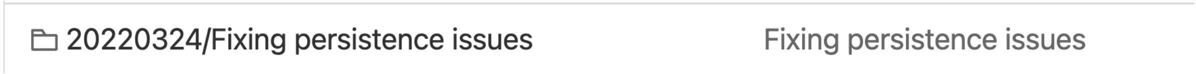
Comments on Test Spec for CS22120 Group 11

Quality Assurance

Repo:
Why a .DS_Store file at the top level. Use .gitignore to screen out certain files.
I've seen this Mac file at other levels too. I said this last time too.

There are still inconsistencies in the number of blog files for different member of the group. Put in blank zero hours ones for anyone who is missing.

This doesn't look right to me:



20220324/Fixing persistence issues   Fixing persistence issues

Formatting of test spec:
Should be version 1.0 for first release. Use 0.1, 0.2 etc for earlier drafts. History table should reflect this.

References section is not correct. This should be for references to other documents, not to data. You can put data in a separate appendix if you wish.

Test Specification

Section 1 is incomplete. See examples from the group project standards
The References section should have references to the requirements spec, SE.QA.06, the UI spec (since this will also influence the test spec, such as error conditions).

I would expect the first test to check that on program start we see x,y,z…

001: Use actual names for each player to make test repeatability easier.
002: it's the players assigning names, not the system.
003: use explicit data.

004: don't you need to run the game several times to check for randomness? Be explicit. This applies to other tests where randomness is being checked.

005: How will you see this if it is hidden data?
I'm assuming you're going to use some kind of test mode here so that you can see the 36 cards e.g. on the command line? If so make that clear somewhere. An alternative approach would be to use a unit test. Again, if that is the intention, then make that clear.
I wonder whether test mode might allow the injection of data to set things up prior to a test? That might though make the code much more complex.

009 and 010: I wonder whether this is best interleaved with FR13 tests where chance cards are being drawn and returned?

The tests to do with the number of points for treasure. How will you test those visually?

Many of the tests are vague, e.g. 022 or 025. Use explicit data. I don’t know how to check without something happening first: starting state: player has 1 treasure and 2 black crew cards and 3 red crew cards, they fight another player with xxxx and lose, they should now have Y amount treasure.

You need to have explicit tests for FR13 based on the requirements spec appendix. Sorry. These cards describe behaviour and it needs to be tested.

I didn’t look at every test but a common theme is the lack of actual explicit data to aid test repeatability.

Make sure you have identified equivalence partitions and boundary cases as discussed in the lecture on system testing. E.g. taking last crew card works.