a) Formal Approaches to Change Management Summary

During our team's retrospective meeting, before the group presentation, we discussed the criteria on which we would judge each prospective project. We were fortunate that the vast majority of other teams in our cohort had also used LibGDX, as we had done. We decided that, where possible, we would like to stick to a real-time game design and avoid reliance on Box2D. This is in order to reduce the time which would need to be spent learning a new engine or turn-based game design, in accordance with our risk mitigation. We also rejected any games which had failed to meet any of the assessment 2 deliverables, as to avoid spending time catching up on this functionality.

We began to look through the code of the remaining games, judging their architecture and reviewing their documentation. We were especially mindful of the simplicity of the design decisions that the teams had made; how this was reflected in their requirements and implementation, as well as how well we felt that their decisions fit with the customer specifications.

As a result of the above criteria, we decided to continue the project of team '*DICY Cat'.* We initially decided that it was best to tackle the changeover as a series of smaller teams with discrete focuses in addition to our pre-assigned responsibilities [found in previous deliverable], such as risk management. These being:

Team Members	**Member's Roles**	**Main Focus & Responsibilities**
Jordan Charles, Golnar Kaviani	Dev; Dev	Feature implementation
Samuel Hutchings, Jack Thoo-Tinsley	WD & PDM; GRM	Refactoring and code maintenance
Chloe Hodgson, Tamour Atlaf	PJM & SR; TM	Documentation maintenance

Working in pairs allowed us to focus on the details of the project, which was beneficial in ensuring our changeover was thorough and that we understood all aspects of the inherited project.

Since *DICY Cat*'s requirements and methods were constructed from the same product brief as our previous project, understanding the overarching vision was simple. In order to notice interpretation differences, our team each played the game and noted down features which differed from our game, such as the inclusion of a scoring system (when we interviewed the customer it was discussed that a scoring system would not be necessary); or things which we felt could be improved upon, such as the need for map zooming capabilities. In each case, we reviewed if this feature directly conflicted with any requirements/brief specifications. If not, it remained in the game in order to best carry on the vision of the previous team, or we implemented the new feature. To maintain consistency with the code and avoid large overhauls, the development groups built upon the class structures which were already in place, and the main architecture was kept too.

Changes made to planning, organisation and risk management were minor due to the shared use of an agile methodology between our team and *DICY Cat*. Our team was able to continue our previous model/structure of team working and development which we believe has proven successful in the previous assessments.

Due to a limitation on Google Sites and to avoid sharing security information, we were unable to continue Dicy Cat's website and so have continued to use our own.

b) i. Testing Report Changes

Updated Testing Justification and Testing Report
Updated Testing Materials
Updated Traceability Matrix

Additions are marked in yellow, reductions are also ~~strike-through~~.

It was decided by the testing team to keep the tests that DicyCat had because their testing did not have any major problems however, the testing team did decide to add further tests for the functional requirements and user requirements. This is because the testing team felt *DicyCat*'s approach to only have play testing was not robust enough to get rid of problems with the game - this was evident in the gameplay. When we received the project as it suffered from a memory leak and other small issues, for example a bug which allowed the fire trucks to be driven through buildings. They did acknowledge in their project that they did not have enough testing, and the testing team agreed that it was necessary to add further testing. Memory leaks would have been discovered if there was non-functional testing in the project. For this reason we added tests to test the non-functional requirements in the form of acceptance tests. We also added tests that reflect the updates made in phase 3 of the project, such as tests for the minigame, tests for the patrols and tests for the fire station being able to be destroyed. Tests were also updated in accordance with the requirements.

Another problem with DicyCat's testing approach was the lack of automated testing. They failed to add any form of whitebox testing. The reason for this was that *DicyCat*'s architecture made it difficult to break any dependencies between the classes and methods. In their testing report they also state they miscalculated time for developing tests which was another reason for the lack of automated unit testing. To solve this issue, a large amount of the code was refactored to reduce dependencies making it easier to implement unit testing. We then implemented JUnit testing to make the testing more robust. Our JUnit testing focussed on testing entities as a lot of the game logic relies on entities. To make sure our game logic is as robust as it can be and to make sure there are no problems with the code we decided that implementing JUnit tests was critical.

Apart from these major issues, spelling mistakes were fixed in the testing report, the Traceability Matrix was updated to reflect the new tests and the testing method and approaches were updated.

ii. Method and Plans Changes

Updated Method and Plans
Gantt Chart

Additions are marked in yellow, reductions are also ~~strike-through~~.

Since *DICY Cat* also used an agile approach and specifically a SCRUM framework as we have done in previous assessments, minimal changes were needed to the methodology documentation upon review as we agreed with the writing. The specification of a “small group of 7 people " was changed to “6 people " to accurately describe our team.

In the *development and collaboration tools* section, we made a couple of changes to the communication choices. The first being that we chose to use Slack instead of Discord as we have been using Slack for the previous assessments and therefore already have a system of pre-established conversation channels. It was decided that moving all of this over to a new Discord would be a time sink. Because of this lack of Discord, we also use Facebook Messenger for video calling. The description of reasoning behind these tools is still applicable as the tools are so similar and therefore did not require changing. *DICY Cat* had previously taken all of their assets from online sources, but we decided to make the additional assets needed for this phase bespokely in order to ensure the aesthetic was maintained - this is described in the tools section.

The group organisation was largely already applicable to our team, again sure to the use of agile SCRUM. However, we did notice their lack of mention of risk managers and a web designer so these were added. A large chunk of their documentation was describing the rationale behind role assignments, which are obviously not applicable to our team. Consequently, this has been removed and replaced with a link to our previous assignment descriptions, in order to avoid repeating ourselves. We feel that our previous role workings compliment the documentation of *DICY Cat* and so minimal documentation changes were needed other than this.

For the project plan of assessment 4, we have decided to resort back to the method of project plan which we have been using previously, and the documentation has been updated to explain this. This was due to *DICY Cat* only providing a flattened gantt chart made using PlantUML, which is consequently difficult to edit. It was decided that it was a better use of our time to simply expand on our own gantt chart in terms of assessment 4.

Other Documentation Changes

Requirements

Link

DicyCat’s requirements were generally good, however it was decided by our group that they were missing some important information. We added requirements that were relevant for phase 3 of the project, such as requirements relevant to ET patrols. We also added requirements that were initially missing such as a core requirement that the game should be controlled with arrow keys. Generally the requirements DicyCat had were good but our team felt there were some fundamental requirements that were missing and they also were missing requirements relevant for phase 3.

Risk

Link

It was decided by the risk managers that the risk assessments provided by *DICY Cat* were thorough and cohesive enough that no changes were needed.