Unity QA Position Assessment

Mathias Wagner Nielsen and Simonas Holcmann

April 5, 2022

1 Purpose

This is the specification for the Unity QA Position Assessment. The goal of this document is to facilitate a proper assessment of the applicant's ability to contribute to our Unity Development team. This document is intended as a suggestion for how to demonstrate your fitness for the project to us.

2 Factorial Calculator

We are building a Factorial Calculator for Mr. Ackermann, who has a very keen eye for detail paired with the highest level of quality standards. Our junior developer looked at the specifications of the application, and bravely exclaimed that he will finish the whole application within a day. As a team we trust in his abilities, however all work produced by our development team must pass the QA layer in order to reach the client.

3 Task

It is up to you to ensure that the application is the in the best shape that it can be. The most important feature here is that the factorial calculator calculates the factorial function accurately! In order to achieve the confidence to ship this application to Mr.Ackermann, we need to back it up with tests find any issues that may reside in the application, report said issues and suggest ways to solve the issues.

Automated testing:

1. Ensure that the function in Factorial Calculator class returns a correct factorial number, by writing a unit test (Editor mode).
2. Ensure that the input field, when given an integer, calculates the factorial and outputs it to the text field below the input field by writing an integration test (Play mode).

3. From your experience, what else might go wrong in this application? Think of possible scenarios, and write integration tests that assert the outcomes and possibilities (2 - 3 tests).

Exploratory testing and documentation:

1. Explore the application. What issues can you find with it? Write a defect report for each issue and inconsistency you manage to find.
2. Create a document with test cases. Document your findings, expected outcomes and actual outcomes. You can find a good explanation of the 7 column test cases here.

Feel free to use free software tools such as Trello for documenting defects, test cases and issue reports, however make sure you share them with the Emperia team!

4 Forking

When writing your submission, please fork this project, clone your own fork, and commit your code and documentation files to that. This way, you can share the forked repository with us.