Robo Desk Buddy - Week 2 Sprint 1 Documentation

Sprint 1: Base Setup and Movement System

Sprint Duration: Week 2
Goal: Establish the foundational robot model, core motion features, and controller threading architecture.

Objectives

1. Create the Desk Buddy robot in Webots with basic geometry (body, head, eyes, and wheels).
2. Implement movement and LED features controlled via Python.
3. Introduce multi-threaded action handling (wave, blink, move, patrol).
4. Design keyboard-controlled interaction system.

Implementation Summary

The Desk Buddy controller was built using Webots primitives and Python threading:

Custom Robot Model — Used `Shape`, `HingeJoint`, `LED`, and `Keyboard` nodes to construct a desktop robot with:

- Body, head, and eye components
- Hinged joints for movement
- LED lights for visual feedback

Python Controller — Developed `DeskBuddy.py` with threaded functions:

- `wave()` → head tilt animation
- `blink_lights()` → LED blinking feedback
- `move_forward()`, `turn_left()`, etc. → differential wheel motion

Thread-Safe Design — Implemented `threading.Lock()` to manage concurrent actions safely, preventing motor conflicts.

Keyboard Interface — Created live simulation control with key bindings (W, B, P, D, Y, etc.).

Key Learnings

- Only the **main thread** should call `robot.step()` in Webots.
- All physical actions (motors, LEDs) must use `time.sleep()` when threaded.
- Proper locking ensures no conflicting motor control.

Sprint 1 Outcome

Feature	Status
Custom Robot Model	✅ Completed
Basic Movements	✅ Functional
Threading System	✅ Stable
LED Animation	✅ Implemented
Keyboard Controls	✅ Working

Result: A fully controllable Desk Buddy robot capable of waving, blinking, and moving in response to keyboard input.
