

Descrição do Projeto: Quiz e Simulação de Corrida da Fórmula E

Este projeto consiste em dois códigos Python que trabalham em conjunto para oferecer uma experiência interativa relacionada à Fórmula E. O primeiro código atua como um menu principal, permitindo ao usuário escolher entre iniciar um quiz, simular uma corrida, acessar o site oficial da Fórmula E ou sair do programa. O segundo código é responsável pela simulação da corrida, onde o usuário pode interagir escolhendo ações durante a competição.

Componentes do Projeto

1. Primeiro Código: Menu Principal e Quiz

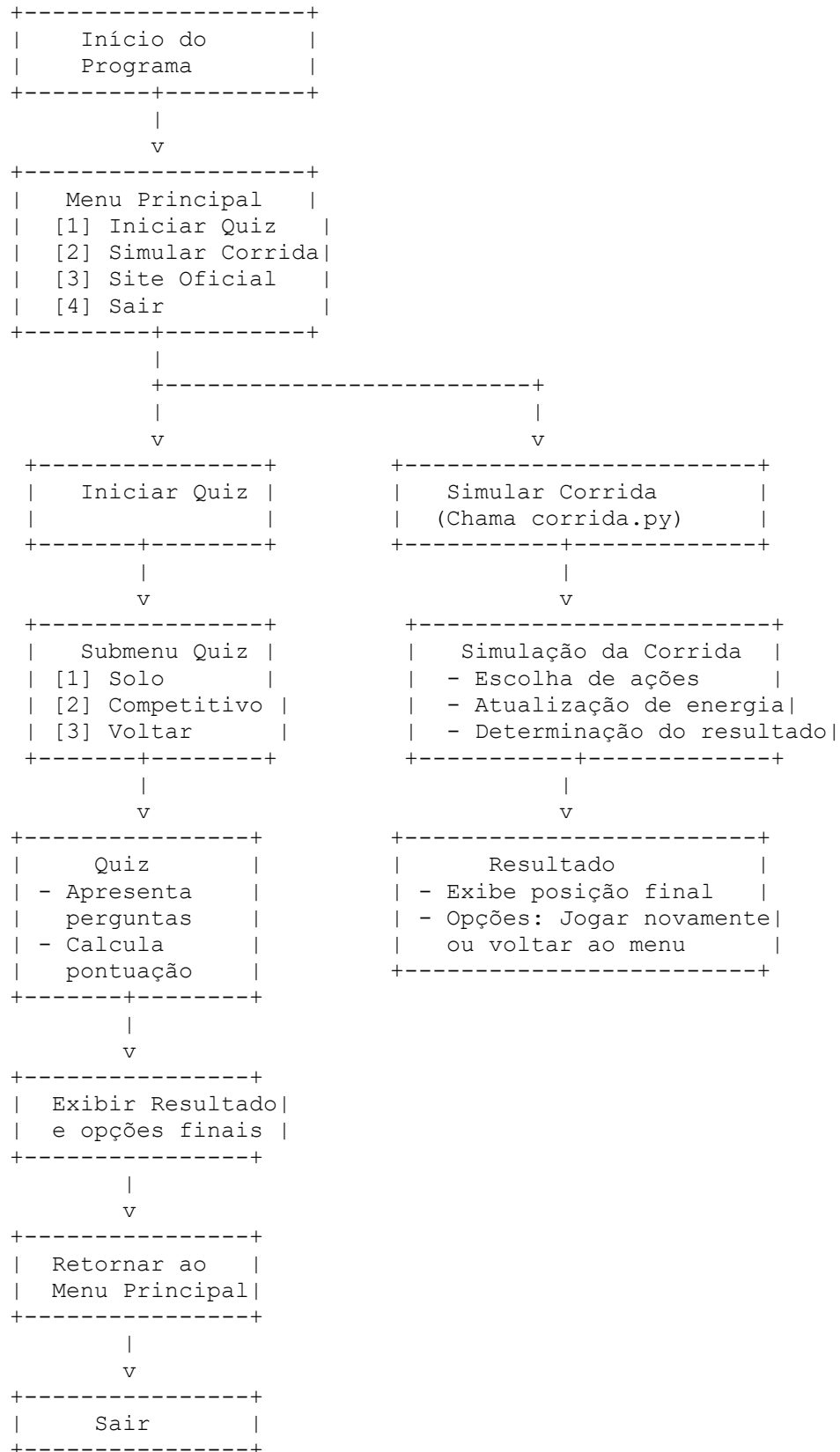
- **Funcionalidades:**
 - **Menu Principal:** Apresenta opções para iniciar o quiz, simular uma corrida, acessar o site oficial ou sair.
 - **Quiz:** Oferece perguntas relacionadas à Fórmula E, permitindo modos de jogo solo e competitivo (dois jogadores).
 - **Navegação:** Permite retornar ao menu principal ou encerrar o programa após a conclusão do quiz.
 - **Integração:** Chama o segundo código (*corrida.py*) para simular a corrida quando a opção correspondente é selecionada.
- **Estrutura:**
 - **Funções Principais:**
 - *menu()*: Exibe o menu principal e captura a escolha do usuário.
 - *main()*: Controla o fluxo principal do programa com base na escolha do menu.
 - *quiz()*: Gerencia o modo de quiz solo.
 - *competitivo()*: Gerencia o modo de quiz competitivo para dois jogadores.
 - *menu_quiz()*: Submenu para escolher entre modos solo, competitivo ou voltar ao menu principal.
 - *fazer_pergunta()*: Apresenta cada pergunta do quiz e valida a resposta do usuário.

