

Implementação do Jogo 2048 em Linguagem de Programação C

Eduardo Luís Marques, Nathan Rodrigues de Brito

***Resumo.** Este relatório descreve a implementação do jogo 2048 em uma linguagem de programação (no caso, C) por dois alunos do curso de Ciência da Computação, na Universidade Federal do Rio Grande do Sul, utilizando dos conhecimentos aprendidos na disciplina de Algoritmos e Programação.*

1. Descrição do Trabalho Realizado

Nossa tarefa nesse trabalho foi de implementar o jogo 2048 na linguagem de programação C. O jogo consiste em um tabuleiro de 4x4 quadrados e o seu objetivo é juntar pares de peças de igual valor até ser obtida a peça com valor 2048.

O jogo inicia com duas peças com valor 2 colocadas em posições aleatórias do tabuleiro. Em cada jogada, todas as peças são deslocadas em uma direção no tabuleiro, usando as teclas direcionais (\leftarrow , \rightarrow , \downarrow e \uparrow), o que fará com que as peças se movam até o fim da linha ou coluna onde estão. Se o bloco encostar em outro com o mesmo número eles se somam e viram apenas um bloco de número maior. No final de cada jogada é colocada mais uma nova peça no tabuleiro, num lugar livre escolhido aleatoriamente, com valor 2 ou 4, também escolhido aleatoriamente em que a probabilidade de ser 4 é de 10%.

Só é possível jogar numa direção se alguma peça se deslocar ou se juntar a outra. O jogo termina quando não for possível jogar nas quatro direções ou for obtida a peça com o valor 2048.

O programa contabiliza o número de jogadas e a pontuação é aumentada por cada junção de peças na soma do valor das duas peças. Quando o jogo termina, é solicitado o nome do jogador caso a pontuação obtida seja melhor que uma das já registradas.

O jogo também termina quando é pressionada a tecla ESC para terminar o programa ou 'n' para começar um novo jogo. Nesse caso é solicitada a confirmação do jogador, e após isso a ação correspondente é executada e nesse caso a pontuação do jogador não é armazenada.

2. Representação dos Elementos do Jogo

Na inicialização do programa, temos um menu inicial com informações pertinentes às opções do usuário. No lado esquerdo, há um desenho estilizado do título do jogo, 2048. No lado direito, há uma descrição das instruções e opções disponíveis ao usuário.

