



**Ministério da Educação
Instituto Federal De Goiás
Curso Superior de Tecnologia em Análise e Desenvolvimento de
Sistemas**



RELATÓRIO TÉCNICO

Gustavo Alves dos Santos e Murilo Faria Machado Diniz

20241020230248 / 20241020230043

Orientador(a): Prof. Dr. Aladir Ferreira Silva Júnior



Jataí, Setembro de 2024

SUMÁRIO

1. Introdução.....	3
2. Desenvolvimento.....	4
2.1 Estrutura Geral do Código.....	4
2.2 Estrutura de Dados.....	4
2.3 Embaralhamento de Perguntas.....	5
2.4 Leitura e Validação da Resposta do Usuário.....	5
2.5 Lógica do Jogo e Sistema de Pontuação.....	6
2.6 Finalização e Classificação.....	6
3. Considerações Finais.....	7
3.1 Análise da Implementação Técnica.....	7
3.2 Sistema de Pontuação e Dinâmica do Jogo.....	7
3.3 Usabilidade e Interatividade.....	8
3.4 Possíveis Melhorias Futuras.....	8
3.5 Desempenho e Eficiência.....	9
3.6 Segurança e Robustez.....	9
4. Conclusão.....	9
5. Referências Bibliográficas.....	10



RESUMO

Este relatório apresenta o desenvolvimento de um jogo de perguntas e respostas, focado em temas ambientais e matemáticos, utilizando a linguagem de programação C. O projeto implementa uma estrutura modular com funções dedicadas para o embaralhamento de perguntas, leitura de respostas e um sistema de pontuação progressiva, visando uma experiência de aprendizado dinâmica e interativa. O jogo também inclui um sistema de feedback ao jogador, que permite acompanhar o desempenho em tempo real. São sugeridas melhorias futuras, como a introdução de uma interface gráfica, modos multiplayer, banco de dados para perguntas e níveis de dificuldade. O código é analisado em termos de usabilidade, compatibilidade entre plataformas, eficiência e segurança.

Palavras-chave: Jogo de perguntas e respostas, linguagem C, estrutura de dados, pontuação progressiva, embaralhamento de perguntas, interface gráfica, usabilidade, compatibilidade.

1. Introdução

O projeto consiste no desenvolvimento de um jogo de perguntas e respostas voltado para a conscientização ambiental e questões matemáticas. O objetivo principal deste jogo é desafiar os jogadores a responderem corretamente perguntas variadas, utilizando uma abordagem educativa e divertida. O sistema foi desenvolvido em linguagem C, utilizando conceitos de estruturas de dados, manipulação de strings, e controle de fluxo para gerenciar o comportamento das perguntas e a pontuação dos jogadores. Este relatório tem como finalidade detalhar a implementação do jogo, explicando os aspectos técnicos mais relevantes, como estruturas utilizadas, funções, e a lógica geral do programa.



2. Desenvolvimento

2.1 Estrutura Geral do Código

O código é composto por três partes principais: a definição da estrutura de dados que armazena as perguntas, a implementação das funções de manipulação do jogo e o loop principal responsável pela interação com o usuário. Cada uma dessas partes será descrita detalhadamente a seguir.

2.2 Estrutura de Dados

O jogo utiliza uma **estrutura de dados** para armazenar as informações de cada pergunta. 'A struct Pergunta' contém o enunciado da pergunta, quatro opções de resposta e o índice da resposta correta. A utilização de uma estrutura facilita a manipulação das perguntas, permitindo que sejam agrupadas e organizadas de forma clara e eficiente.

Por exemplo:

```
typedef struct {  
    char pergunta[256];  
    char opcoes[4][256];  
    char resposta_correta;  
} Pergunta;
```

Descrição: A estrutura Pergunta armazena:

- pergunta: Um vetor de caracteres que contém o enunciado da pergunta.
- opcoes: Uma matriz de strings onde cada linha representa uma das quatro opções de resposta.
- resposta_correta: Um caractere que representa a resposta correta (A, B, C ou D), utilizado para validar a entrada do usuário.

Essa abordagem permite uma organização modular, onde cada pergunta é tratada como uma entidade independente, o que facilita a escalabilidade do código.



2.3 Embaralhamento de Perguntas

O embaralhamento é realizado por meio da função `embaralhar_perguntas`, que utiliza o algoritmo de Fisher-Yates para reordenar as perguntas de forma aleatória. Esse processo garante que a ordem das perguntas seja diferente a cada execução, evitando que os jogadores decorem a sequência.

