Chaps Challenge Report
300492683, sarilroya

My role within the Chaps Challenge was Monkey Testing. This means I was responsible for generating tests to call public methods within Application and Maze modules.

Commit-Urls	Description
https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team38/chapschallenge/-/commit/7b07ac2fed28a507ac69d23a30120df6d7cb23f4	This commit consisted of modifying some of my already written monkey tests, however this shows the tests that I've written and kept through to the final submission. The tests in this URL were mainly based on the public methods in the Maze module; these included testing the movement, tiles, treasure and bomb. These tests I wrote around writing with Assertions.
https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team38/chapschallenge/-/commit/4a96927292cf171df732e0ac867794f3e7c2863e	This commit consisted of writing tests for testing restart levels for level 1 and level 2. These tests trigger interrupted exceptions. The tests makes sure that restarting each level works as expected, by using assertions to ensure its function.
https://gitlab.ecs.vuw.ac.nz/course-work/swen225/2020/groupproject/team38/chapschallenge/-/commit/8ee224c95d33c04ef902226f9ff44f623d35f7c8	This commit was one of the few final commits for my module. This commit consisted of writing tests for randomized movement tests, and testing the key listener within gui and modifying and removing some of the already written tests. I wrote randomized movement tests for both levels 1 and 2; where the test will complete once chap has covered the whole level. These tests call out for general errors in the maze. I also wrote tests for checking the Key Listener within GUI works as expected. These tests trigger interrupted exceptions. Although this commit includes testing for renderer and record modules, these are later deleted as I was only able to generate tests for maze and application.

Group 38:

- Gelan Ejeta (ejetagela) ejetagela@ecs.vuw.ac.nz --- 5
- Morgan Hucker (huckermorg) huckermorg@ecs.vuw.ac.nz --- 5
- Zach Kingsford (kingsfzach) kingsfzach@ecs.vuw.ac.nz --- 5
- Elgene Leo Anthony (leoanelge) leoanelge@ecs.vuw.ac.nz --- 5
- Royan Saril (sarilroya) sarilroya@ecs.vuw.ac.nz --- 3
- Gimani Telikada Palliyaguruge Weerasena (telikagima) telikagima@ecs.vuw.ac.nz --- 5

What knowledge and/or experience have I gained?

The knowledge that I have gained by doing this project, I am now a bit more comfortable and familiar with using GitLab and IntelliJ in terms of commits and merging. In previous assignments from other courses, I did not do amazingly when it came to writing tests however this project has slightly helped me with my work. Another thing that I know I must be more aware of is my time management, as I did not make use of the time period between the Integration Day and the project deadline. Due to this disorganized work, I worked on my module last minute rounding up to the last few days left.

What were the major challenges, and how did we solve them?

One of the major challenges was definitely working with GitLab and learning how to use it to ensure that there were no accidental changes that have been made when merging and pushing commits. We solved this by letting the group know whenever we will push a commit, this is so that everyone saves their current copy of their code just in case changes were made with the push. As well as ensuring that when each member is pushing their commit, only to push their module and no one else's. Another problem that occurred was the communication with each other, we did not

communicate within GitLab and utilize it as much as we should've. This was most likely due to the unfamiliarity everyone had using GitLab, instead most of our discussion was done using Discord. Although it may not be as organized and well categorized, we use this for group calls.

Which technologies and methods worked for me and the team, and which didn't, and why?
The IDE that we all used was IntelliJ, we chose this because we are all familiar with this application and it's very simple to link our GitLab with Intelli, and its useful features. Overall IntelliJ was a good tech tool to write up our project, and with the already built in test coverage this enabled me to check my tests past and check the line coverage of my tests along the way. It was also simple to download plugins within IntelliJ what were needed in the project, these were SpotBugs.

We had quite a bit of issues working with GitLab as we were all not very comfortable with this development tool. The main issue we had with GitLab is the commits, this was the pushing and pulling our parts to the original branch. Whenever we decided to commit our progress and then push it to the branch, there were issues with merging other branches. Whenever merges occurred the original branch would get messed up with either containing the old work of another's module, or accidentally deleting new work of another's module, or work that been commented out due to other problems; for example not downloading a jar file. This issue happened many times and was frustrating for everyone. Moving forward we had to make sure to let the group know if anyone is pushing their commit, and that each person saves their work. We also had to make sure that we pull the current branch frequently, and to save our work as well.

Discuss how you used one particular design pattern or code contract in your module. What were the pros and cons of using the design pattern or contract in the context of this project?
How I implemented Monkey was done using a design by code contract. A majority of my tests was based around the use of assertions and runtime exceptions. I chose this because of the economic benefits, as described in the lecture slides, where fixing bugs is much easier later on. The use of assert* methods in my module are used to check post-conditions and invariants. The pros of using this design was for checking the state of the game after the tests, as well as the implementation of runtime checks and assertions are simple to do. However, from testing my monkey module as a whole I have noticed that overtime as the tests pass this causes the game continue running in the back and will slow down IntelliJ checks. This then forces me to have to terminate the current tests, and rerun the failed tests. Because of this it does not appear with the total line coverage, only the most recent tests that passed. Although I did not use static analysis alongside writing my tests due to the risk of false positives, I will though ensure to have a go at it in future testing.

What would I do differently if I had to do this project again?
What I would do differently if I did this project again was definitely ask for more help and guidance from outside my group. This was because my group members did not feel very confident with writing tests themselves, although they did help me understand my role a little more towards the end of the project. Because of my poor decisions this made my monkey module finish at an inadequate level. The code contract is used to enforce contracts with tests. What I would again do differently is include more variation of types of testing that I'm trying to do, checks on for example; enforcing Pre/Post Conditions, using Assumptions with Assertions and Class Invariants. With these enforcements used in my monkey module, the testing would be much more refined.

What should the team do differently if we had to do this project again?
What the team should do differently if we did this project again together is to increase the communication with each other. Because of the lack of communication between the members there was a lot of rushing and final calling within the last hours of the deadline. We had quite minimal communication during the weeks of the project rounding up to the final deadline. The last couple days of the deadline there were plenty of calls within the group, and very last minute changes that were made with the project. If we had consistently communicated with each other throughout the project would've been polished much better as what the final submission of the project was. This means that we would've been up to date with each other's plans.