Algoritmos e Programação Estruturada
Trabalho em Grupo N2

v:1.0.1

1 Instruções Iniciais

Desenvolver um jogo interativo utilizando a linguagem de programação C.

2 Códigos-fonte

Crie um repositório git (Github ou Gitlab).
Todos os código produzidos devem estar em arquivos *.c*, *.h* ou *Makefile* *.

* exceção descrita na seção 3.3.2.

Algumas dicas para códigos bem feitos:

- **Nomes Claros e Significativos**: Use nomes descritivos para variáveis, funções e classes. Por exemplo, calcularMedia é mais claro do que calc.
- **Simplicidade e Clareza**: Escreva funções pequenas que realizam uma única tarefa. Isso facilita a leitura e a manutenção do código.
- **Indentação Consistente**: Mantenha uma indentação consistente para melhorar a legibilidade. Isso ajuda a visualizar a estrutura do código facilmente.
- **Evite Repetição**: Siga o princípio DRY (Don't Repeat Yourself). Reutilize código sempre que possível para evitar duplicação.
- **Seja honesto**: Você não aprende nada copiando código de terceiros (isso inclui IAs) nem pedindo a uma pessoa externa ao grupo que faça o trabalho por você. Se a cópia for detectada, a nota de vocês será zerada e as devidas providências serão tomadas.

3 Desafio

Para criação do jogo, será necessário atender aos seguintes requisitos:

3.1 Estrutura do Programa

3.1.1 Struct

Cada componente do jogo deverá ser descrito por uma `struct`. Se não for necessário para o componente, não precisa utilizar. Contudo, o jogo deverá ter componentes em `structs`.

3.1.2 Alocação Dinâmica

Todos os componentes do jogo que podem surgir ou deixar de existir durante a fase deverão ser feitos via alocação dinâmica.

3.1.3 Arquivos

O jogo deverá ter 5 fases diferentes escritas em 5 arquivos diferentes, carregadas quando a fase for iniciada.
Também deverá ter um arquivo binário com as estatísticas do jogo.

3.2 Jogo

3.2.1 Tela principal

A tela principal deverá ter 2 menus:

- Iniciar jogo
- Estatísticas

Lembre-se de salvar os placares dos usuários para adicionar em Estatísticas.

3.2.2 Iniciar Jogo

Quando o jogar iniciar o jogo, as fases serão carregadas em sequência, em ordem, quando o usuário finalizar a atual, carrega a próxima. Quando o usuário "perder", o placar dele deverá ser salvo.

3.3 Pontos Extras

3.3.1 Fases geradas Proceduralmente

Você pode criar uma rotina que crie as fases proceduralmente, contudo, essas deverão compor a 6ª fase em diante.
As primeiras 5 fases deverão ser estáticas, guardadas em arquivo.

3.3.2 Interface Gráfica

Você pode criar a interface do jogo utilizando outras ferramentas (como o OpenGL, por exemplo).
É a única parte do trabalho que permite o uso de outras linguagens e ferramentas.

3.3.3 Tela Principal

Você pode criar outros menus na tela principal e outras funcionalidades além das descritas na seção 3.2.1.
Alguns exemplos:

- Continuar
- Configuração de controle
- Tela de Fim de Jogo

4 Exemplo

Um exemplo simples é o clássico jogo Snake.

4.1 Partes da Cobra

Cada parte da cobra pode ser uma `struct` que conterá um ponteiro para a próxima parte, caso seja a ponta do rabo da cobra, o ponteiro é `NULL`.
A cobra será uma lista encadeada, iniciando da boca da cobra.

4.2 Alimentos

Os alimentos também podem ser `structs`.

Os alimentos serão compostos pelas posições x e y. Você pode acrescentar mais componentes que aumentem as características dos alimentos. Use a criatividade.

4.3 Tabuleiro

Você pode construir 5 tabuleiros, em 5 arquivos separados, o primeiro pode ser a fase sem obstáculos, os demais podem ser