# Software engineering report

## Successes

We managed to achieve a good majority of the functionality of the game. The following list will describe the aspects of the game as expected. For more information, see the system tests under the relevant sprints.

* A colourful, functioning board that resembles the board game - however this was not merged with the rest of the functionality of the game, so the final game must be played in the command line interface
* Player
* Player and weapon graphics for the early stages of the game
* Player movement
* Accusations
* Suggestions
* The ability to win the game
* Ability to lose the game

## Unfinished tasks

* The GUI for the fully functioning game
* The AI
* The AI difficulty settings
* The Menu screen
* Ability to play with less than 6 players

## What went well?

### Team roles & delegation

At the start of the coursework we established internal team roles. This decision allowed us to decompose the workload into more manageable sections for the team. This was extremely important as the overall spec for the coursework was very daunting. This was because none had much, if any experience programming games, let alone in a team. Therefore, breaking it down into near isolated sections allowed us to delegate the workload to suit the strengths of the different members of our team. It also allowed us to leverage our solo programming experience, therefore, putting less stress on peer programming, something that we didn’t have much experience with. These team roles weren’t set in stone, however, gave us some much-needed structure to start the coursework effectively. As the programming got more difficult, we started to implement more peer programming to keep up team morale and ensure that we were making continued progress.

Another benefit of this was that it allowed for the members to work on their specific tasks in their own time. As the tasks were independent, it allowed us to theoretically make steady progress. Unfortunately, despite this forethought and planning, we struggled with progress for the first few weeks of the coursework. This was due to different factors, one of which being that members of the team had individual.

Team dynamic

One of the biggest advantages of our group was our consistency with the work, despite having a rocky start, we all stayed consistent with the weekly meetings; having nearly 100% attendance from all members. This increased team morale as we felt that every member was always trying their best.

The benefits of a group coursework were the joint responsibility, being responsible for the grades of your team members as well as your own helped push the team forward as no one wanted to let down the team. Because of this, our overall team dynamic and understanding was better, we were all invested in the success of the entire team. This was tested at the start of the coursework where we were discussing the different options for the coursework, such as what language / software to use. This was one of the most important decisions of the coursework. Despite the team members having different preferences and ideas for this, we maturely decided on the use of python. This was because it fed into the strengths of the main programmers, therefore it was the best choice for the project, even if other members of the team weren’t as comfortable with python.

One of the benefits of our team was the fact that most of us had been good friends since the beginning of university, this meant that we already knew we had a good working dynamic before the coursework started. Although working with friends ran the risk of procrastinating more, we decided it was more manageable than the risk of a poor or clashing team dynamic which may have occurred if we had teamed up with people we didn’t know. Only a few members of the team were less familiar with the group, however, this did not pose any issue, and helped us stay on track as we were responsible for the grades of people we didn’t know as well.

### GitHub

The use of git was one thing that went well throughout the project because it allowed for certain members of the team to work on different areas at the same time however not causing any merging issues as part of the process and also allowed for us to make some changes to the project and try things without causing the project to break or have serious issues. This was something that went well because it seriously reduced the amount of time for bug fixing and fixing things that had been broken, because if there was an error and some code completely broke we could go back to the version saved on git in order to restore the last working version.

We also implemented github actions to automate code testing when pushing to the repository and branch protection for the main branch so if a broken commit has been pushed it cannot be added to the main branch.

Our repository: <https://github.com/Software-Engineering-Cluedo/Clue>

## What issues did we have?

### Independent assignments

One of the first problems we faced as a team was the fact that a lot of us had prior assignments to complete at the time, meaning we couldn’t focus on this coursework as heavily. This fact affected the team morale as we weren’t making as much progress as we had planned to, making it even harder to work. Fortunately, as soon as these assignments were handed in, the entire team was able to switch focus to this and catch up on the progress that we missed.

