Sprint 2 - Spec

Goals & Scope - Pre-sprint planning doc

t 2: The AI-SIM-C2 Integration Blue
-RL
JSBSim
C2 Fron
Sprint 2 - Goals...ope.pdf
15 Jan 2026, 11:18 AM

User Stories

1. **Training**
    a. Train RL agent with JSBSim's instance
        i. Include pop-up NFZs (plus plane appearing inside NFZ during training)
        ii. Use medium complexity scenario for training
        iii. Goal: Converging agent
        iv. Metrics for evaluation
        v. Version control
        vi. Reporting - agree on how to generate progression report
        vii. Integrate trainer code with JSBSim (implement ITrainer)
        viii. Support polygon as NFZ (always fixed) - Sprint 3 (low-priority)
2. **Runtime / inference**
    a. Separate SkySim from RL agent runner - Integrate with RL Agent
        i. Develop agent interface
        ii. RL Agent implements interface for tick() and observation
        iii. Develop protocol using gRPC and Protobuf
    b. Replace agent version for inference
        i. Refactor changes as required
    c. User able to create a scenario using scenario editor
        i. Store scenario in storage
    d. User able to run a created scenario
        i. User selects a scenario to run from C2
        ii. C2 sends scenario to SkySim
        iii. SkySim parses scenario and creates relevant mission+objects files

    iv. SkySim initializes the scenario (i.e. places airplane in its initial position)
    v. SkySim connects to an agent
    vi. SkySim starts sending periodic sim state (CONTROL LOOP STARTS)
    vii. Agent performs action
    viii. SkySim synchronizes action and implements AP change in Cessna310
    ix. User can view scenario in realtime in C2
  e. PM able to configure and run a scenario on QA pyland

3. Scenario file definition gaps C2 sends to SkySim:
    a. Playground definition / suppression_zone:
        i. leave only playground as the suppression_zone
        ii. Add support for defining the playground in C2
    b. Mission description → include if time allows (low priority)
    c. Engine_Type / Vehicle_Type - send hard-coded to SkySim
        i. engine_type = “JSBSim”
        ii. vehicle_type = “C310“
    d. Agent → not required in C2 YAML, remain hard-coded in mission.yaml, will migrate to RL Agent runner
    e. Decision points:
        i. Normalize or split mission files: Single YAML for all services or split into roles/responsibilities
        ii.
4. Scenarios to evaluate
    a. Simple scenario spec (7 NFZs), fixed during episode
    b. Medium
    c. Difficult
5. **Other points**
    a. Performance tracking during validation/inference:
        i. FLOPS
        ii. Memory usage
        iii. CPU/GPU usage
        iv. Disk size (policy etc)