4.1 Testing role-play

0. Overview

In this unit we will be using testing role-play activities to introduce you to concepts without needing to jump straight into technical details. **This is not an activity you can do outside of your applied class, so you need to attend**.

As a reminder of how this works, in general, there are two roles:

- The thing being tested ("machine").
  People taking this role will act as a machine or piece of software. They will be given instructions showing them how they should react to inputs. Their job is to listen to the inputs given by the person doing the testing, and tell them what the output is.
- The person doing the testing ("tester").
  People taking this role will act as a person testing the machine or piece of software. They will be given a description of what they are testing, the inputs they can use, and the general expected behaviour of what they are testing.

The job of the tester is to try and find the bugs within the machine.

This activity relates to ULOs 2 and 3.

2. Testing role-play 3

Overview

For this role-play game you will split into groups, with at least 2 people taking each of the roles (machine and tester). **Instructions will be provided to you in class.**

As you play the game, regardless of which role you are taking, document the tests being run in a spreadsheet with the following column headings:

- Test ID
- Description
- Input
- Expected
- Actual
- Pass/Fail

Once the testers feel they have found a bug, write a bug report. Use the following format:

- Description
- Steps to reproduce
- Expected behaviour
- Actual behaviour
- Priority (use the Google issue priority ratings)

At the end of the activity add any notes, your documented tests, and your bug reports to the "Applied 4" folder in your unit repository. Make sure you add, commit, and push these files to Gitlab at the end of the activity.

Testing approach

As you have learned about some specific testing techniques this week, it's time to put them into practice.

- For the **first 10 minutes**, use a purely **random** approach. Use a virtual dice rolling tool (or real dice if you have them) to determine which inputs you use.
- For the **remaining time, pick any of the other testing approaches (or a combination of them)** that you learned about this week. As a reminder, these were: decision tables, category partitioning, equivalence partitioning, and boundary value analysis.