

In-Player Quiz & Poll Design and Tracking Checklist

Design a question that teaches, score it right, and make sure the answer is actually tracked. The companion article explains every line.

A. Design - a quiz that builds knowledge, not one that just interrupts

- Pick the simplest standard type that tests the idea (true-false / choice for in-flow checks)
- Require recall, not recognition - an obvious-from-the-wording question does no work
- Give immediate, specific feedback right after the answer, while the topic is in mind
- Keep in-flow checks low-stakes; save weighted, high-stakes scoring for an end-of-module test

B. Placement and scoring - decide what the question can promise

- Inline (non-blocking) = momentum, but cannot certify the learner demonstrated knowledge
- Gate (blocking) = ties completion to a correct answer; use only on questions that must pass
- Score = raw (7/10) and scaled (0-1) vs a pass threshold; weight critical questions higher
- Decide attempts: journal every retry (better analytics) or overwrite the last one

C. Tracking - an answer is only data if the player reports it

- Emit an xAPI 'answered' statement (interactive video / cmi5 / beyond the LMS)
- Or write a SCORM cmi.interactions entry (SCORM 2004 keeps the question text; 1.2 does not)
- Polls have no correct answer - record a NEUTRAL result, no correctResponsesPattern
- Pre-flight: launch against a real LMS/LRS and confirm each question records id, response, result

Pre-flight: accessibility before you ship (WCAG 2.1 AA)

Every option, input, and submit control must be keyboard-operable with a visible focus state (SC 2.1.1) and expose its name, role, and state to assistive technology (SC 4.1.2). If a question is timed, the limit must be adjustable - off, 10x extension, or a warning and a chance to extend (SC 2.2.1). A quiz a public-sector buyer cannot certify accessible is one you cannot sell. Build accessibility in with the quiz, not after.