na segunda fase de desenvolvimento do jogo de aventura em texto, introduzimos a história que define a narrativa do personagem:

"Você é um estudante de matemática e computação. É domingo, 21h, você tem que entregar um EP até 24h, você decide jogar video game.

Grande fã de puzzles, você pega seu jogo favorito Portal 2 e começa a jogar.

Você pega no sono e de repente acorda numa sala e uma voz robótica começa a te dar instruções."

Dadas as novas exigências do projeto, foram implementadas diversas funções e um aprimoramento do programa de testes para verificar o funcionamento do jogo. As funções e suas respectivas documentações encontram-se nos arquivos de cabeçalho (.h) e código (.c).

O programa de testes da fase 2 executa uma rotina, no qual o `getchar()` simula um comando, em que imprime na tela a introdução, uma legenda, o verbo TROCAR DE SALA, a descrição da sala, respeitando o requisito de que a descrição longa é mostrada apenas na primeira vez, quando o elemento não é conhecido. Em seguida, imprime a lista dos objetos incluídos nas salas (visíveis ou não), precedidos com artigos indefinidos correspondentes no primeiro comando e a lista de objetos na mochila do aventureiro, nos próximos comandos é mostrado os verbos de cada objeto na sala e na mochila do Aventureiro (por padrão é mostrado os verbos EXAMINAR, PEGAR e LARGAR, porém também mostra-se outros e ANIMAÇÃO quando há). No verbo específico tentarSenha é passado uma senha aleatória de um vetor de senhas possíveis. Para a transitividade = 2, testamos com os parâmetros iguais, ou seja,  $e1 = e2$ .

O ambiente de interação do jogador foi definido em um conjunto de 6 salas: uma sala principal e 5 salas com enigmas.

Na sala principal há 5 portas que levam as outras 5 salas e um robô que pergunta qual a senha (ANIMAÇÃO), esse é o enigma final que o jogador deve descobrir conforme for resolvendo os problemas de cada sala. A senha é "SENHA". Porém, as letras veem fora de ordem e quando o usuário digitar "HNESA" irá aparecer a mensagem "que coisa!! ", referência a uma pergunta da Prova 1 da disciplina no qual tinha que explicar um algoritmo em Assembly que a cada 4 caracteres de uma string, os caracteres eram virados.

Na primeira sala, há um enigma que pergunta "São 8h e ... ?" no relógio que está faltando o ponteiro dos minutos. O jogador deve pegar o ponteiro que está no chão e colocá-lo no lugar correto. O horário correto é 8 horas e 13 minutos, pois conforme as dicas: tem uma mensagem na parede com a dica de " $1+1=2$ " que está no começo da sequência de Fibonacci e temos uma concha em espiral que segue o número de ouro. No momento que o jogador digita 13, aparece uma mensagem dizendo que a letra "H" (parte da senha) está dentro da concha.

Na segunda sala, há um cofre que pergunta qual a senha, um livro com infinitas páginas e uma mensagem na parede com um código morse. Quando o jogador decifrar o código está escrito "ISSONAOECODIGOMORSE", a dica está no título do livro que diz "Existem apenas 10 tipos de pessoas no mundo", famosa frase que diz que existem apenas pessoas que conhecem binário e as que não conhecem (10 em binário é 2). Se o jogador pedir para examinar o livro de novo o "10" vira "-", sendo a dica, " $1 = -$ " e " $0 = .$ ", ou seja, o jogador deve traduzir o código morse para binário e depois para decimal, com esse número ele irá encontrar uma página no livro, assim o jogador encontra a segunda letra "N".

Na terceira sala, há uma grande bobina elétrica, uma galinha presa em uma gaiola, um saco com milho e uma barra de metal. O objetivo é conseguir o que está dentro do ovo de ferro da galinha. Para isso o jogador deve primeiramente alimentá-la, então a galinha botará

o ovo, porém impossível de alcançar. Logo, ele deve utilizar a bobina para magnetizar a barra de metal (cobalto) e assim utilizá-lo como ímã para atrair o ovo para fora da gaiola. Ao quebrar o mesmo, o jogador encontrará a terceira letra "E".

Na quarta sala, há uma mensagem na parede que diz: "há pressão demais por aqui". Há um poema sem o sétimo verso, alguns blocos numerados de 0 à 9 e uma balança. Além disso, a sala tem formato triangular. Os blocos que devem ser colocados sobre a balança são os blocos 1, 2 e 8, que juntos correspondem à soma da linha 7 (poema sem o sétimo verso) do triângulo de Pascal (unidade de pressão). Assim, o jogador recebe a quarta letra "S".

Na quinta sala, há um mapa que pergunta qual país está faltando e a dica é uma piadinha: "O que o cientista disse quando encontrou 2 átomos de hélio? HeHe", assim o jogador deve decifrar que cada objeto na sala corresponde a um elemento da tabela periódica, Poster de cidade futurista = Neon (Ne), caixa de fósforo = Fósforo (P) e lata de refrigerante = Alumínio (Al), portanto, o país é Nepal (Ne\_P\_Al), com isso, o jogador recebe a quinta letra "A".

### 3 Terceira Fase

Na etapa final do projeto desenvolvemos o interpretador de texto para o jogo de aventura em texto.

Os arquivos do flex e bison encontram-se em estruturas/.

O reconhecimento de verbos e elementos é realizado pelo flex enquanto o bison verifica se a entrada do usuário corresponde a alguma expressão ou comando configurado, indicando quando a frase não corresponde a nenhum deles ou então executando o comando correspondente, caso contrário. Foi adicionado o comando "definir <palavra>", para auxiliar a existência de tal palavra, devolve "É um verbo!" se for verbo, "É um substantivo", se for um objeto ou lugar e "Isso não existe!", caso contrário. Além disso, há, na introdução do jogo, instruções para auxiliar o jogador.

Para facilitar a verificação do funcionamento do programa há uma lista completa de comandos simulando o caminho para finalizar o jogo de um usuário no arquivo *comandos*.

Para compilar e executar com a simulação dos comandos: "make && ./avent <comandos"

Para compilar e executar o testes: "make testes && ./testes"