

Universidade Federal do Rio Grande do Norte
Escola de Ciências e Tecnologia

Lucas Otávio Vieira

Mastermind em C++

Natal – RN
Novembro - 2017

Lucas Otávio Vieira

Mastermind em C++

Relatório apresentado à Universidade Federal do Rio Grande do Norte, na disciplina de Lógica de Programação como requisito para avaliação na 3 unidade, solicitado pelo(s) professor(es) Einstein Santos e Rômulo Nunes.

Natal – RN

Novembro - 2017

SUMÁRIO

1. Introdução

1.1. Equipe de desenvolvimento

2. Desenvolvimento

2.1. Funcionalidades implementadas

2.2. Funcionalidades não concluídas

2.3. Cronograma das atividades realizadas

3. Conclusão

1. Introdução

O objetivo principal deste trabalho foi desenvolver o jogo Mastermind através da linguagem C++ incluindo todas as funcionalidades básicas sugeridas na instrução da atividade. Cada etapa do projeto consistiu na tentativa de implementação de cada uma dessas funcionalidades individualmente, a fim de facilitar a sua conclusão.

1.1. Equipe de desenvolvimento

Lucas Otávio Vieira, 20170041970, subturma 5C.

2. Desenvolvimento

2.1. Funcionalidades implementadas

O projeto teve início a partir da implementação do menu do jogo (tela inicial, tela de instruções e tela de nível de dificuldade), após receber orientação do monitor para deixar o menu por último e ter alguns problemas em encaixar o código do jogo na estrutura condicional switch case (ativava a flag fpermissive no compilador) decidi começar pela lógica e deixar o menu para depois.

Em relação ao desenvolvimento da lógica do jogo a primeira etapa foi a geração de uma senha aleatória sem repetição, já conhecia a função rand e após uma rápida pesquisa na internet consegui a sintaxe da mesma. Utilizar a função rand não foi difícil, o entrave aconteceu em relação à não repetição dos algarismos da senha (optei por utilizar uma senha composta por números por ainda não saber desenvolver uma interface gráfica e achar mais prático trabalhar assim), após algumas tentativas frustradas utilizando o laço for a solução surgiu num algoritmo apresentado pelo professor em sala de aula. Através de uma flag era possível ler um número inteiro digitado pelo usuário e só permitir a inserção desse inteiro num vetor específico se este já não estivesse no vetor. Utilizando essa lógica tentei adaptar para a função rand mas havia algo errado com o código, pois a repetição ainda ocorria.

Após visita à monitoria o problema foi identificado: o for utilizado para percorrer o vetor em busca de inteiros repetidos foi finalizado no local errado e

acabava não percorrendo todo o vetor, era necessário percorrê-lo por inteiro e comparar os valores para depois prosseguir.

Resolvida a questão da sequência randômica aleatória uma lógica ainda mais simples (sem a flag) foi utilizada para compor o código. O for percorre o vetor e não incrementa o contador caso o algarismo esteja repetido, dessa maneira, a estrutura só vai adiante no caso de não existir repetição. Concluída a geração da senha aleatória a próxima etapa era a marcação dos pinos brancos e pretos. Contadores foram utilizados para chegar ao número de pinos e alguma dificuldade surgiu em relação à exibição dessa informação numérica em forma de um vetor que mostrasse primeiro os pinos brancos e depois os pinos pretos. Após várias tentativas a solução surgiu. Primeiro, o vetor resultado era zerado, e depois preenchido com a quantidade de pinos brancos para então colocar os pretos quando o lugar já não estivesse ocupado pelos brancos e enquanto o número de pinos pretos fosse maior do que zero.

Essas foram as duas maiores dificuldades em relação à implementação da lógica do jogo, as demais etapas consistiram num trabalho “braçal” em encaixar a lógica do jogo no menu (dessa vez utilizando a estrutura condicional if), inserir uma pontuação, tratar o jogador pelo nome e adicionar o modo multiplayer. O modo multiplayer foi a parte mais extensa em virtude da funcionalidade dos níveis de dificuldade, que triplicou o código tanto no modo single player quanto no modo multiplayer. A questão do multiplayer era mais complexa porque o jogo rodava duas vezes e o resultado era comparado para determinar o vencedor, como o sistema de pontuação era bem simples e por rodadas o vencedor já seria quem fizesse mais pontos, pois a cada tentativa o número de pontos diminuía. A pontuação funciona de acordo com o número de tentativas do nível, para nível 3 o máximo é 140 (14 tentativas), nível 2, 120 (12 tentativas) e nível 1, 100 (10 tentativas). A cada rodada o jogador perde 10 pontos, caso o número de tentativas acabe ele fica com 0 pontos e perde, caso ele acerte de primeira ficará com a pontuação máxima.

Em relação ao menu do jogo a dificuldade foi o tratamento de exceções caso o usuário digitasse uma opção inválida ou nível inválido. A parte da opção funcionava com while, mas não consegui pensar numa maneira de implementar o while pros níveis a tempo e sem causar “bugs” no programa, por isso, o goto foi utilizado para tratar as exceções dos níveis. Outra questão foi se utilizava caractere ou inteiro para a escolha do menu, usando inteiro existe a vantagem de não permitir

que um número fora intervalo definido para as opções seja inserido e a desvantagem de “bugar” quando um caractere qualquer é digitado. Usando caractere existe a vantagem de tratar os casos em que se digita um caractere que não faz parte das opções e a desvantagem de permitir que números fora do intervalo definido para as opções sejam inseridos desde que o primeiro número pertença a esse intervalo (no caso, apenas o primeiro caractere é lido).

2.2. Funcionalidades não concluídas

As funcionalidades não concluídas foram as extras, a parte de interface gráfica não foi implementada por ainda não saber desenvolvê-la e por não ter afinidade com essa área. Já a questão da pontuação por tempo não foi desenvolvida justamente pela questão do manejo do tempo disponível para as atividades de cada disciplina do curso, como o trabalho foi feito individualmente acaba sobrando menos tempo para as outras disciplinas.

2.3. Cronograma das atividades realizadas

| Data | Autor | Descrição da atividade |
|---------------------|--------------|---|
| 04/11 | Lucas Vieira | Esboço do menu do jogo |
| 12/11 | Lucas Vieira | Esboço da lógica do jogo |
| 13/11 a 17/11 | Lucas Vieira | Correção da lógica do jogo e implementação das demais funcionalidades |

3. Conclusão

A princípio a atividade parecia ser bem complicada de ser realizada, no entanto, ao longo do processo dedicando um pouco de tempo a cada dia e trabalhando em cada etapa separadamente o nível de dificuldade se mostrou razoável e bem diferente do imaginado anteriormente. Em relação à importância dos conhecimentos obtidos foi muito importante o aprendizado da execução e planejamento de projetos, manejo de tempo e divisão de uma tarefa aparentemente complexa em etapas mais simples. Além da aplicação dos conceitos ensinados nas aulas e, principalmente, através das listas de exercícios que ajudaram bastante no desenvolvimento de raciocínio lógico para a resolução de problemas.

Provas convencionais realizadas no papel (a meu ver) não combinam com uma disciplina de lógica de programação. Este tipo de trabalho é muito mais vantajoso por estimular a criatividade e o aprendizado por tentativa e erro, além da pesquisa e a busca do conhecimento necessário para concluir a tarefa por meio de várias fontes, o acompanhamento e o apoio dos professores e monitores foram outro fator importante no desenvolvimento do trabalho.