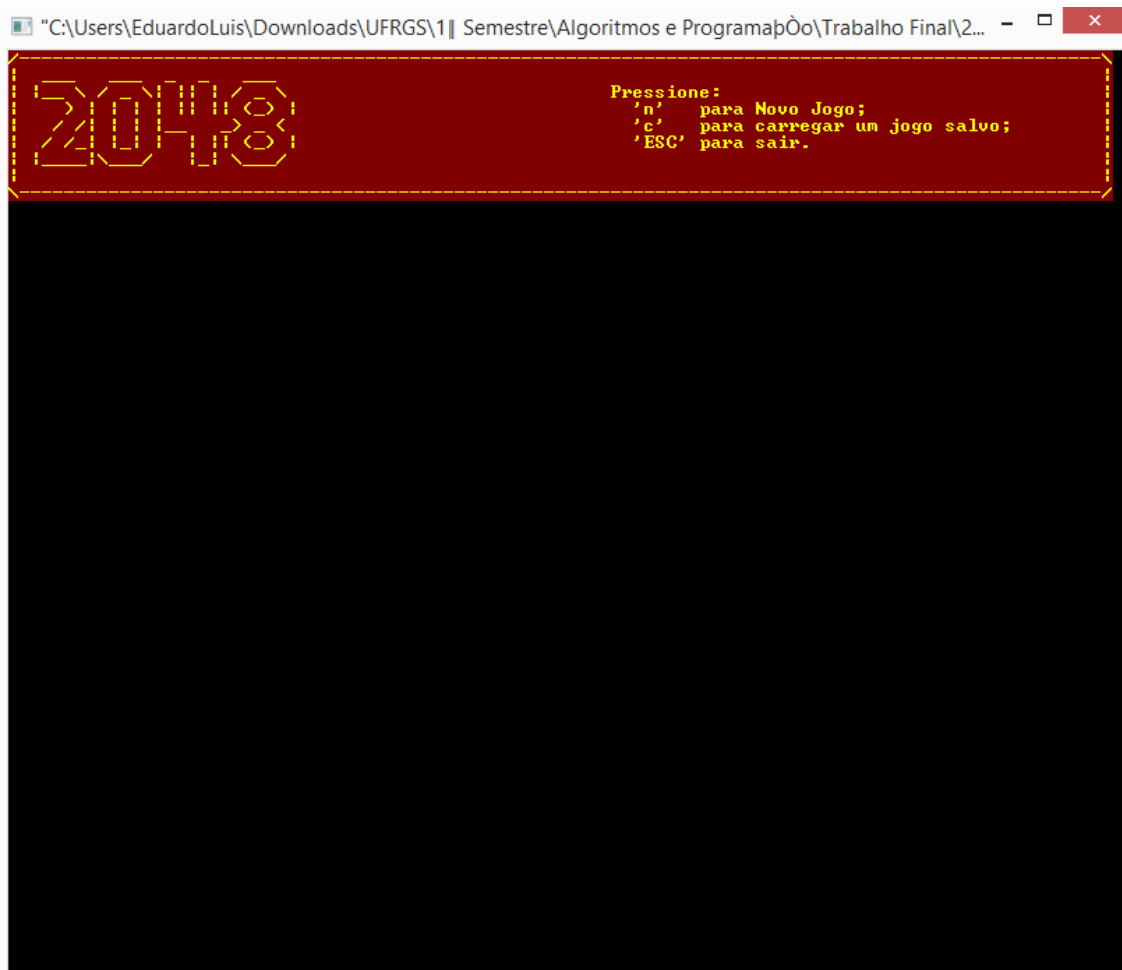


Figura 1. Menu Inicial do Jogo

Após o usuário decidir iniciar um novo jogo ou carregar um jogo salvo. Automaticamente abre a tela com o tabuleiro do jogo.

O tabuleiro foi desenhado através de um printf, na estrutura 4 colunas por 4 linhas, conforme especificação do jogo.

O tabuleiro tem suas linhas pretas e o fundo inteiramente pintado de amarelo para dar um contraste tanto com as linhas, tanto com os números gerados.

