**Tools & Technologies**

Group 14 – Cory Atkinson (s3775626)

**Software:**

The software required for this project would include 3d modelling software for asset creation, a game engine to build and publish the project, audio production software to produce the in-game audio and soundtracks, and an Integrated Development Environment (IDE) to develop the code to manage the game itself and the entities within it. Other programs of use could include illustration software to create textures and texture maps, and 2D artwork for the project. In the scenario that our group is to create our project with our desired programs and licenses, the software we would use is as follows:

**3D** **Artwork** – [Autodesk 3ds Max](https://www.autodesk.com.au/products/3ds-max/overview)

**2D** **Artwork** – [Adobe Substance](https://www.allegorithmic.com/substance) / [Substance Painter](https://www.allegorithmic.com/products/substance-painter)

**Game** **Engine** – [Unreal Engine 4](https://www.unrealengine.com/en-US/what-is-unreal-engine-4)

**Audio** **Production** – [Avid Pro Tools](https://www.avid.com/pro-tools)

**Programming** – [Microsoft Visual Studio](https://visualstudio.microsoft.com/vs/)

These programs would be essential to complete this project. However, most of the tools here are not free to use.

Autodesk’s 3ds Max is a subscription-based software with a hefty monthly/annual fee but given the extensive documentation and the flexibility of the program itself - once the user becomes more knowledgeable - it’s clear this program would benefit greatly in the creation and animation of 3D artwork.

Adobe Substance / Substance Painter is another piece of software that is subscription-based but it’s carved itself into the game development industry as one of the easiest programs to design textures from scratch and can be implemented into any workflow. It’s a must for 2D artwork and asset creation.

Avid’s Pro Tools will be the powerhouse behind the production of audio assets for this project. Any in game sound effects, voice acting, and music will be composed/created within Pro Tools and because it is an industry standard software, the plugins and documentation available will be liberal to say the least.

As for building and compiling the game, Unreal Engine 4 fits the bill. We had initially considered Unity for it’s ease of use, and gentle learning curve for those who have not used engines before. However, we found that plugins are expensive if you do not have the time or experience to develop them yourself and the time it takes to polish a game in Unity is vastly greater. UE4 out of the box offers a greater polish using default settings and with only minor tweaks through blueprints or C++ code, obtaining visual fidelity close to that of a AAA game is relatively straight-forward.

Lastly, Microsoft’s Visual Studio is the go-to for the bulk of code development; it’s extensive range in plugins and seamless integration with Unreal Engine 4 makes it a no brainer with the remaining of back end development completed using the blueprints feature available in UE4.

Of course, this goes without saying that all software used will be the latest version released.

Updates will be necessary during the process, though not a deal breaker, as the licenses for the paid software listed can be updated for a small fee if not for free.

**Hardware:**

An important thing to note is that a project like this will need more powerful computers for the creation of high polygon count 3D assets and the actual build process of the game. Baking scene lighting, rendering large quantities of objects on screen and compiling the finished project or experimental builds can be strenuous and require the appropriate hardware.

For any voice acting and in-house sound effect creation, recording equipment such as microphone(s), audio interface(s), headphones, and any stands will also be needed.

Another area that will also need specific hardware will be 2D production. Concept art is an important factor of developing a video game. Without it, artists are left to blindly create assets without guidance. To help design concept art creative tablets, such as Wacom, can be plugged into the artist’s computer so they may draw digitally; aiding the process of producing object/character vectors and transferring those to a 3D artist so that models can be created from them. Keeping all content digital also helps streamline everything during the project.

Lastly animation will need to be handled through dedicated hardware. Given the scope and potential production size, motion capture tracking hardware and mocap suits will be necessary to maximise the quality of animations for the project.

**Experience:**

Between the members of group 14 we hold the necessary experience to use most of the software listed. However, we will need some time to prepare and become competent with two of the programs. None of us have used Pro Tools beyond a novice level – Nathan being the exception. His experience with audio production software is extensive as his background is primarily in music- as such he’s been using Pro Tools for 5 years. On the same note, Lee is currently employed at a studio and works alongside sound engineers who can guide her to become a much more advanced user.

Substance Painter was only recently introduced to our arsenal and though it’s perfect for the project we lack significant experience with the software. That said, both Nathan and Nick are adept at Photoshop and Illustrator and the majority of the skills are transferrable to Substance.

As for 3D assets, once more, Nick is all over 3ds Max with his previous experience in the software being quite notable with work dating back to 2012. His grasp of 3ds Max’s complex UI and inbuilt tools gives him an advantage when producing high quality models and animation.

When it comes to Unreal Engine 4, Michael and Harry are the go-to members for level design and bringing all assets together in-engine. Although we will all need to learn how to use UE4 for testing assets and code we create and compiling test builds for showcase, Michael and Harry are currently the most experienced with Unreal Engine making them best suited to handling the bulk of in-engine development and asset compilation.

With all group members holding their individual forms of programming experience, using Visual Studio would be a breeze. However, Cory and Michael are the most familiar extensive experience; Cory primarily focusing on the Visual Studio IDE and Michael on the lightweight Visual Studio Code. Both are well versed in Unreal Engines natively supported language C++ making code development for the engine and the project to be built within it an easier process.