

# Team iPatch

## Assessment 3: Change report

---

Christian Pardillo Laursen

Filip Makosza

Joseph Leigh

Mingxuan Weng

Oliver Relph

## **Change management**

One of the first steps of change management detailed by the IEEE 828-2012 - IEEE Standard for Configuration Management in Systems and Software Engineering is the identification of configuration items in the project [1]. In order to identify the configuration items in the first week of assessment 3 we decided to individually inspect the code and then during our first meeting we reviewed the tests which group The Element of SEPRise failed to achieve by the end of Assessment 2 [2].

During the meeting we discussed the which failed tests were required to meet the Assessment 2 requirements and which were extra requirements created by the team in order to pad out their game. From there we were able to equally distribute implementation and documentation tasks amongst the members of the group. The implementation changes can be found in the changes document on our groups Assessment 3 page[3].

Our version control was managed by GitHub, we forked a version of team The Element of SEPRise's code for development use. All changes the team committed were first submitted to the development branch for testing and code review before being merged to the master branch by the group leader. We used a continuous integration and deployment system called Travis CI [4] in order to test code during the merging process to check for code conflicts and errors. Task progress was handled informally by our group leader who checked in with group members regularly to make sure that the tasks were being completed.

We left the majority of our documentation up until the end of the Assessment 3 deadline so that all of the appropriate requisites were in order before starting the report, however we collected evidence for the testing document as we progressed with the assessment for use when writing up the appropriate documentation.

## Testing report

In the assessment 3, we still test the game and code by the mixture of black box testing and white box testing. The black box testing will test most of the feature and function of the game without considering the internal code and the white box testing will test the code inside and make sure these will not be impacted by the code written after. Since we chose the project of The Element of SEPRise, we also inherited their project documentation. For the black box testing, we still keep their previous black box testing result and update it. And we also retain the JUnit test they write before.

In assessment 2, several black testing failed mostly because of that these features were not implemented at that time. The test which is highlighted in green passed in assessment 3. Since we need to ensure the completeness of the game and the limited time, we decided not to implement the questing system and loot system.(Test 23,24,28,29 has been deleted)

Test ID	Reason of failure
2,4,5	Menu not yet implemented (This menu has been implemented now)
23,24	Questing system not yet implemented (deleted)
25,26	Ship upgrade not yet implemented(This shop has been implemented now)
27	Minigame not yet implemented (minigame has been implemented now)
28,29	Loot system not yet implemented (deleted)

### Statistics

#### Black box testing

All the initial black box testing has been checked again to make sure that the additional functions and features had not rendered the effectiveness of the old testing. As the table above, the test 2,5,25,26,27 are now re-tested again and pass the black box testing, test 23,24,28 and 29 still fail since it has not implemented. Out of all the additional 10 main game black box testing, all of them pass. And the 3 minigame testing also pass.

#### White box testing

We still use the initial 7 unit test that the element of SEPRise provided to ensure that the NPC boat AI still work as designed and check the AI logic, boat navigation and shape operations. All of the 7 unit tests still pass after our implementation.

### Testing evidence

[https://drive.google.com/open?id=1WUZsXoEfitqGdLBP6r\\_nDzhFOH4Ikszl-rJv3yLX4Tc](https://drive.google.com/open?id=1WUZsXoEfitqGdLBP6r_nDzhFOH4Ikszl-rJv3yLX4Tc)

## Method and plans

After choosing the project of The element of SEPRise, we found that the method and general plan is very similar to ours so we will still stick to our plan and method. But we have to change the role of the team member to ensure that we will complete the game and get familiar with the new developing tool. And since we only have about 3 weeks to implement all the functions and features of the game, most of us have focused on the coding and implementation of the project.

Due to the fact that we will continue the project of the Element of SEPRise, the most difficult challenge was that we need to switch the development tool from Jmonkey engine that we have used in the previous two assessment to the Libgdx engine. And we planned to have more meeting from once a week to twice a week to know the progress of each other. The team leader would know all the progress of the team member and might change or add the work and task to each member.

Except for the developing tool, the collaboration tools still remained unchanged since all the tools we were using have already done what we expected and worked well. Because more members have participated in the coding, so the Github has been used more frequently than the previous assessment. Github can storage for our code and allowed us to easily understand the code the other group member write and all the functions and features they add. Every member has their own branch on the Github and the code will merge to the master branch after checking with the team leader.

The plan for assessment 4 can be found [here](#). It is constructed around the need for planning the changes, to which the first two weeks will be dedicated. We recognised that the easter holidays might hinder progress so implementation time extends into the second week of summer term. Testing will be carried out as the game is developed, and the documents will be produced when everything is ready to be summed up. Finally, we will prepare for the presentation when all other work is handed in.

## References

- 1) <https://standards.ieee.org/standard/828-2012.html>
- 2) <https://sepr4.github.io/web/submission/assessment2/Test2.pdf>
- 3) <http://team-ipatch.github.io/docs/assessment3/changes.pdf>

4) <https://github.com/travis-ci/travis-ci>