Module	SEPR
Year	2019/20
Assessment	4
Team	Krojan Horse Expanding on NP Studios’ Assessment 3
Members	Abel Kent, Toprak Onat Sen, Vinu Jey, Tom Emmerson, Diana Udrescu, Toby Popov, Jamie Todd
Deliverable	Project Review Report

Team management outline

Structure

Our team structure changed over the course of the project to better manage the developing needs of the project.

One aspect that remained from our early team structure was the idea that as each stage of the project involved some iteration of something included in the previous, as group members gained familiarity with their sections of the project, they would continue with that in the next, ensuring maximum efficiency.

This is briefly outlined below:
Implementation: Tom, Jamie, Toby, Vinu, Toprak
Reports and documentation: Abel, Diana
Change reports: Abel, Diana
Architecture: Toby, Vinu

However, as opposed to the start of our project we dissolved the practice of defined roles for each of our team members - wherein each one is strictly responsible for one aspect of the project. This became unnecessary as varied time constraints and lack of contact due to Covid-19 made it difficult for some roles to perform their detailed duties, as well as each member becoming more well rounded in terms of what they could bring to the table, making the roles largely redundant. This was true both in terms of the team management roles we defined, as well as the work each member undertook as most members worked on a variety of different sections of the project including the documents and the code. This allowed more thorough read-through checking and allowed all members to keep developing some part of the project.

Management

As the project has developed, our communication methods evolved to become more focussed on remote work. This was due to a number of reasons, namely that members of our group had other responsibilities that affected our ability to all meet in person, and as we became more accustomed to the project and what each individual could bring to the table without needing to discuss it in person. This became particularly vital when the Covid-19 pandemic resulted in many of our number leaving York or being unable to leave their homes. We began having Discord meetings at various intervals to check up on progress in replacement of face-to-face meetings where we could all share screens with each other allowing group work to still take place effectively.

Tasks continued to be delegated and tracked through Monday.com right up until the end of the project for a number of reasons. We had grown familiar with the systems over the course of the project and the features it provided proved to be useful throughout, it’s presence as an industry standard also worked to lend our project some credibility and familiarise ourselves with it and similar software for later in our careers.

Software development outline

Methods

Over the course of the project, our focus on an Agile based software development style remained key, due to our small team size and rapid sequential deadlines. As explained in the first section of the project, Agile was the best management style to meet these ends.

However, we drifted away from a Scrum based style to a more generalised Agile one. Scrum is more suitable for projects where there are rapidly changing requirements [1], whereas in our work, the requirements were clearly defined at the beginning of each stage, and were then fairly static with only a handful of additions at the fourth stage, and so we were unable to benefit from this aspect of Scrum. In addition to this, once we were unable to personally interact with each other due to isolation during the COVID-19 outbreak and our schedules overlapped less, we found it more conducive to our work for team members to focus on their work as individuals. Team discussion still took place, but at wider intervals and at less formal times over discord [2]. We did maintain a relatively flat team structure, similar to that of Scrum, rather than having a project manager as there would be in a traditional Agile team. This was possible due to each person having their own tasks and deadlines set on Monday.com.

Tools

Tool	Purpose	Changes of use throughout project
IntelliJ IDEA + VSCode	IDE's for Java development	Used consistently throughout the project as they provide a powerful set of tools for Java development. As the same language was used throughout we saw no need to change IDE's, particularly as we became more used to their features.
Piskel + Pixel Studio	Asset creation (pixel art)	Used consistently through the project. Allows easy creation of pixel art assets, both with a large collection of various tools and brushes. Fits in with our teams and other teams' chosen art styles.
PyxelEdit	Asset creation (tile maps)	Used through the first 3 phases of the project, no longer used in the fourth due to having all of the required tilemap assets and levels which this software is intended for.
Gitlab	Git-repository and DevOps lifecycle tool	Used consistently throughout the project. Allows for individual work development in a collaborative environment, includes built in conflict management and other useful tools for various versions of the project to be created and fallbacks being possible.
Sourcetree + GitHub Desktop	GitUI	Sourcetree provided an easy to use UI for accessing Git-repositories, with a couple members using GitHub Desktop as an alternative interface to GitLab. We did not need to change our interfacing software to repositories since we used Git for all stages rather than Mercurial.

Citations:

1. ReQtest, “A guide to Agile and Scrum”, 7th August 2018 Available: https://reqtest.com/agile-blog/agile-scrum-guide/ [Accessed 10th March 2020]

2. Codete, Paweł Dyrek, “Software Development Process in the Coronavirus Reality”, 2nd April 2020 Available: https://codete.com/blog/software-development-process-in-coronavirus-reality/ [Accessed 15th April 2020]