Another issue linked to independent assignments was the issue of scheduling, all members of our team are studying different modules this term and therefore have a very different schedule. This, with the addition of our large team size, it was difficult to organize times that worked for all members. Whether it be meetings, or peer programming, scheduling caused us a few issues with progress. Fortunately, we were all flexible with each other which allowed for us to finalize meeting times and other scheduling needs.

### Lockdown and restrictions

Several members of the group were severely affected by the lockdown which meant that they were struggling to cope with the workload that was needed alongside concentrating on other lectures. The lockdown and stress of the university course alongside the severe changes to life as we know it caused many of the group members to have bad days where they suffered and so there were sometimes delays being caused. The effect of the lockdown also meant it was more difficult to arrange meetings because despite the fact we were able to communicate through instant messengers, we were not able to have the in person group meetings and renting out of rooms in order to work through the project together. This meant that certain members of the group had never met others in person until the last couple of weeks before the due date.

### Varying strengths

Although our teams' varying strengths proved to be very helpful with the coursework, it did cause one issue. We found that all of our members were familiar in different coding languages. For example, we had 2 coders very confident in python and 2 coders who were reasonably strong in Java. The other 2 members were comfortable with both which caused a slight divide and a tough decision for us over what language we wanted to use. In the end, we decided to come to the decision to prioritize the 2 strongest coders, and use the language they wanted to use, which unfortunately led to the other 4 playing catch-up to get to the same level of coding as the other 2. This was especially hard for the weaker 2, as they had a lot more catching up to do as it was the first time they were doing python. We also found that once we had found out what language we wanted to use; we were unable to find a suitable GUI option for us to use. Again, this caused a slight problem between the group, however, due to the previous experiences of our 2 best coders, we decided to go with what they were most comfortable with.

Another issue that we have was that some of us had issues getting Gitkraken and Visual Studio Code to work. There were set commands we had to input, however, for some of us, this caused issues causing us to redownload and install python multiple times for it to work. However, this was resolved in a reasonably short amount of time, and so allowed us to continue working on the assignment with only a minor set-back.

## How well did the team work together on the task?

The team worked together very well. We had a high morale throughout the entire project, even when the team or individuals were struggling with either work or personal reasons. We were extremely supportive of each other and made sure to respectively raise our concerns if we had any. This was helped by the fact that we all wanted to achieve a high grade and therefore didn’t hesitate to put the work in.

We managed the team’s strengths and weaknesses well and ensured that everyone had tasks suited to their strengths. We also decided to split the team into 2 separate groups. As we had a large team; of 6, we thought it would result in higher levels of progress and mitigate any clashes. We were conscious of our team’s progress when we were working as 6, and quickly realized that it would be more effective for all of us if we split into two groups, tackling different problems. The main subjects were mechanics and graphics. Splitting up the group also ensured that everyone was involved as much as possible. The logic behind this was that it becomes much easier for people to become distracted from something when they are working by themselves because if they hit an issue that they do not know how to fix then they could be stuck for a long time however if there are 2 or 3 people working then it is likely that they will be able to help each other solve their issue. It also improved the group morale because it ensured that there was some conversation going on at the same time to prevent stress and increase productivity.

There were also many cases in which the benefit of the team worked out to seriously improve the productivity of the project, for example there are many cases in which many people in the team were able to resolve the issues of the other’s.

## What did we learn during the coursework?

There was a lot that was learnt through the coursework, interestingly different members of the group learnt different things based on what their strengths and assigned tasks were. For example, two of the members of the group who were particularly skilled in the management areas learnt a lot about how to code from the coding experts, and the coding experts learnt an awful lot about the process of a project from start to completion.

This was our first experience programming as a team to deliver a program. This by itself came with a lot of lessons, including the importance of commenting code and ensuring that your code is simple and understandable, for others to edit and add to it. We had all learnt this from previous modules, however experiencing it allowed us to realize the importance of it. Another important lesson was to ensure constant communication throughout all members in the team, regardless of their strengths or assigned tasks. Making sure that everyone knew the progress of the project was important for team morale and allowed us to help each other more effectively. The team roles described before were useful but did not stop us from helping each other in any way we could.

