

Discente: Gabriela Sarah Moura Silva.

Disciplina: Estrutura de Dados I.

Projeto: Snake Quest.

Breve Descrição: uma versão do jogo Snake, onde a cobrinha come uma maçã e a cada comida ela cresce em comprimento, cada maçã coletada adiciona 10 ao Score. Sendo Game Over caso a cabeça da cobra colida com o próprio corpo, ou colida com as paredes/bordas da janela. Usei um arquivo para salvar o ranking, sendo que só aparece as 5 primeiras melhores pontuações.

Texto Explicativo Sobre Onde Foram Utilizados os Itens do CheckList

- **Vetor ou Matriz:** Essa estrutura foi utilizada para mapear os espaços vazios, e espaços possíveis de inserir a comida. No projeto a matriz booleana `boolean[][] walls` em `GameBoard` mapeia paredes; a mesma estrutura é usada temporariamente por `SearchAlgorithms` no método `floodFill` para marcar células alcançáveis.
- **Recursividade:** Não há uso de recursividade no código enviado; todos os algoritmos de busca usam BFS iterativo com `Queue`.
- **Generic:** Criei algumas estruturas de dados usando `generic` (templates), sendo elas: `GenericArray`; `LinkedList`; `Node`; `Queue`. `GenericArray` é usada pelo `ScoreManager` e por `FoodManager`.
- **Arquivos:** Foi criado um arquivo `scores.txt` para guardar as pontuações dos jogadores, e ser lido na classe `ScoreManager` pelo método `loadScores` para carregar os 5 melhores placares. Sendo exibido em ordem decrescente de Scores.
- **Pilha ou Fila:** Não a pilha; a fila foi implementada usando a `LinkedList`. Ela é usada pela classe `SearchAlgorithms` para BFS.
- **Lista Encadeada:** Classe `LinkedList` usada na `SearchAlgorithms`, é uma lista encadeada genérica usada para busca de caminhos e como base da `Queue`.
- **Algoritmo de Busca:** classe `SearchAlgorithms`, que é usada na `FoodManager` para implementar BFS usando a `Queue`; testando o mapa para ver se tem paredes ou não nele, assim garantindo que a comida apareça apenas em células alcançáveis.
- **Algoritmo de Ordenação:** Classe `InsertionSort`, usada na `ScoreManager`, para ordenar o `loadScores`. Usada para ordenar o ranking com base no Score atual.

Estrutura	Onde usada	Observação
GameBoard	SnakeGame, FoodManager, SearchAlgorithms	Armazena walls e fornece isWall
Snake	SnakeGame, FoodManager, SearchAlgorithms	Movimento, occupies, colisões
Segment	Snake, SnakeGame	Representa coordenadas (x,y)
Direction	Snake, SnakeGame	Controle de direção e prevenção de reversão
GenericArray	ScoreManager, FoodManager	Armazenamento dinâmico genérico
LinkedList	SearchAlgorithms, Queue	Representa caminhos e base da fila
Node	LinkedList	Nó interno da lista encadeada
Queue	SearchAlgorithms, BFS	Fila FIFO usada em buscas
SearchAlgorithms	FoodManager, testes	BFS, flood-fill, shortestPathBFS
InsertionSort	ScoreManager	Ordena scores ao carregar arquivo
ScoreEntry	ScoreManager	Registro de nome, score e timestamp
scores.txt	ScoreManager, SnakeGame	Persistência do ranking