Part I

Reflective Comments

0.1 Technical Issues

0.1.1 Version Control System

At the beginning, GitHub was utilized as the version control system. Later on, unity collaboration Cloud is being used, for several reasons:

1. Difference shown in the project

   The project consists of not only codes but lots of sprite and prefab assets. Difference of sprite and prefab between last version and latest version could not be shown on GitHub, because images(sprites) could not be shown by GitHub. However, unity Cloud is embedded in Unity, the difference could be shown directly by opening that sprite in Unity Project.

2. Simple push and pull mechanism

   Working concurrently with GitHub means that lots of branches should be created. Unity Cloud is simpler, the number of people working on the project, what they are working on, if you should update or upload, they all shown in the Unity Cloud, which lowers down the possibility of version confliction.

3. Easy integration

   Integrating the project using GitHub is complicated, for instance, if subsystem A and B are required to be integrated together, B should be downloaded first, and then it should be imported into the project. With Unity Cloud, above two steps are developed as one step. One single button could integrate the work.

However, it still have a problem. To upload the work, Unity Cloud requires proxy server or eduroam, but sometimes proxy server and eduroam might not work, so the version could not be updated.

0.1.2 each part of the game system is hard to implement

0.1.3 not interface and abstraction first

0.2 Project management Issues

0.2.1 Work Allocation Improvement

When the project starts, the division of collaboration is simply divided into coding, testing, user interface design and prototyping. After a long discussion within the team members and the supervisor, the project is finally divided into three divisions: game system design and testing, user interface design, game play context design.

The division of the project has three main modification.

Firstly, testing and system design are combined together, since the people who design the system know which parts should be tested (The testing above just consists of unit testing, integration testing and system testing.The acceptance testing will be delivered by customers.)

Secondly, the work of prototyping is deleted, since the initial prototype satisfies all requirements. Next phase the team members will implement the properties of the prototype. The prototype would be modified if the requirement is changed or a more acceptable prototype is given, based on our weekly meeting discussion.

Thirdly, game play context design is added into the project division. When the project was in the design phase, the importance of context throughout the game is neglected. The users might not be interested in playing the game if they are not attracted by the game scenario. Moreover, without context the educational property of this project will be less likely to implement. Dialogue with non-character player, context shown in the scrolls are included in the game play context design.

The disadvantage of this kind of collaboration is that not everyone could write the unity script, and not everyone could design the user interface (although the user interface design could be discussed by the team, the details are decided by the user interface leader). Therefore, not everyone could learn all skills involved in the project. However, since the team did not do much in

the winter vacation, working concurrently would be an efficient way to finish our project with expected quality.

0.2.2 Collaboration Improvement

specific plan,