RT_SLAM_AVOC

Comprehensive Project Report

Author: Abhishek Chauhan

Project Files

.git/HEAD

ref: refs/heads/main

.git/config

[core]
 repositoryformatversion = 0
 filemode = true
 bare = false
 logallrefupdates = true
[remote "origin"]
 url = https://github.com/deploymentqorix/RT_SLAM_qorix_incubDeployment.git
 fetch =...

.git/description

Unnamed repository; edit this file 'description' to name the repository.

.git/index

DIRC<h¢ñ"<qzh¢ñ"<qzN—‘•¤èè×øa‹2È42êû»ÜüØg>”pÁÇ.github/workflows/npm-gulp.ymlh¢ñ"Gnh¢ñ"GnN—’•¤èè0ÅÀjY€iH•ÜjÑ±}ë&tEâ'LICENSEh¢ñ"Gnh¢ñ"GnN—“•¤èèë°"]5)ÅôW‡õÅ¤¤ëý3aõMakefileh¢ñ"Gnh¢ñ"GnN—••¤èè.G!f•[“rHÍqòž...

.git/info/exclude

git ls-files --others --exclude-from=.git/info/exclude
Lines that start with '#' are comments.
For a project mostly in C, the following would be a good set of
exclude patterns (uncomment them if you...

.git/logs/HEAD

0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from...

.git/logs/refs/heads/main

0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from...

.git/logs/refs/remotes/origin/HEAD

0000000000000000000000000000000000000000 533a0d96e0beff3ef5f5aa66a021ef808b9d93ab L-100791 <l-100791@Vaibhavi.(none)> 1755452401 +0530 clone: from...

.git/packed-refs

pack-refs with: peeled fully-peeled sorted

.git/refs/heads/main

533a0d96e0beff3ef5f5aa66a021ef808b9d93ab

.git/refs/remotes/origin/HEAD

ref: refs/remotes/origin/main

LICENSE

MIT License

Copyright (c) 2025 deploymentqorix

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to...

Makefile

The main Makefile for the entire project

backend/api_server/main.py

```
from fastapi import FastAPI
from fastapi.middleware.cors import CORSMiddleware

app = FastAPI()

app.add_middleware(
    CORSMiddleware,
    allow_origins=["*"],
    allow_credentials=True,
   ...
```

backend/data_streamer/main.py

```
import rclpy
from rclpy.node import Node
from nav_msgs.msg import Odometry
from nav_msgs.msg import OccupancyGrid
import asyncio
import websockets
import json

CONNECTED_CLIENTS = set()

async def...
```

backend/data_streamer/vehicle_pose.py

```
import rclpy
from rclpy.node import Node
from geometry_msgs.msg import PoseStamped

class PoseSubscriber(Node):
    def __init__(self):
        super().__init__('pose_subscriber_for_streamer')
      ...
```

backend/dev_scripts/mock_backend.py

File: backend/dev_scripts/mock_api.py
Description: A standalone FastAPI server to provide mock data for frontend development.
To run: python3 mock_api.py

backend/install/_local_setup_util_ps1.py

Copyright 2016-2019 Dirk Thomas
Licensed under the Apache License, Version 2.0

backend/install/_local_setup_util_sh.py

Copyright 2016-2019 Dirk Thomas
Licensed under the Apache License, Version 2.0

backend/requirements.txt

Python packages for the backend services

backend/ros2_nodes/ros2_nodes/slam_node/CMakeLists.txt

```
cmake_minimum_required(VERSION 3.8)
project(slam_node)

# Find all dependencies
find_package(ament_cmake REQUIRED)
find_package(rclcpp REQUIRED)
find_package(geometry_msgs...
```

backend/ros2_nodes/ros2_nodes/slam_node/src/slam_logic.cpp

```
include <Eigen/Dense>
include <vector>
include "rclcpp/rclcpp.hpp"
include "geometry_msgs/msg/pose_stamped.hpp"
include "sensor_msgs/msg/imu.hpp"
include "sensor_msgs/msg/laser_scan.hpp"
include...
```

docs/rest_api.md

REST API Endpoints

Get System Health

docs/websocket_api.md

Message for Live Pose Updates

frontend/Dockerfile

Stage 1: Build the React application

frontend/README.md

Getting Started with Create React App

frontend/public/robots.txt

https://www.robotstxt.org/robotstxt.html

frontend/src/App.js

```
import React, { useState, useEffect } from 'react';
import { connectWebSocket, disconnectWebSocket } from './services/websocketService';
import MapView from './components/MapView';
import...
```

frontend/src/App.test.js

```
import { render, screen } from '@testing-library/react';
import App from './App';

test('renders learn react link', () => {
  render(<App />);
  const linkElement = screen.getByText(/learn react/i);
...
```

frontend/src/components/MapView.js

```
import React, { useRef, useEffect } from 'react';

const MapView = ({ pose, trail, mapData }) => {
  const canvasRef = useRef(null);

  useEffect(() => {
    const canvas = canvasRef.current;
   ...
```

frontend/src/components/SystemHealthPanel.js

```
import React, { useState, useEffect } from 'react';
import { fetchSystemHealth } from '../services/apiService';

const SystemHealthPanel = () => {
  const [health, setHealth] = useState({
```

...

frontend/src/index.js

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';

const root =...
```

frontend/src/reportWebVitals.js

```
const reportWebVitals = onPerfEntry => {
  if (onPerfEntry && onPerfEntry instanceof Function) {
    import('web-vitals').then(({ getCLS, getFID, getFCP, getLCP, getTTFB }) => {
     ...
```

frontend/src/services/apiService.js

```
const API_URL = "http://localhost:8000";
const MOCK_MODE = true; // Uses mock data from websocketService

export const fetchSystemHealth = async () => {
  if (MOCK_MODE) {
    // --- MOCK MODE:...
```

frontend/src/services/websocketService.js

```
const MOCK_MODE = false;
const WEBSOCKET_URL = "ws://localhost:3000/ws";

let socket = null;
let mockInterval = null;

export const connectWebSocket = (onMessageCallback) => {
  if (MOCK_MODE) {
   ...
```

frontend/src/setupTests.js

```
jest-dom adds custom jest matchers for asserting on DOM nodes.
allows you to do things like:
expect(element).toHaveTextContent(/react/i)
learn more: https://github.com/testing-library/jest-dom
```

rest_api.md

```
REST API Endpoints
# Get System Health
```

websocket_api.md

Message for Live Pose Updates