DRTN Tool list over all of SEPR

Tool	How it was used	Why it was used	How did it evolve
Github	Provides version control and file sharing via the use of repositories. This allows collaborative work on source code as well as the ability to revert code if, for instance, there is a change in requirements.	It is extremely user friendly due to the straightforward GUI and substantial documentation. It also provides a platform for integration of other project management tools.	Used for the entire project. This was due our consensus that there wasn't another tool available that provided any additional functionality..
Google Drive	Provides collaborative storage and editing of the documentation for the project.	It was a tool that all the team were familiar with and provided all the functionality that we required. Was also useful as we have unlimited storage space, as provided by the university.	Used for the entire project for managing our documentation as we didn't require a tool that provided any further features.
Zenhub	Allows us to manage our sprints via the use of 'burndown-charts' that show outstanding tasks along with which member of the team is assigned to it.	It is integratable into GitHub which allows us to assign issues within the Github repository as tasks in the burnout chart	Used for assessments 1-3, however during assessment 4 we discovered **Projects**, a board based system similar to Zenhub and Trello but native to Github. This alternative was much more transparent and useful as it keeps more information easily accessible in the repository.

Slack	Let the team communicate via an online private chat room. Has the functionality to provide different channels for different topics.	The feature of allowing different chat channels is what made us choose this tool over a more mainstream chat tool such as Facebook messenger.	Used for the whole project as no greater alternative was found and its features never became redundant
Travis CI	Automatically tests the builds of any commits we make. This prevents the need for us to manually download, compile and test our code after every commit.	Like Zenhub, the tool is supported by Github meaning that integration isn't an issue. This was useful when changing our codebase in assessments 3 and 4 as the tool didn't need to be configured especially for a certain project	This tool has continued to be compatible with the code that we have developed so hasn't become redundant throughout the software development process
IntelliJ	A java development environment that was used by the whole team to write the source code for the project	The environment provides many unique features such as advanced debugging and refactoring mechanisms.	Despite the changing requirements, we always used Java to develop our project. Therefore we didn't feel we needed to change the IDE as it was always the most extensive environment available to us throughout the development process
Gradle	A build automation system that supports incremental builds by determining which parts of the build are up to date and preventing parts which aren't	We concluded that this would be the most useful tool to manage our code development/testing due to its ease of use and compatibility with our IDE, IntelliJ.	As both projects that we took control of in assessments 3 and 4 both used Gradle, it meant that we could use the tool for the entire development process

	up to date from being re-executed		
Smartsheet	A tool that gives the facility to construct Gantt charts.	This tool provided all the capability we needed to construct and update the Gantt chart for our project.	The tool has been used throughout the whole process due to the requirements of the Gantt chart not evolving
Pixelmator	An image editor for macOS, it was used to draw some of the graphics for our game.	This tool provided all the functionality that we needed in order create our graphics and export them in a specific format	This tool was used throughout every assessment which meant that we could keep a consistent design for our graphics.
Paint.NET	An image editor for Windows that was used to draw the background map of our project	This was a tool that was already familiar to the whole team and provided all the features necessary to create the graphics we needed	Like Pixelmator, this tool was used throughout the whole process as it provided all the features we needed.
LucidChart	A web-based chart creator that was used to create the UML diagram for an abstract representation of our system architecture	It's simple design and compatibility with Google Drive meant this tool met the requirements needed to design our abstract architecture	This tool was replaced by StarUML in assessment 2 as it was too abstract to provide a concrete representation of the system's architecture
StarUML	A UML chart creator that can be used to create very detailed and concrete UML diagrams	This tool was used as it allowed us to create more concrete UML diagrams by allowing us to include details such as the stereotype of a class and the private/public nature of an attribute	From assessment 3 onwards, we decided to use the UML exportation tool in the IntelliJ IDE as it generates the UML charts automatically therefore saving time.