Por exemplo:

```
void embaralhar_perguntas(Pergunta perguntas[], int num_perguntas) {  
    for (int i = 0; i < num_perguntas; i++) {  
        int j = rand() % num_perguntas;  
        Pergunta temp = perguntas[i];  
        perguntas[i] = perguntas[j];  
        perguntas[j] = temp;  
    }  
}
```

Descrição: A função embaralha o vetor de perguntas utilizando a função `'rand()'`, que gera números pseudoaleatórios. A cada iteração, uma pergunta é trocada de posição com outra, garantindo uma nova disposição a cada execução do jogo. Isso proporciona mais dinamismo ao jogo e evita repetições previsíveis.

2.4 Leitura e Validação da Resposta do Usuário

Para capturar a entrada do usuário, o código implementa a função `ler_resposta`, que solicita ao jogador que digite a letra correspondente à sua escolha. A função utiliza `'toupper()'` para padronizar a entrada, aceitando letras minúsculas ou maiúsculas.

Por exemplo:

```
char ler_resposta() {  
    char resposta;  
    printf("Digite a letra da sua resposta (A, B, C, D): ");  
    scanf(" %c", &resposta);  
    resposta = toupper(resposta);  
    return resposta;  
}
```

Descrição: Esta função garante que a entrada do jogador esteja no formato correto, padronizando a resposta em maiúsculas para comparação com a resposta correta armazenada. Isso evita problemas com a sensibilidade a maiúsculas e minúsculas, que poderiam levar a erros de validação.



2.5 Lógica do Jogo e Sistema de Pontuação

O jogo possui uma estrutura de controle em loop que percorre todas as perguntas embaralhadas, exibindo-as uma a uma ao jogador. A pontuação do jogador é gerenciada de forma incremental e progressiva, com um sistema de bônus que dobra os pontos a cada resposta correta consecutiva.

Descrição:

- **Estrutura de Controle:** O jogo percorre o vetor de perguntas utilizando um loop for. A cada iteração, uma pergunta é apresentada ao usuário, juntamente com quatro opções de resposta.
- **Pontuação:** A cada acerto, a pontuação do jogador é dobrada. Caso o jogador erre, o valor da pontuação volta ao valor inicial (2). Essa abordagem de pontuação progressiva incentiva o jogador a acertar o maior número de perguntas consecutivamente.
- **Feedback ao Jogador:** A cada pergunta respondida, o jogo informa se a resposta está correta ou errada e exibe a pontuação atual do jogador.

2.6 Finalização e Classificação

Ao final do jogo, o sistema exibe a pontuação total e classifica o jogador com base em seu desempenho:

- **Profissional:** Se acertar todas as 30 perguntas.
- **Experiente:** Se acertar mais de 20.
- **Novato:** Se acertar mais de 10.
- **Necessidade de Estudo:** Se acertar menos de 10.

Este sistema de classificação motiva os jogadores a buscar um melhor desempenho e aprender mais sobre o meio ambiente.



3. Considerações Finais

O projeto desenvolvido oferece uma plataforma interativa para testar os conhecimentos dos usuários por meio de um jogo de perguntas e respostas sobre temas ambientais e matemáticos. Através da combinação de perguntas diversificadas e um sistema de pontuação dinâmico, o código proporciona uma experiência desafiadora e educativa. A seguir, são detalhados os pontos mais relevantes do sistema, suas implicações, e possíveis aprimoramentos.

3.1 Análise da Implementação Técnica

A implementação se destacou por utilizar estruturas de dados adequadas e uma lógica bem organizada. A escolha da linguagem C para este tipo de aplicação foi apropriada, dada a sua capacidade de manipulação de strings e estruturas de dados de maneira eficiente. No entanto, alguns aspectos do código podem ser analisados em maior detalhe:

- **Organização e Modularidade:** O código foi escrito de forma modular, dividindo-se em funções independentes para cada uma das funcionalidades do jogo, como 'embaralhar_perguntas' e 'ler_resposta'. Essa abordagem facilita a manutenção e futuras expansões. Por exemplo, se novas funcionalidades forem adicionadas, como a criação de um banco de dados de perguntas, o código atual pode ser facilmente adaptado para incluir essas melhorias.
- **Uso de Estruturas de Dados:** A utilização da 'struct Pergunta' para armazenar as informações de cada pergunta e suas respostas foi uma decisão acertada. Esse tipo de estrutura permite um acesso mais organizado às perguntas e respostas e facilita a manipulação de múltiplas questões ao mesmo tempo. Além disso, a utilização de arrays para as opções de resposta assegura uma manipulação eficiente das escolhas do jogador.

3.2 Sistema de Pontuação e Dinâmica do Jogo

O sistema de pontuação é um dos elementos que mais influenciam a experiência do jogador. A implementação de um esquema de pontuação progressiva, onde a cada acerto consecutivo o valor da pontuação é dobrado, adiciona uma camada de desafio ao jogo. Essa característica encoraja o jogador a manter uma série de acertos e recompensa o desempenho contínuo.



