

TANNA – MiniGames Creator

Joeth Balczar | No-2222116 | TGPSIP22 | Carla Macedo

The logo for Digital Escola Profissional consists of a solid blue rectangle. Inside the rectangle, the word 'digital' is written in a large, white, lowercase, sans-serif font. Below 'digital', the words 'escola' and 'profissional' are written in a smaller, white, lowercase, sans-serif font, stacked vertically.

TANNA é um criador de minijogos simples desenvolvido em C# com Windows Forms, projetado para permitir uma criação fácil e rápida de jogos personalizados. Os jogadores podem criar seus próprios jogos, configurar mundos e desafiar um chefe final após derrotar todos os inimigos.

Índice

1. Introdução	3
2. Tecnologias e Recursos	3
Tecnologias Utilizadas	3
Recursos e Ferramentas	4
3. Implementação	4
3.1 Cronograma de Desenvolvimento do Projeto	4
3.2 Aspetos Técnicos do Desenvolvimento do Projeto	5
4. Conclusão	5
5. Bibliografia/Anexos	5

1. Introdução

O projeto TANNA é um criador de minijogos desenvolvido em C# com Windows Forms, projetado para simplificar a criação e personalização de jogos. Destinado a iniciantes no desenvolvimento de jogos, oferece uma interface intuitiva onde usuários podem criar jogos personalizados ajustando variáveis como número de inimigos, características dos mundos e desafios. Este relatório detalha o processo de desenvolvimento, desafios enfrentados, soluções adotadas e sugestões para futuros aprimoramentos do TANNA, seguindo as diretrizes estabelecidas para relatórios técnicos.

2. Tecnologias e Recursos

Tecnologias Utilizadas

No desenvolvimento do projeto TANNA foram utilizadas as seguintes tecnologias principais:

- **C#:** Linguagem de programação escolhida devido à sua robustez e suporte nativo para o desenvolvimento de aplicações Windows Forms.
- **Windows Forms:** Framework gráfico da Microsoft utilizado para criar a interface de utilizador (UI), proporcionando uma experiência integrada e familiar aos utilizadores dos sistemas operativos Windows.
- **SQLite:** Sistema de gestão de base de dados relacional escolhido para armazenamento de dados do TANNA. Optou-se pelo SQLite devido à sua leveza, facilidade de integração e capacidade de operar com um único ficheiro de base de dados (.db), o que simplifica o transporte e distribuição da aplicação.

Recursos e Ferramentas

Além das tecnologias principais, foram utilizados os seguintes recursos e ferramentas durante o desenvolvimento do projeto:

- **Visual Studio:** Ambiente de desenvolvimento integrado (IDE) utilizado para escrever, depurar e compilar o código-fonte do TANNA. O Visual Studio oferece ferramentas avançadas que facilitam a criação e manutenção de aplicações Windows.
- **SQLite Studio:** Ferramenta utilizada para gerir a base de dados SQLite do projeto. O SQLite Studio permitiu visualizar e editar graficamente a estrutura da base de dados, realizar consultas SQL complexas e monitorizar o desempenho, contribuindo para uma administração eficiente dos dados do TANNA.
- **Hardware Padrão de Desenvolvimento para Aplicações Windows:** Utilização de hardware compatível com as especificações mínimas para aplicações Windows, garantindo que o TANNA seja executado de forma eficiente e estável em ambientes finais.
- **ChatGPT (Fonte de Informação de Código):** Recurso utilizado para consultas de informação específicas relacionadas com programação e desenvolvimento, contribuindo para resolver desafios técnicos e melhorar o conhecimento necessário para o projeto.

3. Implementação

3.1 Cronograma de Desenvolvimento do Projeto

O projeto foi desenvolvido ao longo de aproximadamente um mês, dividido nas seguintes fases:

- **Pesquisa e Planeamento:** Investiguei as tecnologias viáveis para o projeto e defini suas bases.
- **Modelagem da Base de Dados:** Criei as tabelas necessárias, como Player, Games, World, Games_Enemies, Enemies e FinalBoss, utilizando o DBeaver para uma modelagem gráfica eficaz.
- **Conexão com a Base de Dados:** Optei pela utilização do SQLite devido à sua simplicidade e eficiência, permitindo armazenar todos os dados em um arquivo .db acessível diretamente pelo projeto através do arquivo program.cs, que inicializa a aplicação de forma centralizada.
- **Implementação do CRUD:** Desenvolvi funcionalidades para criar, ler, atualizar e deletar dados, possibilitando que administradores gerenciem usuários e que usuários normais criem e editem seus próprios jogos, mundos, inimigos e chefes finais.

- **Integração dos Jogos Criados:** No final do desenvolvimento, integrei os jogos criados em uma lista acessível aos usuários, onde cada jogo personalizado pode ser selecionado e carregado com base nos dados armazenados, como nome e tamanho do mundo, tipos e quantidades de inimigos, e atributos do chefe final.

3.2 Aspectos Técnicos do Desenvolvimento do Projeto

Requisitos do Projeto: O projeto visava permitir a criação de jogos personalizados através da gestão de jogadores, mundos e confrontos com chefes finais.

Modelo de Dados: Utilizou-se SQLite para armazenar de forma eficiente informações sobre jogadores, jogos e mundos, com um diagrama de entidades-relações detalhando a estrutura da base de dados.

Interfaces e Fluxos de Dados: Foi implementado um conjunto de classes que suportam a lógica de negócio do projeto, facilitando a interação do usuário com a aplicação e garantindo fluxos de dados claros e eficientes.

3. Conclusão

O projeto TANNA alcançou êxito ao proporcionar um ambiente intuitivo para a criação de jogos simples. Durante o desenvolvimento, enfrentamos desafios significativos, como a integração eficiente de uma base de dados em C# sem recorrer a código repetitivo. Organizar as tabelas e estabelecer suas relações foi facilitado pelo uso consistente de um único arquivo para centralizar as operações em todas as tabelas. A abordagem foi complementada por um processo de desenvolvimento que exigiu paciência e reflexão profunda, resultando na resolução eficaz das complexidades das relações entre tabelas.

Para o futuro, planejamos expandir as capacidades dos inimigos e do chefe final, tornando-os mais dinâmicos e desafiadores. Além disso, buscamos incrementar o jogo com novas funcionalidades que evitem a monotonia durante o gameplay, proporcionando uma experiência mais envolvente e atrativa para os jogadores.

5. Bibliografia/Anexos

- chatgpt.com
- [Fazer jogo em c#](#)
- [Fazer jogo em c# 2](#)
- [Fazer jogo em c# 3](#)