**Overview**

**Extreme Programming and Pair Programming**

In our project’s first sprint we used extreme programming via pair programming, each taking it in turns to code and while one person is coding the other person would look at what they have done to see if they could improve it or if the person got stuck on a part of the code then the other two helped with the solution.

Example commit where Pair Programming was used: https://git.cs.kent.ac.uk/ai261/Ascension\_project/commit/1377d183

**Development Methodology**

In our project we used scrum to direct how we would plan and produce our project, as evidenced by the product backlog and the scrum backlog which includes the sprint burn down chart. The first scrum artefact created was the product backlog, including everything that we wanted to include in the game. Then from this we created the sprint backlog which is based on how many points from the product backlog we felt that we could do in the 3 week sprint time limit that we had. The time taken to complete each of these items were estimated using planning poker and story point values.  
As we did not have a previous sprint to calculate team velocity from, we made a conservative estimate as a team.

**Testing**

During the first sprint, any new functionality added (whether solo or in a pair) was tested by the person(s) not present. This was usually done during our group meetings so feedback and ideas could be bounced around between us. This allowed both effective finding and squashing of unintended behaviour, as well as increasing the speed of iterating and improving each module.

**Acceptance criteria**

For the acceptance criteria we would look at this while we were creating the objects, then what we would do is that if it works then we would move onto the next task, however if it was not satisfactory then we would fix the issue. An example of this would be the sword; to start with we had a working but stationary sword, to make it look polished we added in a swing animation.

**Iterations**

We are going to use iterations in the Scrum for example Sprint 1, 2, and 3 to create and improve on parts of our project.

However in our Sprint 1 we used iterations to create basic classes such as the sword, gun and enemy classes. First the basic class was created, for example the enemy class started unable to move. Then it was improved to make it follow the character, then it was made to attack and take health off of the character and finally it was made so that when the enemy was killed it would split into 4 pieces to represent it being chopped by the sword.

You can also refer to our git hub as an example of iterations as each commit is a different version of the project and then each time it is committed, the project will have been improved in some way to make it better.

**Creating and assigning tasks**

We formed tasks by meeting as a group and discussing what each of us wanted out of the game and the best points would be written down on the product backlog. When we felt that we had enough we refined the points so that they all suited the game that we were making.

Tasks from the sprint backlog were assigned to team members by going through each task and discussing who wanted it, normally one of us put ourselves forward. However if we all felt that the task was particularly hard (or had ramifications for other modules/future development), then we would assign the task for all of us to do using the pair programming technique.

The way that the team assigned the tasks was we put the necessary tasks into the sprint backlog then we went down each task one at a time and discussing who wanted the task, normally one of us put ourselves forward. However if we all felt that the task was particularly hard (or had ramifications for other modules/future development), then we would assign the task for all of us to do using the pair programming technique.

**Evidence of using Git**

We used git to manage the different versions of our product. The Git URL is: https://git.cs.kent.ac.uk/ai261/Ascension\_project.git