Além disso, o **feedback instantâneo** após cada pergunta contribui para a imersão do jogador, já que ele pode acompanhar o seu progresso em tempo real e ajustar suas estratégias para as perguntas seguintes.

3.3 Usabilidade e Interatividade

Do ponto de vista de **usabilidade**, o código atende bem ao propósito, sendo fácil de jogar e interagir. No entanto, algumas áreas poderiam ser melhoradas para aprimorar a experiência do usuário:

- **Interface do Usuário:** O jogo é, atualmente, baseado em texto, e, embora seja funcional, poderia ser melhorado com uma interface gráfica (GUI). A adição de uma interface gráfica tornaria o jogo mais atraente e fácil de usar, especialmente para jogadores com menos familiaridade com a interface de linha de comando.
- **Compatibilidade entre Plataformas:** A função 'limpar_tela' utiliza o comando 'system("cls")', que é específico do sistema operacional Windows. Para melhorar a compatibilidade com outros sistemas operacionais, como Linux ou macOS, seria interessante implementar uma verificação do sistema operacional e ajustar o comando de limpeza de tela de acordo.

3.4 Possíveis Melhorias Futuras

Este jogo, apesar de funcional, pode ser expandido em diversas áreas para oferecer uma experiência mais rica e educativa. Algumas sugestões de melhorias são:

- **Banco de Dados para Armazenamento de Perguntas:** Atualmente, o conjunto de perguntas é codificado diretamente no programa. Um aprimoramento útil seria a implementação de um banco de dados externo (como um arquivo CSV ou um banco de dados SQL) para armazenar as perguntas. Isso permitiria a fácil adição, remoção ou atualização das perguntas sem a necessidade de alterar o código-fonte diretamente.
- **Modo Multiplayer:** Uma expansão interessante seria a introdução de um modo multiplayer, onde dois ou mais jogadores competiriam entre si em um formato de perguntas e respostas. Isso poderia ser implementado utilizando a mesma lógica de pontuação, com alternância de turnos entre os jogadores.
- **Aprimoramento do Sistema de Feedback:** Para melhorar a experiência de aprendizado do jogador, seria interessante fornecer explicações adicionais sobre as respostas erradas. Quando o jogador errar uma pergunta, além de informar qual era a resposta correta, o sistema poderia exibir uma explicação sucinta



sobre o porquê daquela ser a resposta correta, agregando um valor educativo mais significativo ao jogo.

- **Níveis de Dificuldade:** Outra melhoria seria a introdução de níveis de dificuldade. As perguntas poderiam ser categorizadas em diferentes níveis (fácil, médio, difícil), e o jogador escolheria o nível ao iniciar o jogo. Isso tornaria o jogo mais adaptável às diferentes capacidades dos jogadores.
- **Adaptação para Dispositivos Móveis:** Com a crescente popularidade de dispositivos móveis, a adaptação do jogo para plataformas móveis (por meio de uma versão em C compatível ou até a migração para outras linguagens, como JavaScript para aplicações web) aumentaria significativamente o alcance e a acessibilidade do jogo.

3.5 Desempenho e Eficiência

Do ponto de vista de desempenho, o código é eficiente para um jogo de perguntas e respostas simples. O embaralhamento das perguntas com o algoritmo de Fisher-Yates é rápido e adequado para o número de perguntas definido. Contudo, em sistemas maiores, a utilização de algoritmos de ordenação e busca mais otimizados pode ser necessária, especialmente se o número de perguntas aumentar consideravelmente ou se houver necessidade de buscas complexas no banco de dados.

3.6 Segurança e Robustez

Embora o código seja funcional, algumas medidas de segurança poderiam ser implementadas para melhorar a robustez do sistema. Por exemplo, a entrada do usuário poderia ser mais rigorosamente validada para evitar erros, como a inserção de caracteres inválidos na função 'ler_resposta'. Implementar um tratamento de exceções e erros mais robusto aumentaria a confiabilidade do sistema.



4. Conclusão

O projeto foi bem-sucedido em atingir seu objetivo de criar um jogo interativo de perguntas e respostas, proporcionando uma experiência educativa aos jogadores. A estrutura do código é clara, modular e eficiente, o que permite futuras expansões e melhorias. No entanto, há várias oportunidades de aprimoramento, tanto na interface quanto nas funcionalidades, que poderiam transformar o jogo em uma ferramenta ainda mais interativa e educacional.

5. Referências Bibliográficas

SCHILDT, Herbert. C completo e total. 3. ed. São Paulo: Makron Books, 2004.