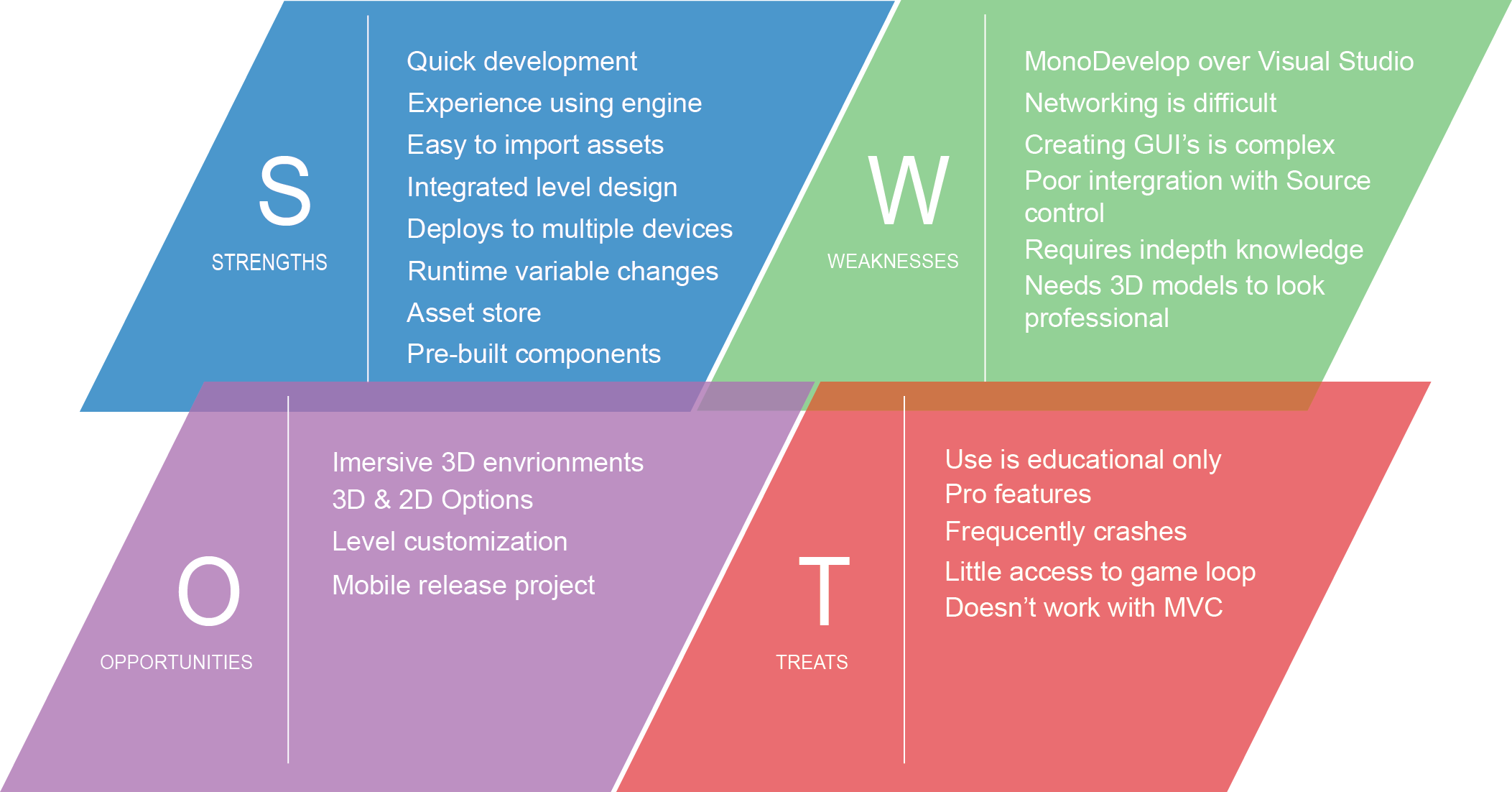
# Approach

As the project and its initial ideas began to unfold it was clearly a sensible plan with enough scope for expansion if necessary. As stated in the background and objectives there were little changes to the original project. These changes were made to make the game more challenging to design, develop and test. Although the changes were not set in stone it was important to evaluate them as this would ensure a better overall project.

## Unity

One of the first hurdles that needed jumping before the development started was to assess the different technologies available and how each one was going to benefit the development stages. To give a better understanding of the options, a research paper was completed in which several game engines where assessed to demonstrate their strengths, weaknesses, opportunities and treats (SWOT). See appendices. Not only did the research paper bring insight into which game engine was going to help develop a top quality project it enlightened the areas of the project that could have proved to be more difficult that first anticipated. The end result of the paper highlighted Unity as the chosen game engine. Some of the deciding factors were the experience the developer had with the environment, the rapid development made possible with the built in level design and the access to its own asset store. These are only some of the factors that played a part in the decision but deemed the most important. Knowing that the project needed to be developed swiftly, there was a massive need for external assets, mainly 3D models.



Using Unity allowed the developer to play to their strengths as a programmer. He was able to use C#, which is the programming language he is most comfortable coding in. There was less emphasis and pressure on level design as Unity has its own built in level editor. Unity can be debugged extremely easily, giving the option to edit variables during runtime. Unity makes it effortless to deploy to multiple platforms, this means that if the direction of the project changed and moved towards another platform, it would require little change if any at all.

