Miltech Mission Hub

Building Detailed Missions: Objectives, Triggers, and Activities

A great mission guides the player through clear, meaningful steps. This guide helps you build a structured mission using objectives and triggers, and shows best practices for organizing your mission.

Controlling when objects are active

Before you start building your mission, it’s important to understand how activation works for mission objects. Knowing when objects are active or inactive helps you control the player’s experience and keeps your mission logic clear.

Activated means the object is “live” in your mission:

- **Objectives, markers, simobjects:** Visible to the player.
- **Triggers:** Actively checking their condition (e.g., area, timer, speed).

Deactivated means the object is completely dormant:

- Not visible and do nothing until activated. Triggers don’t check conditions or send events.

How to set the activated state

- Each object has an **Active On Start** setting (default: true)
- To activate or deactivate something during the mission, connect another object to its 'Activate' or 'Deactivate' action using connections.

Why Deactivate Objectives and Triggers at the Start?

- **Objectives:**\ Deactivating most objectives at the start, and then activating them one by one in sequence, ensures the player follows the intended mission flow and doesn't see or complete objectives out of order.
- **Triggers:**\ Deactivating triggers you don’t need right away prevents them from firing at unintended times (for example, a landing trigger before the player has completed a firefighting task).
- **Best Practice:**\ Activate objectives and triggers only when the player has reached the relevant stage. This approach makes your mission more robust and easier to manage as it grows in complexity.

How to create a structured mission

Plan your mission flow

Start by sketching a rough outline of your mission. What story do you want to tell? What are the main stages or challenges? Think about:

- **Scenario**: What’s the overall goal or setting?
- **World State**: Set up scenario settings such as time, weather, and traffic.
- **Spawn Point**: Where does the player begin?
- **Key Objectives**: What must the player accomplish? What are the steps they have to go through?

- **Triggers**: What should happen as the player progresses?

This “blueprint” helps you stay focused and ensures your mission is logical and rewarding.

Use objectives for mission structure

Objectives are the backbone of your mission flow. Each objective represents a step or milestone the player needs to complete, such as “Pick up water bucket,” “Fly to fire,” or “Land at base.”

Objectives show up as messages on the screen when active, and can point to a specific location on the map.

How to Use Objectives:

- **Place objectives on the map** in the order you want the player to complete them.
- **Deactivate objectives at the start** (except the first one). Activate each one in sequence as the previous is completed.
- **Connect objectives** so that when one is completed, the next is activated.
- **Define outcomes** for each objective:
- **Success**: What completes the objective (e.g., entering an area).
- **Failure**: What causes failure (e.g., timer runs out).
- **Abort**: What cancels the objective (optional).

Example Objective Flow:

1. "Taxi to runway" (completed by area trigger at runway start)
2. "Take off" (activated by previous success, completed by takeoff trigger)
3. "Fly to destination" (activated after takeoff, completed by reaching area or altitude)
4. "Land at base" (activated after destination, completed by landing trigger)
5. "Debrief" (activated at end)

Add and connect Triggers

A trigger is a map object that checks if something happens, and can activate or deactivate other objects. They are essential for advancing your mission.

Types of Triggers:

- **Circle & Rectangle Trigger:** Fires when the player enters/exits an area.
- **Timer Trigger:** Fires after a set amount of time.
- **Property Trigger:** Fires when something like speed or altitude is reached.
- **Takeoff & Landing Trigger:** Fires when the user leaves the ground or lands.
- **Parking Trigger:** Fires when the aircraft is stationary and on the ground. Optionally, it can wait for the aircraft to be shut down (engines off).

Best Practices:

- Deactivate triggers at the start if you don’t need them immediately.
- Activate them only when needed (e.g., activate the landing trigger after firefighting is done).
- Use triggers to complete, fail, or abort objectives (connect the trigger’s event to the objective’s action).

Example:

- Area trigger completes “Taxi to runway” objective.
- Takeoff trigger completes “Take off” objective.
- Timer trigger fails an objective if time runs out.

Keep your missions organized

- **Name objects clearly**: Use names that describe their purpose or location (e.g., "Lake Area Trigger", "Start Area Objective"). These names help you in the editor; players don’t see them.
- **Organize visually**: Group related objects together on the map for clarity.

Example: Firefighting Mission

Here’s an example mission flow using objectives and triggers:

1. **Briefing**
2. Objective: Hover above the bucket to pick it up (area trigger)
3. Objective: Fly to the lake and collect water (area trigger)
4. Objective: Fly to the fire and drop water (area trigger)
5. Objective: Return to the lake if more water is needed (area trigger)
6. Objective: With all fires out, fly back and park at base (parking trigger)
7. **Debrief and End Mission**

Final Tips

- Start simple: Focus on objectives and triggers first.
- Build in sequence: Activate each objective as the previous one completes.
- Name everything for easy editing.
- Group related objects together on the map.

By following these steps, you’ll be able to create clear, engaging missions that guide players through every stage of your scenario.

Next steps

This guide covers the core concepts of mission flow: objectives, connections, triggers, and routes. Once you’re comfortable, you can explore advanced topics like loops to create even more dynamic missions.