

Projeto da Disciplina Introdução à Programação



Implementação de boop na linguagem C

Professor: Roberto Hugo Wanderley Pinheiro

Período: 2024.1

1. Introdução

O projeto é a implementação do jogo de tabuleiro boop usando as regras tradicionais do jogo (encontradas [aqui](#)). Se preferir ver um vídeo, existem vários no [Youtube](#).

2. Nota

A nota se dará de acordo com os itens elencados neste documento de acordo com o quanto do jogo está funcional e sem *bugs*. Não é permitido o uso de variáveis globais no projeto e será obrigatório algumas funções específicas no projeto, de acordo com a Seção 3. Recomendo acompanhar esse texto com as regras ao lado para não se perder.

2.1. Preparação [1,0 ponto]

Exibição da tela contendo o tabuleiro 6x6 e o estoque de peças que cada jogador possui (gatinhos e gatos). O tabuleiro inicializa vazio e cada jogador começa a partida com 8 gatinhos e 0 gatos. O tabuleiro deve ser referenciado por coordenadas (letras e números para indicar linhas e colunas).

2.2. Jogar Peça [1,0 pontos]

O jogador deve inserir uma peça sua no tabuleiro a cada rodada. Para tal, ele indica a coordenada (ou usa qualquer método que o grupo preferir). As coordenadas escolhidas devem ser validadas, pois o jogador precisa respeitar as regras de posicionamento. Dentre as regras de posicionamentos temos: i) espaço precisa estar vazio; ii) espaço precisa ser dentro do tabuleiro. Lembre-se também de que o jogador precisa escolher qual tipo de peça irá jogar (gatinho ou gato).

2.3. Executar Boop [3,0 pontos]

Após posicionamento da peça, realize os boops necessários. Lembrando que gatinhos não empurram gatos e que quaisquer peças empurradas para fora do tabuleiro retornam ao estoque de peças do jogador.

2.4. Graduar Gatinhos [2,5 pontos]

Quando existe um agrupamento de 3 ou mais peças de um mesmo jogador na diagonal, na horizontal ou na vertical, existe uma graduação dos gatinhos. Isto é, o grupo de 3 peças é

removido do tabuleiro e todos os gatinhos são convertidos em gatos. Ajuste o estoque das peças do jogador corretamente. Caso existe mais de uma possibilidade de graduação, realize a primeira encontrada pelo sistema.

2.5. Fim da Partida [2,5 pontos]

A partida pode encerrar se um jogador tiver um agrupamento de 3 ou mais gatos ou se um jogador tiver todas as suas 8 peças no tabuleiro (estoque zerado).

3. Funções Obrigatórias

Durante a implementação do projeto, deverão ser implementadas algumas funções obrigatórias:

- **void exibeTabuleiro (void **tabuleiro, int linhas, int colunas):** essa função irá imprimir todo o tabuleiro na tela de acordo com a matriz do jogo
- **? naCasa (void **tabuleiro, int linha, int coluna, int linhas, int colunas):** essa função retorna o que está em uma determinada casa (espaço) do tabuleiro. Todas as casas do tabuleiro são endereçadas como no Xadrez, usando números e letras. O número indica a linha e a letra indica a coluna. O tipo de retorno da função provavelmente seria um struct, mas fica a critério de vocês.
- **int validaJogada(void **tabuleiro, int linhaA, int colunaA, int linhas, int colunas):** essa função verifica se a intriga realizado pelo jogador é possível ou não.
- **int fazBoop(void **tabuleiro, int linhaA, int colunaA, int linhas, int colunas):** essa função realiza os boops necessários de fato.
- **int graduar(void **tabuleiro, int linhas, int colunas):** função verifica se há graduações a serem feitas e as realiza.
- **int verificaVitoria(void **tabuleiro, void casa, int linhas, int colunas):** essa função verifica se o jogador conseguiu obter a vitória ou não.

Os voids que botei nos parâmetros é simplesmente para deixar vocês livres para usarem o tipo que quiserem (int, char ou até algum struct). Os voids de retorno de função, é simplesmente pela função não ter um retorno mesmo. Talvez algumas funções estejam com parâmetros faltando, insira mais se desejar ou precisar. A falta de qualquer uma dessas funções acarretará em perda na nota.

4. Pontuação Extra

Você poderá conseguir pontuação extra trazendo as seguintes adições ao seu projeto:

- Fazer uso de arquivos para salvar o estado atual da partida e carrega-la, assim o jogador pode voltar de onde parou (totalizando 1 ponto extra).

5. Datas Importantes

Datas	Atividades
22/10/2024	Demonstração do Projeto
24/10/2024	Especificações do Projeto (esse documento)
29/10/2024	Formação dos Grupos
28/11/2024	Entrega do Projeto
03/12/2024	Apresentação dos Grupos
06/12/2024	Divulgação das Notas do Projeto