2. Segundo Código: Simulação de Corrida (*corrida.py*)

- **Funcionalidades:**
 - **Configuração da Corrida:** Solicita o nome do piloto e a escolha da equipe.
 - **Simulação Dinâmica:** Gerencia a corrida com múltiplos rounds, onde o jogador escolhe ações que afetam sua energia e posição.
 - **Eventos Aleatórios:** Introduce elementos de sorte, como sucesso ou falha em ultrapassagens e reabastecimento de energia.
 - **Resultados:** Determina a posição final do jogador e oferece opções para jogar novamente, voltar ao menu principal ou sair.
- **Estrutura:**
 - **Funções Principais:**
 - *main()*: Inicializa a corrida solicitando informações do jogador e iniciando a simulação.
 - *simular_corrida()*: Controla a lógica da corrida, incluindo escolhas do jogador e atualização de energia e posição.
 - *obter_opcao()*: Valida a entrada do usuário para selecionar opções durante a corrida.

Diagrama em Blocos: Funcionamento do Projeto

A seguir, apresentamos um diagrama em blocos simplificado que descreve o fluxo de funcionamento do projeto:



Descrição do Diagrama em Blocos

1. **Início do Programa:**
 - O programa começa executando o primeiro código, que apresenta o menu principal ao usuário.
2. **Menu Principal:**
 - **Opção 1 Iniciar Quiz:**
 - Leva ao submenu do quiz, onde o usuário pode escolher entre jogar sozinho (solo) ou de forma competitiva (dois jogadores).
 - Após a conclusão do quiz, o usuário pode optar por retornar ao menu principal ou sair.
 - **Opção 2 Simular Corrida:**
 - Chama o segundo código (*corrida.py*) que gerencia a simulação da corrida.
 - O usuário insere seu nome, escolhe uma equipe e interage durante a corrida, fazendo escolhas que afetam sua energia e posição.
 - Ao final da corrida, o usuário pode jogar novamente, voltar ao menu principal ou sair.
 - **Opção 3 Site Oficial da Fórmula E:**
 - Abre o site oficial da Fórmula E no navegador padrão do usuário.
 - **Opção 4 Sair:**
 - Encerra o programa.
3. **Simulação da Corrida (*corrida.py*):**
 - **Configuração Inicial:**
 - Solicita o nome do piloto e a seleção da equipe.
 - **Durante a Corrida:**
 - A corrida é dividida em rounds, onde o jogador escolhe ações como tentar ultrapassagem, manter posição, forçar uma ultrapassagem ou abandonar a corrida.
 - As escolhas afetam a energia e a posição do jogador, com elementos de sorte determinando o sucesso das ações.
 - Se a energia do jogador se esgota, ocorre um reabastecimento com perda de posições.
 - **Final da Corrida:**
 - Determina a posição final do jogador e exibe uma mensagem de vitória ou derrota.
 - Oferece opções para jogar novamente, voltar ao menu principal ou sair.

Fluxo de Integração Entre os Códigos

- **Menu Principal → Simular Corrida:**
 - Quando o usuário seleciona "Simular Corrida" no menu principal, o primeiro código utiliza o módulo subprocess para executar o segundo código (*corrida.py*).
- **Simular Corrida → Menu Principal:**
 - Após a simulação da corrida, o usuário pode optar por retornar ao menu principal, encerrando a execução do segundo código e retomando o fluxo no primeiro código.