

Metrics and QA

An important concern in game development is, of course, that the game must be *fun*.

- a) The first founder argues in favor of doing user studies on prototypes of the game, and asking participants to fill out a survey about how much fun they had while playing it; the second founder argues instead in favor of logging and analyzing metrics on the game once it's released (like looking at the app store ratings, or anonymously collecting metrics like time spent playing). What are the tradeoffs, in terms of arguments for or against each, for each of these options?

- a. Conduct lab studies of a prototype:

- b. Analyze measurements on the deployed game

- b) The third founder disagrees with the first two, arguing that "fun" cannot be verified through testing or surveys and must instead be left to the intuition of the game designer. The founder says those surveys and app store ratings really don't capture the core of "fun"; they're pointless!" Do you agree or disagree? Why or why not?

- c) For each of the following quality attributes, give a good metric that you can use to determine if the implemented game is satisfying the associated quality requirement:

- a. Performance:

b. Overall Usability:

- d) Your teammate gets cranky during this discussion, and argues that “overall usability” cannot be verified or measured using metrics and must instead be left to the intuition of the designer. Your teammate says “none of these options really capture the core idea of ‘usability’, it’s pointless!” Do you agree or disagree? Why or why not?

Requirements:

- a) Succinctly state one relevant system-level functional requirement for the high-score tracking functionality.
- b) Anticipating that more future games might need a server backend, the team considers extensibility as an important quality attribute for the server implementation. Name and describe a metric that you can use to provide evidence to management that the server is meeting its quality goal.
- c) A team member has scheduled a requirements elicitation interview with the head of Public Relations at Heinz. They have already prepared a number of open-ended questions, but are considering other materials you might bring with you to facilitate the interview. For each of the below, explain whether you would or would not recommend to use it in the interview, and why/why not:
- a. An architecture diagram showing how you plan to minimize the risk of cheating
 - b. A phone with the latest version of Candy Crush, a popular mobile game in which users match candies and can compete with their friends as identified through Facebook.

- d) Kraft Heinz's legal team is worried about the liability involved in users playing the game while walking. They ask for a feature to pause the game if the phone detects that the player is moving. You are concerned that the team might fall into the trap of denial by vagueness (i.e., imply (vaguely) that machine descriptions are actually those of the world). Explain possible bad consequences of not clearly separating world and machine phenomena in this scenario and how the team can avoid such problems.

Appendix: Scenario description

The following fictional scenario will be the basis of most questions within this exam. You may detach this page from the exam.

The Company.

A local computer games company has asked you to consult because they are facing a number of problems. The company was founded by three high-school friends who liked computer games and always wanted to make their own. After a couple of attempts, they found their market niche in creating simple smartphone games, mostly jump and run and puzzle games that often use the phone's gyro sensors and microphone as user input. The games are cheap and easy to produce quickly. The team hopes that one of these games becomes a hit some day. The three friends and two additional developers produce between one and three games each month, which generates (barely) sufficient income to pay the bills. However, the fast pace leaves the developers exhausted and unsatisfied with their (often low quality) products. They do all marketing themselves; the only external help is for accounting. The three founders have no formal computer science training and are self-taught enthusiasts; the other two developers have college degrees but little practical experience.

The Tomato Crusher Game.

At a local party, the team has come in contact with some people working for the Kraft Heinz Company food conglomerate. Kraft Heinz is frequently releasing branded games for free as part of their advertising campaigns, and they are interested in possibly working with this small, local games company. The three friends are excited about this possibility as it could provide a stable revenue stream, but they are also concerned about the higher stakes. In a brainstorming meeting with the Kraft Heinz marketing people, the idea of the Tomato Crusher game is born: By tilting their phone, players can roll a large round stone through a labyrinth to crush tomatoes hiding in the labyrinth (roughly similar to PacMan). The player gets points for every tomato crushed in a given time window.

The Server.

None of the previous games needed a dedicated server backend. However, for the Tomato Crusher game, a server is needed for multiple reasons. First, users should be able to create accounts. Second, high scores should be shared and a user should be able to compare their own high score to that of other players. Third, Kraft Heinz wants to have analytics of how many people play the game how often. Fourth, high scores should also be shown on a web page for a global ranking that might be used in advertisement campaigns. Specifically, Kraft Heinz wants to provide \$20,000 prize money for high scores or specific achievements