One thing that was less so learnt, more so improved, was the teamwork ability. From the start of the project there were a few members that weren’t as familiar with the rest of the group as the others were. This gave the chance for those members to get to know the others as well as learning how to work with essentially strangers. This is a valuable lesson for the workplace.

## How would we improve next time?

Next time, we would put more emphasis on the planning stages of the project, although

Another idea that we could have changed if we did this task again, would be potentially doing the assignment in a different language. This was because the bulk of our time was us learning a different language, almost from scratch. This was a huge setback for us as not everyone was able to help code for the task. Had we had the choice to do it again, we would’ve chosen to use Java, as more people were comfortable with this, meaning it would’ve taken less time for us to get up to the speed we were needing to complete this assignment.

## Breakdown of workload

This shows the breakdown of the tasks and which team member completed the different tasks

|  |  |
| --- | --- |
| Team member | Tasks completed |
| Ethan | * Most if not all programming regarding game logic and functionality * UML diagrams (with Sarah) * Class breakdown and diagrams (with Sarah * Order of tasks (with Sarah) * Code commenting |
| Adam | * Use case diagram (with Sam) * Gantt chart (with Sam) * Sequence diagram * Minuted meeting (with Sam) * Contributed to programming * Contributed to Javadoc testing * Group report (with Umar and Will) * Contributed to requirements |
| Umar | * All 6 sprint cycle documentations * System tests * User requirements * User story * Risk analysis (with Sam) * Graphics for board, weapon, player * Group report (With Will and Adam) * Contributed to JavaDoc testing |
| Sam | * Use case diagram (with Adam) * Gantt chart (with Adam) * Sequence diagram (with Adam) * All minutes meetings * Contributed to JavaDoc testing * Contributed to requirements * Risk analysis (With Umar) * Contributed to group report |
| Sarah | * Programmed entire GUI * Contributed to programming * UML diagram (with Ethan) * Class diagram (with Ethan) * Order of tasks (with Ethan) * Code commenting |
| Will | * Group report (with Umar and Adam) * Contributed to Javadoc tests * Contributed to GUI programming * Researching python and cluedo game structure |

## Is there anything the team should be aware of when grading the submission?

The team is requested to take into consideration the issues mentioned above that were unrelated to the project and caused delays of long term that were totally out of our control. It is worth noting that despite some people in the group not fully knowing each other, there were other members of the group who do know each other well, this means that there is an aspect of when one of those members suffer severely from mental health issues, the other members of the group are affected as well because it is never easy to see a friend or a colleague struggling. This fact delayed progress to the project, especially near the end as the stress was higher nearer the deadline. Unfortunately this affected the entire team as the

We ask that you consider the extenuating circumstances of our project. Not only did the effects of lockdown affect our team's ability to make progress as we couldn't physically meet up, but certain members of the team were and are struggling from severe mental health issues which affected not only the progress of the game but also the team morale as we all trying to console said members and help in any way we could. This got worse near the end of the project due to the increased stress of the upcoming deadline. This led to the team member(s) taking a step back from working in order to help their mental health. All team members understood and supported this decision however it led to unforeseen hurdles regarding the work, resulting in us not being able to reach the same quality of project as we had intended and planned to.

Regarding the submission, we have three different zip files, one for the main branch, one for the currently worked on, and one for the gui which was not merged due to aforementioned extenuating circumstances

We have attempted to make a [documentation website](https://soft-eng-clue.readthedocs.io/en/latest/index.html) like how javadocs worked, however the creation of the documentation site through the config was fiddly and doesn’t show all the details in the code we commented on.

## Delegation of marks:

Splitting up 120marks between 6

adam - 21

sam - 21

umar - 21

ethan - 24

sarah - 18

will - 15