

3.5.1 Introduction

We've made a ton of progress in our Robot Gladiators game! Now that our MVP, the `shop()` function, and object refactoring is finished, we can use the remaining time to add polish and shine to our game. We don't have a lot of time left, so we'll add small improvements one by one rather than trying to deliver everything at once and risk missing the deadline.

It would be a shame if the game were to crash while the judges were assessing it. To prevent this, we shared our game with other attendees to test it and get their feedback.

ON THE JOB

In the real world, releasing a software product to a small group of people to get feedback and catch errors before it's made available to the public is known as **beta testing**.

Our game testers gave high praise for the gameplay and entertainment value, but they found the following bugs:

- Empty player names are accepted.

- Pressing the Cancel button in the player name prompt assigns "null" as the player's name.
- Empty or mixed-case (e.g., Skip, Sklp, skiP, etc.) input to the fight-or-skip prompt results in the `fight` option.

They also asked for some new features:

- Ability to save the high score.
- Randomize who attacks first in each confrontation.
- Simplify the input process for the `shop()` prompt to lessen the amount of typing needed to reply to prompts.

As we complete the tasks at hand in this lesson, we will be learning some important concepts such as:

- Learn how to use the Web Storage API to store data.
- Use the `while` loop to validate user prompts.
- More ways to use the `Math.random()` function to add variability to the game.

© 2020 Trilogy Education Services, a 2U, Inc. brand. All Rights Reserved.