## Visual Studio & Resharper

One of the main concerns with using Unity was the integrated development environment (IDE) Mono Develop. Although this may not strike many people as a real issue, it was believed that programming on a poor IDE could lead to sloppy coding. Luckily after only a small amount of research it became clear that Unity could be integrated with Microsoft Visual Studio. The Visual Studio plugin offers full syntactic support along with simple debugging options, everything needed to get the ball rolling. Another bonus of using Visual Studio was its additional plugin Resharper. Resharper is like a second pair of eyes for your code, it will ensure you code in a uniform format, keeping the variables consistent throughout the project and offering alternatives to certain coding approaches. For example using a LINQ statement instead of iterating through a list of items in a for loop.

These are very small enhancements for such a large project but the payoff can be immense when passing on a code base to another programmer or simply ensuring the code is clean.

## Game Design Document

After deciding on a game engine to use a game design document (GDD) was created. See Appendices. The aim of a GDD is to be a more in-depth, descriptive document, outlining the story, characters, level, gameplay, art, sound, music, user interface, game controls. All of these sections are vital when developing a successful game and each area was carefully assessed before starting the development of the project. Traditionally as the development of a game grows so does the size, structure and information stored in the GDD, however creating a skeleton for this document was important before the development started. It added direction towards all of the important areas required. The GDD is a very visual document expressing ideas with a variety of different media, whether that be links to videos or music, pictures from existing games, concept art scanned in from a scrap book, they all play a key role in the document. With this mixture of resources it was easier to visualise the intended game.

When first deciding to create a GDD it was unsure if it should be completed in the form of a blog. This would allow the developer to reach out to an audience, granting them feedback on development and providing external advice to a very remote project. The decision went against this method as a thriving GDD seemed to be a better approach and resource.

After the creation of the GDD several different areas needed to be explored before starting any of the programming.

## Illustrator

First of which was which resource was going to be used to create any 2D and or 3D assets. At this point it was obvious both were needed but knowing whether or not there was a need to create them or find external resources was important.

2D illustrations were need to act as the user interface (UI), without custom made UI the game ran the risk of not looking very professional. There was less emphasis on finding a piece of software to create the 2D illustrations therefore, the research didn’t go into the same deapth as the game engine comparison. However the more popular software options were used to test which one was going to be most useful. The software tested consisted of Illustrator, Photoshop, Inkscape. Each one was used to created basic assets, the comparison and thoughts can be found in the appendices. The key elements that were trialled were the variety of options available and finding which one I was going to be able to use quickest and easiest. It soon became apparent that Illustrator was going to quickest way of creating simple assets and was the chosen application.

## Blender

Continuing from the need for assets brought the research to 3D modelling. 3D modelling is extremely complex and takes years to master, however the need for creating models with beautiful texturing was not one of concern. It would be ideal for a full release as it increases the engagement but this was not the case. Knowing very little about 3D modelling made it important to find a tool that was going to be easy, yet efficient to use. This is where Blender came to shine. Free to use, with thousands of tutorials online to help with the basic requirements, it was the software used throughout the designing and modelling phase.

## Competition Comparison

Next a competitor analyses was completed. It was beneficial to go through some existing turn based strategy games to gain a little inspiration. This research consisted of scoping out potential competition, whilst giving a better understanding of the initial project objectives. It was important to loosely compare the project idea against other fully developed game as they will have been completed by a full team opposed to one person. It was important to understand that the game being produced would not be finished to the same level and or polish as they required a team but knowing the future potential and building for expansion was interesting.

## Platform

Upon deciding on all the above, the need to focus the game towards a specific platform was still undecided. Due to picking Unity as the desired game engine this opened up the options to target the project to several different platforms. Unity can easily build the application into Android, IOS, console (Xbox, PlayStation), web or standalone PC. Although it would be ideal to have the game available for each and every option it needed to be narrowed down.

### Mobile IOS & Android

There has been a massive spike in popularity for mobile based applications and games. These are normally tailored towards people on the go, being intuitive to use, simple to maintain and engaging. Both IOS and Android are worthy candidates. However they might not produce the system power required to run the game. Knowing that Android alone has thousands of different devices all varying in sizes and specs it was enough to sway the decision against the mobile platform.

### Consoles Xbox & PlayStation

Next were the consoles. Both Xbox and PlayStation offer a lot more power than a mobile device and would comfortably be able to run any game developed through Unity. Despite their extended power the need to develop a game in such a way that is passes the criteria required by either Microsoft and or Sony proved problematic. Not only would it be mandatory to abide by their strict rules the game would need to be finished to a professional standard and the time required to do so was not available.

### Web & PC

Targeting the game towards the web would not have been a bad idea. It can be hosted on game sites, have a large audience ready and waiting, allow for instant exposure. However the need to play the game online is a large obstruction. Ideally the game will be able to be played offline as it maximises the opportunity to be played. This brought around the final choice of a stand along PC game. Offering the options of online and offline play, being run on a powerful machine and with no need to meet the expectations of an external source. Building a standalone PC game was seen to be the best option for the proposed game.

## Bayesian Probability

Lastly was the new focus of the project, the AI. This was going to be a very important part of the initial research as the developer had not completed any AI based programming beforehand. It was key to understand how different AI techniques could be used to really bring out the true potential of the game. After being provided with a little direction into which AI techniques could be appropriate, a research paper was completed outlining the potential AI techniques. See Appendices. Knowing which AI system or systems were going to best benefit the games overall performance was vital. The end result showed an adaptation of the Bayesian probability technique, in which the AI will assess its given situation, calculate the probability of a series of different outcomes and execute an appropriate response. See Appendices for more on Bayesian