Below is a list of all of the pieces of software that we used to create our game, I will be explaining what the software is, a short description describing why we chose them and what they were used for. A lot of the software was chosen simply because of our past experience with using the software, as we wanted to achieve our goal of creating the final product as soon as possible.

**Unity**

Unity is the game engine that we decided to use for creating our game. Unity is GUI Driven game engine that allows you to develop games with relative ease, the GUI itself has a huge variety of features that can be easily added to any object within the scene, meaning a complex game idea can be created relatively quickly. Unity contains a huge amount of features built in, such as lighting, asset loading, a visual representation of your game, physics, liquid simulation, all of which can be massively modified using the user interface. When using these features Unity does a large amount of the coding for you, meaning if you choose to use the built in features you can make complex projects quickly and get a good base idea down within a few hours of work. We are going to be using unity to do all of our game, this includes the various maps that we implement, the menu system and the gameplay itself. We chose Unity over other products such as Unreal Engine or Game Maker because all of our group members have used it in the past for various projects and we wanted to get a base game working as soon as possible so we could easily see the progress of our project. Another reason for using Unity is the way prefabs work, this allows any of our group members to create work at their own leisure and create a prefab. This prefab can then be imported into our main game project and it will have all of the features built in, this feature is invaluable when working in a group environment.

**MonoDevelop**

MonoDevelop is the IDE we decided to use when creating all of our code for our game. MonoDevelop is an Open Source IDE which supports multiple languages and operating systems, itis the recommended IDE that you can use alongside Unity to create script files in C#. These script files are then imported into the Unity project and can be applied to any of the objects in the scene, one of the benefits of doing this is the fact that variables declared as Public can be seen from inside Unity and can be edited using the GUI of Unity. We main reason we used MonoDevelop because it is installed alongside unity and requires next to nothing to setup. Another reason we are using this is because the group members using unity to create our game are also confident in using MonoDevelop as they have previous experience with using it with unity. Also MonoDevelop has auto completion available for all of the features within Unity. One of the main alternatives was Visual Studio, but this takes a little more preparation to setup when working with unity and can cause some problems when saving files. Another alternative would be creating the scripts yourself in a text editor such as Notepad++ or Sublime Text and manually importing them, but this has 0 debugging potential and error handling when typing the code, whereas Monodevelop has code prediction and spell checking.

**3DS Max**

3DS Max is a piece of 3D modelling software that has a massive amount of features for both basic and advanced 3D modelling, the GUI gives the user multiple viewports of the model they are working on and allows them to easily see all sides of the model. 3DS Max also supports Animation that is relatively simple to do for the models. We chose 3DS Max because most of our group members have used it before and the members that have felt confident that they could create the models that we wanted with the tools that are available. Another reason we chose this was because it’s very simple to save the models we create and import them straight into unity, we can then start using them as soon as possible with no compatibility issues. We chose 3DS Max over other software such as Blender or Maya simply because we are much more comfortable using this software as it can be relatively easy to create the models once you have an idea of what you want to create. 3DS Max can also be used to create and add textures for the models that support lighting and reflection.

**Photoshop**

Photoshop is the art package that we decided to use for creating all of the Images and Textures that we needed for our game. Photoshop is a very powerful piece of photo editing software, with a large amount of built in features and effects which can manipulate and edit images. One of the most important features in Photoshop is the fact that images can have multiple layers, meaning you can have two images in separate layers, whichever layer is on top will be drawn above the other image. This is very useful when working with editing and creating your own images. Another reason we chose to use Photoshop is because all of the group have used it multiple times in the past, and it has a lot of useful tools for creating the textures we want, such as customizable gradients, as well as hundreds of Pen and layer options. Two alternatives that we considered using were Paint.net and Microsoft paint, Paint.net is very similar to Photoshop, it has less features but it is free, which is useful for our group members that don’t own a version of Photoshop, as the university recently installed Photoshop CS6 we created all of our textures and assets that we needed using that.

**GitHub**

GitHub is an online directory hosting service that allows you to easily share your repository with other people, who can clone the repository and edit their own version, this can then be uploaded into a new branch if needed. All of our group members have been added to our GitHub and our Unity project has been posted up there, meaning each member can download the project and create their own assets, prefabs or anything else they wish to add or change. GitHub is primarily used for source control and versioning, This means that old versions of the repositories can be accessed, meaning that any changes in a new commit that may cause something in the game to not function correctly can essentially be “undone” using and downloading a previous version from GitHub. This also means that if any of our work is lost we can still recover a previous version using GitHub.

**Filter Forge**

Filter Forge is a ‘plugin’ for Photoshop that allows the user to create very unique and customizable textures using sliders and properties. This software uses code with specific parameters to create textures which can then be used by the user. This is a standalone piece of software with a monthly fee, but can be used with a trial basis. We used filter forge to create some of our textures that we didn’t create using Photoshop. We used this over Photoshop because it gave us great control over what we created, and we found some great pre-sets online that allow you to make similar textures to other users, as the settings for the filters can be shared and loaded into the software, allowing you to simply save your own version. Filter Forge allows you to save the textures in multiple file formats at multiple resolutions ensuring that it can be useful to as many users as possible.

**Facebook chat**

Facebook has been the main platform we have been using to communicate between our group members, we have organised most if not all of our group meetings using it and have been able to send images and links to what we have been working on during the week when we aren’t at group meetings, we also used it to discuss any problems that we found when working, allowing us to help each other with the work. When we are completing work we aim to get as much done of the Unity project within the Lab session that we set, meaning that we can work on the Game together to ensure we are creating what we want, and we can all help if there are any problems. We then discuss what document work we will be completing in our own time. This can then be posted in the Facebook chat via pasting or google docs for briefing and consideration from the group.

**Google Docs**

Google docs has been used throughout our project in combination with Facebook to upload and share our work between the group when we haven’t been at meetings. Google Docs is extremely useful as it allows both word documents and spreadsheets to be uploaded and viewed using a web browser. This means that any quick notes we make during group meetings will be placed into google docs so that all the group members can see them, and also draft pieces of work can be placed into a document, we can then notify the group about the work and ask them to get read through allowing us to get some feedback on the writing from multiple people.

Talk about methodology, scrum waterfall etc

For our development we used a waterfall development cycle when working on our game. This means that we would implement one feature and ensure that it is working correctly before moving on and implementing another feature. This meant that after we have implemented numerous features we would have a base working game with all of the key features that we need. When working on the document we assigned each section to a member and allowed each person to work on that section in their free time. This work is then reviewed by all of the group members; this was done via the group meetings, Facebook chat and google docs.