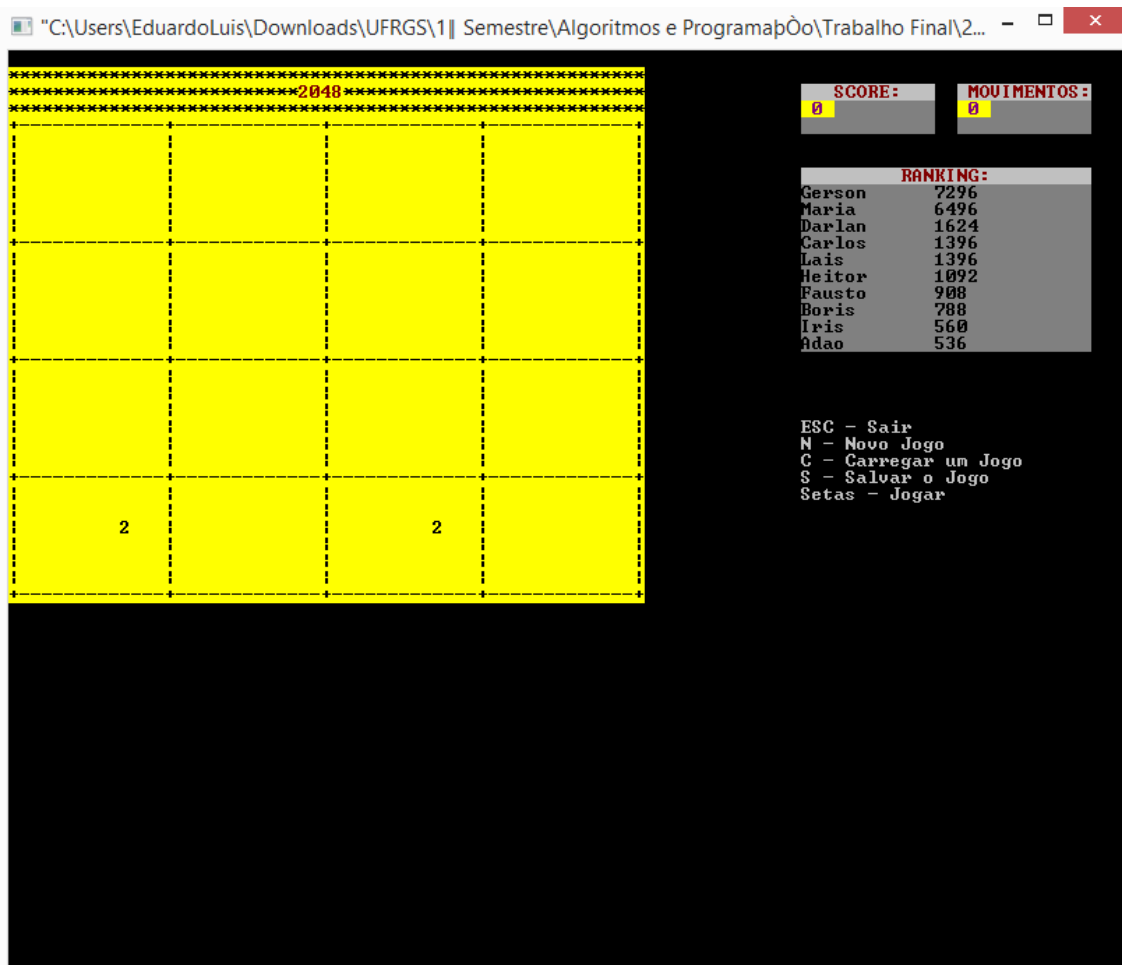


Figura 2. Tabuleiro do Jogo

Ao lado direito superior, primeiramente temos o score atual do jogador, atualizado a cada jogada. Ao lado do score se encontra o numero de movimentos realizado pelo jogador até o atual momento, sendo incrementado a cada jogada.

Abaixo disto, se encontra a tabela do ranking, expondo a lista dos 10 melhores jogadores de forma ordenada e suas respectivas pontuações. Ao final de cada jogo é atualizado esse ranking.

As peças representadas no tabuleiro são estruturas do tipo peca, possuindo como elementos um valor do tipo int, que representa o seu número, e uma cor do tipo COLOR, que representa a cor que seu valor será pintado. O tabuleiro, em si, é representado como uma matriz de ponteiros de elementos da estrutura peca, sendo eles impressos na representação gráfica do tabuleiro. Cada vez que um novo jogo é iniciado, os valores de todas as peças são definidos como zero, além de ser gerado duas peças de valor 2 em lugares aleatórios no tabuleiro.

Cada vez que uma tecla direcional é pressionada, é verificado se a movimentação para essa direção é possível. Caso seja, ocorre o deslocamento das peças, incrementando o placar de movimentos, e, se houver, a soma das peças que se juntam, incrementando o score. Após cada movimentação, é gerada um novo número, 2 ou 4,

em algum lugar aleatório disponível no tabuleiro. A probabilidade de gerar 2 é 90% e de gerar 4 é de 10%. Caso não seja possível se mover, nenhuma mudança acontece no tabuleiro.

No pressionamento das teclas de opção (ESC, 'n', 'c' ou 's'), é gerada uma tela de confirmação da opção escolhida. Nos casos de 'n' e ESC, apenas confirma com o usuário se ele realmente quer iniciar um novo jogo ou sair do atual. Caso pressione 's', a opção escolhida é feita. Caso pressione 'n', volta para o jogo atual. Nos casos de 'c' ou 's', é solicitado ao usuário o nome do arquivo de jogo que quer carregar ou salvar, respectivamente. Se não for encontrado nenhum arquivo com aquele nome para carregar, volta para o jogo atual. Após salvar o jogo atual com o nome do arquivo, também volta para o tabuleiro.

No fim do jogo, seja por falta de movimentos disponíveis ou por alcançar o número 2048, é impresso o resultado e, após um delay, o usuário é levado até uma tela de encerramento. Caso sua pontuação seja maior que as registradas no ranking, seu nome é solicitado para incluí-lo no ranking. Caso não seja, apenas é impressa uma mensagem de encerramento, o ranking e se encerra o jogo.

3. Interação dos Componentes Interativos

Aqui há uma descrição de cada funcionalidade do jogo e a lógica utilizada para implementá-la.

