**Development Process**

At first, we held several brainstorming sessions to decide the game and the twist we were going to implement. We found it useful since everyone can express their thoughts, so a bunch of great ideas were generated from these sessions. Also, we could handle conflicts immediately and negotiate to get a suitable solution. Successfully, we reached an agreement on the goals and contents of the game.

We chose the agile methodology to manage our project since it was hard to get a specification at the beginning. Scrum is a widely used methodology in agile development. Due to the lack of time to follow a daily schedule, we decided to divide the development process into some short periods. In each development stage, we got an executable game with some functions expected to implement in that version. For instance, in the first version, we’ve only got a game with one level and one player without weapons and enemies. Then, other elements like river, diamond and other maps were added to our game.

Another tool we used in agile development was Kanban. We had an offline meeting every time we stepped into another stage of development. Most of them were held in the form of brainstorming. Every member introduced the work they’d done then and discussed the features to be implemented. After a meeting, we recorded tasks categorized by in-progress, to-do and done in the WhatsApp group.

We valued communication. Apart from offline meetings, we communicated through WhatsApp, from which we can message a certain person or the whole team instantly. When some glitches were found or the game was updated, our members can give a response immediately. Using this online instant messaging app has greatly improved communication efficiency.

The tool that we used most was GitHub. And it turned out that it made our collaboration much more efficient. After some discussion, we clarified the duty of every member. There are five branches for every member personally, a develop branch for merging and a main branch for the final version of the game. Every time we tried to modify or add some features, we would pull the develop branch, which contained the newest version. When it comes to pushing, we merged our branch into develop branch and updated the two branches together.

**Team Roles**

We assigned a specific role to every member according to their preference and strength. Although members took charge of different parts, we shared thoughts regularly and considered the availability and comprehensibility of APIs in separate sections.

Yuewen was in charge of the requirement section and the implementation of maps at every difficulty level. Specifically, she designed three new maps and then added elements like lava, diamonds, and buttons.

Zongjian took the responsibility of designing and coding the whole system structure and finished the design section. He drew the UML class diagram and specified the function of each class, which made our development process clearer and easier. Also, the significant part – the twist of adding enemies and weapons was implemented by him.

Hings was responsible for the implementation section and the implementation of multithreading. His work has resulted in smoother animations and controls throughout the game. Also, he helped fix some glitches in bullet reusing, which increased the robustness of the system.

Francesca was responsible for the evaluation section and the implementation of the level Manager. She also designed our game's opening animation and victory scene, contributing to improving the user experience.

Siru was responsible for the process section and the implementation of multiple players, which is based on Zongjian’s work. She helped to implement the control of the second player.

**Reflection**

Overall, we worked together well. Agile development allows us more flexibility especially when we don’t have a specific plan. We can also continuously improve and change features when we get feedback from evaluators or when we come up with better ideas. Also, the assignment of roles and use of GitHub encourage the increase in productivity since every member has clear accountability. We can easily identify issues and complete every required task.

However, there are still some areas that we could improve in the development process. One of them is equal contribution. Because of the imbalanced workload in each area, some members make more efforts on specific features. Although it’s hard to distribute work in a completely fair way, we should adjust everyone’s workload in the process of development.

Another critical challenge is efficient communication. For obvious reasons, it’s easy to record to-do lists and use Kanban in WhatsApp groups. But if we chose to use some online tools like Jira, the recording of tasks would be much more professional. Also, using online meeting software like Zoom instead of text message can help to save time.