🛠️ Manual do Projeto Prático de Controle e Sistemas Embarcados - Trainee EDRA 2025.1

Este manual expões e detalha materiais de auxílio de projeto para a área de Controle e Sistemas Embarcados do processo trainee da EDRA. O manual está separado em assuntos de maneira similar com a que as "subáreas" de C&SE divididas no projeto.

📚 Materiais de Estudo:

Visão Computacional para Detecção de Formas Geométricas Coloridas

- **OpenCV Documentação e Tutoriais:**
  - Documentação Oficial OpenCV
    - **Uso:** Referência completa da biblioteca OpenCV, útil para pesquisa detalhada de funções e módulos.
  - Tutoriais OpenCV-Python na documentação oficial: Basic Image Operations, Image Filtering, Morphological Transformations, Color Spaces, Image Thresholding, Contours e GUI Features.
    - **Uso:** Guia prático com exemplos em Python, ideal para aprender os fundamentos do processamento de imagens com OpenCV. Concentre-se nos seguintes módulos:
      - **Basic Image Operations:** Operações básicas como leitura, escrita e manipulação de pixels em imagens. Essencial para começar a trabalhar com imagens no OpenCV.
      - **Image Filtering:** Técnicas para suavizar imagens, remover ruído e realçar bordas. Inclui filtros como Gaussian Blur e Median Blur, importantes para pré-processamento.
      - **Morphological Transformations:** Operações morfológicas como erosão, dilatação, abertura e fechamento. Úteis para refinar formas geométricas detectadas e remover pequenos ruídos ou imperfeições.
      - **Color Spaces:** Conversão entre diferentes espaços de cor (RGB, HSV, Gray). O espaço HSV é particularmente útil para segmentação de cores, pois separa a informação de cor (Hue) da intensidade (Value) e saturação (Saturation).

        - **Image Thresholding:** Técnicas para segmentar imagens baseadas em limiares, criando imagens binárias. Essencial para isolar regiões de interesse baseadas em cor ou intensidade.
        - **Contours:** Detecção e análise de contornos em imagens binárias. Funções como `findContours`, `approxPolyDP`, possibilitam a extração de contornos e caracterizações desses.
        - **GUI Features:** Ferramentas de Interface Gráfica do Usuário, como sliders e trackbars. Permitem criar interfaces interativas para ajustar parâmetros em tempo real e criar versões do programa feitas para testarem opções e encontrar configurações ideais.
    - Tutoriais de Processamento de Imagem OpenCV-Python
        - **Uso:** Tabela de conteúdos detalhada dos tutoriais de processamento de imagem em Python, onde se concentram os tópicos citados, facilitando a navegação por tópicos específicos.
    - Exemplo de interface com parâmetros para teste
        - **Uso:** Referência visual de uma interface de usuário interativa para ajuste de parâmetros em um detector de formas geométricas. Demonstra uma sequência de operações de visão computacional e como os parâmetros podem ser controlados via interface.
    - Exemplo de detecção de formas geométricas com OpenCV
        - **Uso:** Tutorial prático de detecção de formas geométricas básicas (triângulos, quadrados, círculos) em imagens usando OpenCV e Python. Útil para entender a detecção de contornos e formas.

Criação de Missões Paramétricas em Simulação ROS2/PX4

- **Docker e PX4:**
    - Documentação Docker da PX4 (Recomendado): Instruções para instalar e configurar o ambiente de simulação PX4 com Docker.
- **ROS2 e Integração PX4:**
    - Documentação geral de ROS2 na PX4: Guia de integração ROS2 com PX4, incluindo navegação, offboard control e simulação, navegue dentro dos tópicos na barra lateral para encontrar informações relevantes
    - Exemplo de navegação básica com ROS2 e PX4 em C++: Interessante ler as partes de setup e observar os links para outras partes da documentação da PX4 na página mesmo se for implementar em Python
    - Exemplo de navegação básica com ROS2 e PX4 em Python: Repositório que tem um exemplo que pode ser usado como base da implementação que vá fazer, inclui métodos e atributos que podem ser reutilizados da classe OffboardControl que extende Node

- **ROS2 Tutoriais Gerais:**
  - ROS2 Tutorials: Focar em "Understanding ROS2 nodes", "Topics", "Services", "Actions", "Writing a simple publisher and subscriber", "Using parameters"
  - ROS2 Tutorial em vídeo da implementação de parâmetros para nós: Vídeo sobre um robô terrestre mas que explica bem como usar parâmetros em nós ROS2
  - Tuturial em vídeo de ROS2 com PX4 em Python: Extende e explica o exemplo de navegação básica em Python, inclui dicas importantes e detalhes de instalação e setup do ambiente e de uso do Gazebo

Planejamento de Missão com Máquina de Estados ou Árvore de Comportamento

- **Máquinas de Estados e Árvores de Comportamento:**
  - **Bibliotecas com Implementações:**
    - python-statemachine: Biblioteca Python para Máquinas de Estados Finitas.
      - Exemplos na documentação: Clique nos exemplos para ver o código e detalhes.
      - Github da biblioteca: Tem um exemplo e explicação de como gerar o gráfico da máquina de estados. Mas é preferível usar o PIP para instalar a clonar o repositório
    - py_trees: Biblioteca Python para Árvores de Comportamento, tem um módulo que pode ser usado para a integração direta com ROS2.
      - Tutoriais na documentação: Passo à passo da implementação de alguns exemplos.
    - BehaviourTree.cpp: Biblioteca C++ para Árvores de Comportamento, apesar de não ser em Python é uma biblioteca com boas explicações na documentação e com 2 versões de uma excelente ferramenta para desenvolver as árvores de comportamento com uma interface gráfica, fazendo com que seja possível cumprir os requisitos do trabalho com muito pouco ou quase nenhum conhecimento de C++. É uma documentação interessante para a leitura mesmo se for usar outra implementação, traz boas explicações dos conceitos básicos das Árvores de Comportamento. Também tem funcionalidades para a integração direta com ROS2.
      - Tutoriais na documentação: Clique nos exemplos para ver o código e detalhes.
      - Github do Groot Open Source: Interface gráfica para criação de árvores de comportamento, mais antiga mas de código aberto e sem recursos pagos.
      - Groot2: Uma interfaçe gráfica mais moderna e com mais recursos, tem todos os recursos necessários para completar a atividade na sua versão gratuita mas também possiu uma versão paga.
  - **Artigos e Tutoriais:**
    - Artigo sobre Árvores de Comportamento em Robótica: O artigo traz, para um escopo intrudutório, um boa explicação do funcionamento e da implementação de uma árvore

de comportamento, além de comparar as duas principais implementações `py_trees` e `BehaviorTree.cpp` e uma breve comparação com o uso de Máquinas de Estados Finitos

🗂️ Recursos e Referências Adicionais

- ROS2 Documentation: Documentação oficial ROS2.
- PX4 Documentation: Raiz da documentação PX4.
- OpenCV Documentation: Raiz da documentação oficial OpenCV.
- QGroundControl: Interface gráfica para controle de drones PX4.
- Gazebo: Simulador de robótica 3D.