3.1. Menu Inicial

No início do programa, é definido a janela do sistema com 55 de altura e 100 de largura, para que o jogo melhor se adeque as dimensões. Após isso, definimos o fundo do que está escrito na tela como vermelho e a cor do texto como amarelo. Utilizamos comandos printf para imprimir o título do jogo e as opções para o usuário. Em um laço, verificamos o caractere digitado pelo usuário para definir a opção a ser seguida. Caso seja pressionado o ESC, imprime uma mensagem de encerramento e encerra o programa. Caso seja pressionado 'c', vai para a tela de carregamento de jogos e, se foi possível carregar um arquivo, inicia o jogo com as configurações do arquivo. Se não, volta para o menu e espera nova instrução. Caso seja pressionado 'n', cria um novo jogo do zero.

3.2. Novo Jogo

Assim como o menu, é esperado uma tecla de instrução do usuário para realizar algum movimento, sendo as opções ESC, 'n', 'c', 's' e as teclas direcionais. No pressionamento de uma das opções, é atualizado o tabuleiro, score e movimentos com as informações novas. Há duas variáveis que encerram o jogo. Uma que verifica se não há mais movimentos disponíveis e outra que verifica se o objetivo, 2048, foi alcançado. Caso alguma seja verdadeira, o usuário é levado para uma tela de encerramento, de acordo com o encerramento alcançado.

3.3. Imprime Tabuleiro

O tabuleiro foi impresso através de um printf, seguindo as especificações de 4 linhas por 4 colunas, utilizando a biblioteca conio para pintar o fundo do tabuleiro de amarelo e as linhas do tabuleiro de preto para um melhor contraste para visualização.

Para imprimir as peças no tabuleiro, é percorrida toda a matriz e as coordenadas x,y de cada casa correspondente no tabuleiro. Nessa coordenada, é impresso o valor da peça pintado com a sua cor.

Para imprimir o score e os movimentos, vamos até a coordenada correspondente no tabuleiro e imprimimos o valor.

3.4. Gera 2 ou 4

No início do jogo, para gerar as duas peças de valor 2 no tabuleiro, sorteamos uma linha e coluna aleatória da matriz e verificamos se seu valor é 0. Caso seja, seu valor passa a ser definido como 2. Se a posição sorteada não for vazia, verifica todas as outras posições e, caso há alguma vazia, sorteia outra linha e coluna.

Para gerar as peças de valor 2 ou 4, durante a execução do jogo, sorteamos também uma linha e coluna aleatória e fazemos a verificação de espaço vazio. Também calculamos uma probabilidade para definir o valor da peça, sendo de 10% para 4 e 90% para 2.

3.5. Movimentos

Verifica se há movimentação disponível por princípios bem parecidos entre si. Percorre a matriz na posição indicada e move as peças pelos espaços disponíveis, somando se for possível. Devolve uma flag para indicar se houve a movimentação.

3.6. Salvar ou Carregar

Faz a instrução de acordo com a opção escolhida (salvar ou carregar). Pede o nome do arquivo ao usuário, salva com extensão .bin e salva/carrega. Na opção de salvar, escreve no arquivo o score, os movimentos e o tabuleiro, imprimindo uma mensagem caso haja sucesso e voltando para o jogo. Na opção de carregar, lê o score, os movimentos e o tabuleiro no arquivo solicitado pelo usuário e imprime o tabuleiro com essas informações. Se não há arquivo com o nome digitado pelo usuário, imprime uma mensagem de erro e volta para a tela anterior.

3.7. Encerramentos

Mostra mensagens de confirmação para o usuário caso ele realmente queira sair do jogo atual ou iniciar um novo jogo, definido de acordo com a opção digitada.

Para verificar se houve derrota no jogo, percorre toda a matriz para ver se há algum espaço com valor vazio, retorna 1 se não há.

Ao final de cada partida, o jogador é colocado na última posição do vetor, caso entre no TOP 10, e é realizado um ordenamento bolha para formar um novo ranking atualizado. Mesmo que o jogador não obtenha score suficiente para entrar no ranking TOP 10, é realizado uma verificação de ordenamento.

Quando se chega no fim do jogo, imprime uma mensagem de acordo com a forma que chegou nele, seja por falta de movimentos ou por alcançar o 2048.

E, também, é gerado o ranking atualizado, já com a integração do score obtido pelo jogador atual e na posição que ele ocupa.

4. Estruturas e Funções Utilizadas

4.1. Estruturas

peca: estrutura para representar as peças do jogo. Possui a variável int valor, representando o valor numérico da peça, e a variável COLORS cor, representando a cor que será pintado esse valor.

jogador: estrutura para representar os jogadores salvos no ranking. Possui a variável string nome, de tamanho 10, que representa o nome do jogador, e a variável long int score, representando a pontuação do jogador.

