

# Human Drone Interaction Study

## Project Definition: Human-Drone Interaction Study (Outdoor AR)

### 1. Overview

A research-driven interactive AR experience that simulates drone deliveries in public spaces. The goal is to evaluate how human recipients respond to operational uncertainty in drone behavior, with both recipients and bystanders participating simultaneously in a synchronized mixed-reality environment.

### 2. Core Components

#### 2.1 Participants

- **Recipient:** Primary user interacting directly with the drone.
- **Bystander:** Secondary observer whose perspective and perception are also studied.
- Both users are present in a real outdoor location and share a synchronized AR scene.

#### 2.2 AR Hardware & SDKs

- Devices: Meta Quest headsets (with passthrough AR)
- SDKs:
  - Meta All-In-One SDK
  - Unity Netcode for GameObjects (NGO)
  - Unity XR (for gesture input and raycasting)

#### 2.3 Scene Coordination

- **Shared Spatial Anchor** (via MR Utility Kit)
- **Collocation** ensures that both users align to the same world origin.
- Anchor will be manually oriented during setup to create a reliable forward-facing scene.

### 3. System Logic & Interaction

#### 3.1 Drone Behavior State Machine (Refined)

##### Primary Drone States

- Idle
- SessionStart
- FlightToRecipient
- EvaluateLandingConfidence
- Landing\_Autonomous (*high certainty*)
- Landing\_Confirmation (*gesture approval*)
- Landing\_Guidance (*user pointing*)
- LandingInProgress
- LandingSuccess

### Sub-States / Triggers (Not individually networked)

- SetLightState\_Certain
- SetLightState\_MinorUncertainty
- SetLightState\_MajorUncertainty
- WaitingForConfirmation
- WaitingForGuidance
- EvaluatingInput
- TimeoutFallback (*internal loop to ensure robustness—still leads to success*)
- Visual and animated feedback (e.g., blinking, winking, rotor spin) will be triggered locally via an `ExpressionController`, not managed as separate states.
- Lights and rotors are child GameObjects of the drone and reflect operational confidence via color and intensity.
- All transitions eventually lead to a successful landing to support the constraints of a controlled study.

## 3.2 Positioning Logic

- Spawn point calculated using:
  - Participant positions
  - Relative orientation
  - Scene layout constraints (avoid hovering over people or near obstacles)
- Adjustments made at runtime if space is tight or visibility is poor.

## 3.3 Animation & Control

- Animator controller handles expressive cues (hover, wobble, descend).
- Transforms driven by state transitions and dynamic calculations.
- Final landing spot determined by user pointing gesture with raycast to ground.

## 4. Networking & Synchronization

- One participant hosts the session.
- State changes are synchronized via `NetworkVariable<DroneState>`.
- Gesture and raycast inputs are submitted from clients via `ServerRpc`.
- Drone animations and light cues are handled locally based on the synchronized state.

## 5. Research Objectives

### 5.1 Goal

Explore how autonomous drones can manage edge cases in real-time by involving human input.

### 5.2 Research Questions

- RQ1: How does the recipient's guidance during drone deliveries in public places impact the user's trust and perceived safety?
- RQ2: How do drone-initiated guidance requests affect user acceptance and perceived workload?

## 6. Interaction Scenarios

### Independent Variable: Operational Disturbance Level

This variable dictates the degree of recipient involvement during the drone delivery process.

#### 1. No Disturbance (Low Involvement)

- Drone completes delivery fully autonomously.
- No user input is required.
- Use case: High confidence, ideal GPS, clear environment.

#### 2. Moderate Disturbance (Moderate Involvement)

- Drone pauses to request confirmation via a thumbs up/down gesture.
- User must acknowledge and approve landing.

- Use case: Moderate uncertainty—drone seeks human verification.

### 3. **Major Disturbance (High Involvement)**

- Drone requires recipient to point to a safe landing zone.
- User uses hand tracking to point and raycast target location.
- Use case: Low confidence, unclear landing zone, GPS ambiguity.

Each scenario activates specific branches in the drone's state machine and transitions toward a successful landing.

## 7. Next Steps

- Implement refined drone state machine using Unity and Netcode for GameObjects.
- Build gesture input system (thumbs up/down and pointing with raycast).
- Set up MR Utility Kit anchor and collocation test in an outdoor space.
- Create local `ExpressionController` to trigger animations/lights based on state.
- Finalize and test scenario logic to ensure clean, linear flows with no mission aborts.