RELATÓRIO DE ICC - TRABALHO 5

Membros do grupo:
Luigi Quaglio - 11800563
Vinicius Rodrigues Geraldo - 10438122
Gabriel Sotto Rodrigues - 11800952

INTRODUÇÃO

Qwirkle é um jogo baseado blocos que assumem uma combinação de 6 cores e 6 formas diferentes 3 vezes para cada forma possível, totalizando 108 peças. Para vencer, deve-se ser alinhados seguindo uma lista de regras e para cada peça colocada em jogo, uma determinada quantidade de pontos e dada ao jogador que a colocou. Ao final, aquele com mais pontos vence.

Substituindo as cores por letras e as formas por números, criou-se uma versão do jogo desenvolvida totalmente em C para compor a nota do último trabalho da disciplina de Introdução a Ciência de Computação.

SISTEMA

Dividiu-se o código entre 8 arquivos, sendo todos interligados pelo "**libqwirkle.h**". Exceto o main.c, cada um dos outros era responsável por definir uma das funções necessárias para a execução do programa. Para cada função do programa, existe um arquivo C de mesmo nome no qual a função está. A funções do sistema são:

- **aumentatab():**
  Recebe um ponteiro para a dimensão atual do tabuleiro, que assume sempre o tamanho de um quadrado NxN, e um ponteiro para o tabuleiro em si. Cria um novo tabuleiro de dimensão (N+2)*(N+2) e preenche-o com as peças do tabuleiro em suas devidas posições. Desaloca o antigo tabuleiro e retorna o novo.

- **tabu():**
  Recebe um ponteiro para a dimensão atual do tabuleiro e um ponteiro para o tabuleiro em si. Imprime o tabuleiro com suas coordenadas e peças.
- **trocar():**
  Recebe um ponteiro para as peças que podem ser compradas, um ponteiro para as peças do jogador e um inteiro indicando quem está jogando. Pergunta a quantidade de peças a serem trocadas e depois pergunta uma a uma qual a peça a ser trocada, verificando se é uma peça da que está na mão do jogador. Se sim, retira a peça da mão do jogador e coloca de volta nas peças que podem ser compradas e, em seguida, sorteia uma das peças a ser trocadas e coloca no lugar da anterior. Se não, pede para colocar uma peça válida.
- **jogar():**
  Recebe um inteiro indicando quem está jogando, um ponteiro para as peças do jogador, a dimensão do tabuleiro, o tabuleiro, as peças que podem ser compradas, um ponteiro para a pontuação do jogador, quantas vezes já foram jogadas nessa rodada, ponteiros para a última posição jogada e uma flag que indica em qual dos eixos é permitido jogar novamente. Em uma primeira jogada, é permitido jogar em qualquer lugar, desde que a peça a ser jogada seja uma peça do jogador e o lugar a ser jogado está vazio e próximo de uma sequência de peças que validam a jogada. Na segunda jogada que o jogador fizer em uma única rodada, será possível jogar as peças em suas mãos apenas em uma mesma linha ou coluna que a peça anterior, de forma que elas permaneçam em uma mesma sequência de peças. Para a terceira jogada em diante, a flag de eixo bloqueia qualquer jogada que seja feita em uma linha ou coluna diferente das duas jogadas anterior, sendo que essa jogada também obriga as peças a permanecerem conectadas. Ao final de cada jogada feita, a função **contapont()** é chamada para acrescentar os pontos obtidos pelo jogador após aquela jogada.
- **contapont():**
  Recebe como parâmetro **x**, que é a coluna na qual o jogador irá colocar a peça , **y** onde é armazenada a linha em que o jogador

jogará, posição é o conjunto **(y,x)** que representa a casa da jogada ,um ponteiro para **pontos**, para que armazena os pontos do jogador, **tam,** que corresponde a dimensão atual do tabuleiro, um ponteiro para o **tabuleiro**, **ordem,** indicando qual jogador está jogando, e **times** para contar a quantidade de vezes que o jogador realizou uma jogada na rodada.

- **jogarch():**
  Recebe os mesmo parâmetros da função jogar, com exceção do *jogador, pois esse ponteiro era para saber quais peças que o jogador tinha, mas, no modo cheat, o mesmo pode jogar qualquer peça do jogo. Nela, acontece o mesmo que em **jogar()**, porém sem a verificação das peças que estão na mão do jogador, visto que não importa isso no modo cheat.
- **ganhador():**
  Essa função é responsável por verificar quem ganhou a partida e recebe como parâmetro **njog**, onde contém o número de jogadores na partida , um ponteiro para **pontos**, para obter a quantidade de pontos que os jogadores tem e um ponteiro para ponteiro para **nomes**, onde obtém o nome dos jogadores da partida.
- **main():**
  Esta é a função principal onde se encontra o menu do jogo. Ao iniciar o programa, esta função pergunta o número total de jogadores **njog**. Após isso, é separada memória para guardar os pontos de cada jogador no ponteiro ***pontos**, mais um bloco de memória é alocado para guardar os nomes de cada jogador que são declarados pelo usuário ****nomes**. Somado a isso, todas as peças do jogo são feitas e armazenadas na memória pelo ponteiro ***pecas**. Nesta função é possível ativar o modo cheat, o qual permite a utilização de qualquer peça do jogo. O último bloco de memória é alocado nesta função tem como objetivo armazenar as peças de cada jogador ***jogador**, cada peça é composta por dois caracteres letra e número, desta forma, o vetor jogador deve ter espaço o suficiente para guardar 12 caracteres para cada jogador (6 peças). Em seguida as peças de cada jogador são distribuídas e substituídas por ‘\0’ no vetor peças (saco de peças), se o modo cheat estiver

ativado as peças nao sao substituidas inicialmente, pois não os jogadores podem pegar qualquer peça do jogo, incluindo as dos adversários. Finalmente aparece o menu do jogo, o primeiro jogador a jogar, suas peças e pontos são exibidos, além das opções jogar, passar a vez ou trocar de peças. Ao escolher a opção "jogar" a função **jogarch()** ou a **jogar()** é chamada, sendo a primeira quando está em modo cheat e a segunda em modo normal. Uma flag recebe o retorno destas funções. Se ela for '-1' (houver uma peça na borda do tabuleiro) a função **aumentatab()** é executada. Para o caso da opção troca de peças, a função **troca()** é executada. Após isso passa a vez do jogador automaticamente. A última opção simplesmente incrementar o valor da variável ordem (responsável por marcar quem está jogando). Após todas as jogadas serem feitas, ou seja, não haver mais peças disponíveis, o looping se encerra e é chamada a função **ganhador()**, exibindo então, o vencedor do jogo.

LINKS RELACIONADOS

repositório git do programa:
https://github.com/Luigi1606/Qwirkle

vídeo explicando o código:
https://drive.google.com/file/d/1MnbXbWZJHzOoaj3S1Hi4Xz1UfBHYPbzS/view?usp=sharing

vídeo da execução:
https://drive.google.com/file/d/1XLUeR1fi1aVySDNLiAy-R6TyyAxrVXZM/view?usp=sharing