4.2. Funções

novo_jogo(): função que lida com o funcionamento do jogo em si. Recebe como parâmetro o tabuleiro, uma flag para indicar se o jogo é novo e, caso sim, zera as peças do tabuleiro, e duas variáveis long int para representar o score do jogador e seus movimentos. Chama as funções para imprimir as representações visuais do jogo e define se chegamos no fim ou não.

gera2(): função que gera as duas peças 2 iniciais no tabuleiro.

gera2ou4(): função que gera uma peça 2 ou 4 em um local aleatório do tabuleiro a cada movimentação no jogo.

imprime_tabuleiro(): imprime um tabuleiro 4x4 com fundo amarelo e linhas pretas.

imprime_numeros(): imprime as peças no tabuleiro de acordo com a posição da coordenada (x,y).

imprime_score(): imprime o score no tabuleiro.

imprime_movimentos(): imprime os movimentos no tabuleiro.

imprime_ranking(): imprime o ranking no tabuleiro.

cores(): define uma cor pra peça de acordo com seu valor. Para todos os valores disponíveis até 2048, há uma cor especificada.

pode_cima(): verifica se há movimentação para cima. Percorre todas as colunas, a partir da linha 1, deslocando as peças para os espaços vazios e somando-as com seus valores iguais encontrados no caminho.

pode_baixo(): verifica se há movimentação para baixo. Percorre todas as colunas, a partir da linha 2, deslocando as peças para os espaços vazios e somando-as com seus valores iguais encontrados no caminho.

pode_esquerda(): verifica se há movimentação para esquerda. Percorre todas as linhas, a partir da coluna 1, deslocando as peças para os espaços vazios e somando-as com seus valores iguais encontrados no caminho.

pode_direita(): verifica se há movimentação para direita. Percorre todas as linhas, a partir da coluna 2, deslocando as peças para os espaços vazios e somando-as com seus valores iguais encontrados no caminho.

salvar_ou_carregar(): função para salvar o jogo atual ou carregar um jogo salvo.

confirma_novo(): função para confirmar o início de um novo jogo.

confirma_saida(): função para confirmar a saída do jogo atual.

derrota(): percorre toda a matriz e, caso não tenha mais espaços disponíveis, retorna o valor 1, indicando essa situação.

ordenamento(): faz um ordenamento bolha da lista de jogadores no ranking.

fim_de_jogo(): função para imprimir mensagens de encerramento e atualizar o ranking, caso o jogador atual tenha atingido uma pontuação melhor que o último colocado.

5. Como Usar o Programa

Após rodar o programa, no menu inicial há as opções de iniciar um novo jogo, carregar um jogo salvo, ou sair para o usuário. Pressionando ESC, você encerra o programa e terá que rodar de novo. Pressionando ‘c’, você pode digitar o nome de um arquivo (sem a extensão) que o programa irá pesquisar se há um jogo salvo com esse nome. Caso não tenha, uma mensagem de erro será mostrada e o programa retornará para o menu. Se houver, o jogo salvo será carregado. Pressionando ‘n’, você inicia um novo jogo, onde será impresso o tabuleiro na tela.

No tabuleiro, você verá duas peças 2 impressas em locais aleatórios. Com as teclas direcionais, você pode movê-las para as respectivas direções, caso não seja o fim da matriz, e verificar o aumento da tabela de “movimentos”. Se você juntar duas peças de valores iguais, quando fizer uma movimentação, o placar de “score” será aumentado com essa soma e a peça nova será substituída no tabuleiro. Você pode ir jogando da maneira como preferir, sempre se atentando para os limites de movimento. Caso atinja uma situação onde não há mais movimentos disponíveis ou caso ganhe o jogo, atingindo a peça 2048, o programa para e uma mensagem é impressa na tela, seja de derrota ou vitória. Após um tempo de espera, você irá para uma tela de encerramento.

Se sua pontuação for maior que o último colocado no ranking, o programa irá solicitar seu nome para incluí-lo. Você deve digitá-lo e pressionar a tecla Enter quando terminar. Após, uma mensagem de agradecimento é impressa junto com o ranking atualizado e o programa encerra. Se sua pontuação for menor, apenas a mensagem e o ranking são impressos e o programa é encerrado.

Referências

Schnorr, L. e Pessutto, C., R., L (2021), Disciplina de Algoritmos e Programação, Brasil.