# Statement of Key Risks

* Deadlines – Missing agreed or already defined deadlines could cause elements of the project to slip and create a backlog further down the line. Missing key milestones will likely mean that planned functionality will have to be removed closer to the final deadline to ensure working software can be delivered.
* Underestimating Deliverables – Projecting that a planned deliverable will take a lot less time than it takes in reality, particularly if it is a high priority feature, will mean less time to work on other areas of the software, reducing the overall quality.
* Commitment & Effort – Team members putting in less effort than required to work on their assigned jobs will either mean incomplete or low-quality features being implemented, or less work being done developing other features to bring this work to a reasonable standard.
* Poor Testing – If the testing strategy is poor, the end-product will have a significantly higher number of bugs. Testing will need to be consistent and thorough across all areas of the application. Similarly, not allocating enough time to the testing side of things will result in a big drop in overall quality.
* Poor Implementation – Implementing features in a poor or inefficient way can cause the software to be of low quality. The code also needs to be clear and maintainable, or too much time will be spent on trying to interpret code rather than adding to or fixing it.
* Not Understanding the Tools – The team will have to learn the tools being used for the project (e.g Unity, GitHub) or risk the final product being hastily put together in a poor way. Avoiding this will mean individual team members learning the chosen tools and understanding Unity before the development work begins.