Module	SEPR
Year	2019/20
Assessment	3
Team	Krojan Horse Expanding on NP Studios’ Assessment 2
Members	Abel Kent, Toprak Onat Sen, Vinu Jey, Tom Emmerson, Diana Udrescu, Toby Popov, Jamie Todd
Deliverable	Change Report

Change management outline

As we worked on NPStudio's project, we made some changes to their methods and tools that would allow us to work with maximum efficiency. We decided to keep using Monday to track our progress and split the work within the team, as opposed to GitKraken Glo Boards used by NPStudios, as we were more used to it and found that it matched our workflow best.

Our work for this assessment was based on the following structure: "Identify potential change -> Analyse change request -> Evaluate Change -> Plan change -> Implement change -> Review and close change" [1] as we identified as a sensible approach for this project.

Code

NPStudios used very similar coding styles and methodology to our own, so we could continue using our previous method of coding and commenting (that of javadoc standard).

We continued to develop their code using a Git Repo, so we were able to keep track of changes made by various members of our team using flexible branching. New features and changes in the code were regularly commented to keep track of what was being written at this stage of the assessment with "#assessment3". As for tracking new assets added to the project we aimed for including a short message with every push of new code.

Documentation

In order to preserve the current documentation structure, we have followed NPStudio's approach: each member was allocated a task and a deadline in the form of a chart (https://npstudios.github.io/files/Updated_Plans.pdf - page 2) except our team achieved this using a Monday board (Assessment 3 Board) and Messenger as our collaboration tool. However, towards the end, the team was collectively making changes to the documents as we were reviewing them by commenting sections in the documents using the "comment" option in Google Drive. New additions to NPStudio's artifacts are highlighted yellow to evidentiate them properly and locate them easily.

Deliverables

The deliverables received from NP for this stage of the project remain largely unchanged at this point in time. We have been using them as reference points, examining their development process, and drawing from particular points, notably their project requirements, which we used when justifying changes we made to their implementation.

Testing, methods and planning

Testing

Upon receiving the game, we played it to see if, outside of their tests, we could spot any errors from a player's point of view, rather than a developer. Doing this showed us numerous errors we needed to fix, more on this can be read in the implementation report.

After this we then ran the tests used by NP to see how reflective their test coverage is of their code. The tests used by NP were of the JUnit style, and tested the pure functions of the game, as opposed to those based on external modules.. The tests written were of little use in ascertaining general playability of the game as the tests were covering parts of the code that did not need testing (i.e getters and setters, default variables etc).

The code coverage of the tests they did develop amounted to only 5% despite claims in their documentation and code comments stating otherwise.
The code that was developed was not created in a way that was easy to test, there was *very* high coupling between components and contained few pure functions which would allow testing. Due to this and the previous testing flaws stated above the decision was made to forgo improving testing coverage etc.. as this would take too much time away from core development given the short development period.

Instead of working on automated tests we moved to manual testing, where we would test each of the aspects and functions of the game through gameplay.

Testing Examples

A couple of examples of manual testing we completed are testing features of the game such as fire station refilling and the tetris minigame. The fire station testing allowed us to see that the the distance between the trucks and the fire station is correctly calculated, and refills and repairs at a steady rate.
Fire Station testing

The tetris testing allows us to see that; pieces rotate properly and maintain their shape, can't rotate into other pieces, move down each frame, clears the correct number of rows, loses properly when at the top of the screen etc..
Tetris testing

These methods allowed us to find numerous bugs in the code prior to our development and during it as well.

Methods and plans

For this stage of the project, we changed the operational standards of NP in the following ways:

Team organisation

We continued to use an Agile SCRUM team organisation strategy for this stage of the project, as we had experience with it from the previous 2 stages, and felt that it would allow us to quickly develop the code given to us, particularly in the short time frame we had for this stage of it.

We divided the work for this stage of the project amongst group members by means of a discussion to isolate what tasks each person would be most suited for. This ended up as follows:

Implementation: Jamie, Tom, Toby, Vinu, Diana
Change report: Abel, Toprak, Diana
Implementation report: Abel, Toprak, Jamie, Toby

Organisational tools

We continued to use the tools outlined in the previous section of the project, as they have allowed us to work quickly and effectively, as opposed to those developed by NP, due to our relative unfamiliarity with them. During this stage of the assessment we relied more heavily on those which allowed for task allocation and online communication since a number of our group were unable to always attend meetings in person due to external commitments, and the ability to allocate tasks remotely was of great use to us.

Developmental tools

NP used a very similar toolset, including the same IDE and language, to our own when developing their version of Kroy, so we did not have to make any significant changes in terms of the development tools used.

We did continue to use Piskel and Pyxel to design sprites and tile maps respectively, tools which NP did not use in their own development process, but we found necessary in order to develop their game further.

Gantt Chart for Assessment 4:
Gantt Chart

Bibliography

1. *“Change management (engineering)”*, available at: https://en.wikipedia.org/wiki/Change_management_(engineering)