

*Some common issues with the user guide (not true for all, but in each case, the majority of reports missed these points):*

If there is a contents page (and there should be) then don't waste space in the introduction reiterating the contents.

Should provide an overview of the whole system, e.g. "X is a robot that plays football on specific sized and coloured pitches, with an overhead camera, in a team of two. The robot control runs on a separate PC, using the camera feed, and RF communication to an Arduino on the robot." Ideally with a diagram showing the complete set up, not just the robot.

You could have tested the adequacy of your instructions for installation and running of software on an actual 1<sup>st</sup> year CS student, to flag up any missed steps or ambiguity.

Good usability decisions: includes factors such as scripts that do all the necessary set-up, an explanation of how to just run the robot (without opening the development interface) as well as how to reprogramme it if desired. Simple and obvious placement of the batteries. Redesigned or additional GUIs to help with set up, and clear explanation of how this has advanced from the software you started with.

Adequate citation: be clear about the parts that are derivative, e.g., that the interface software is based on a previous system.

Robot interaction:

- Not just how and where to plug in batteries but tips on how long they last or how often should be replaced, checking the level with a monitor.
- Should mention need to position the top plate correctly, with dots aligned a particular way, as this is crucial to game play.
- Should include any typical mechanical tweaks needed, e.g. checking wheel or sensor alignments, any known weak spots in the construction, or wires that come loose.

Software interaction should include describing what expect to see in monitoring windows (e.g. passed commands). Similarly, explain what should see (directory structure, outline of contents) after the git clone.

Include how to turn everything off.

Good to create/mention different modes for robot play (solo vs joint, defence vs attack).

Provide a clear description of what the robot *should* be doing in the game situation, so that the user knows when the behaviour is wrong and needs troubleshooting. E.g. are there situations where it might remain motionless for a while, or spin, or do something else that might appear to be a problem but is not? Should the robot ever intentionally kick in directions other than the goal e.g. to pass the ball, or are misaligned kicks errors? How should kicking be coordinated with grabbing and orienting? Can the robot get unstuck from an edge or corner without help?

Troubleshooting is most helpful when the problem is very specifically described (e.g. robot does X in situation Y, or communication fails with message Z), the likely cause is stated, and then the solution(s) listed. Also organising this in a logical fashion to make it easy to find the relevant fix.