

Grade	Feature Set (40%)	Implementation Quality (30%)	Source Control (20%)	Documentation (10%)
A	Excellent range of features that combine to build a near complete game. The submission is a well featured demo and there is evidence of experimentation with features beyond what is covered in class	Near flawless implementation that makes broad use of C# and Unity features, contains very few issues, and is clear and easy to parse. Comments are used to increase clarity. Classes and variables are sensibly named. The codebase is consistent across team members.	Excellent use of Version Control which shows regular commits and sensible commit logs. Branches are self-contained and used appropriately.	Insightful exploration of the chosen problems with a discussion of how they were solved and why the team believed the solutions were appropriate.
B	Very good selection of features that goes beyond the brief requirements and are fully realised within the engine. They combine to make well featured demo.	Very good implementation that makes broad use of C# and Unity features, contains few issues, and is generally clearly written. Comments are used to increase clarity. Classes and variables are sensibly named.	Very good use of Version Control which shows regular commits and sensible commit logs. Branches are used to separate work on features.	Very good exploration of the chosen problems with a discussion of how they were solved and why the team believed the solutions were appropriate.
C	Good selection of features that either go beyond the brief requirements or are fully realised within the engine.	Overall, good implementation that explores C# and Unity features but could be further improved. It may contain some flaws in logic, or other issues. Comments are used to increase clarity.	Good use of Version Control which shows mostly regular commits and sensible commit logs. Some comments may be lacking in detail. Branches are primarily used for the work of individuals.	Good exploration of the chosen problems with a discussion of how they were, or were attempted to be, solved but is not clear as to why the approaches taken were appropriate.
D	Satisfactory selection of features that meets the brief requirements.	Satisfactory implementation that makes basic use of C# and Unity features. It may also contain significant, but non-critical issues. The code might be confusing in places or poorly commented.	Satisfactory use of Version Control which shows semi-regular commits. Logs may be lacking in detail. Branches are sparse or entirely dedicated to developer branches.	Satisfactory exploration of the chosen problems with a discussion of how the problems manifested but not a clear identification of the solutions.
MF	Features do not meet the requirements of the brief, or are non-functional.	Overall, poor implementation. It may fail to build, or have other critical issues. The code might be confusing or poorly commented.	Poor usage of Version Control which shows sparse commits, poor commit logs, or very limited usage of branching.	There is little or no attempt to understand the problems and the solutions provided are not appropriate to the task.
F	Performance well below the expected threshold and does not meet the requirements of the brief.			
NS	Submission either missing entirely, does not evidence sufficient work or contains work unrelated to the assessment.			

% Weightings are indicative

Grade	Feature Set (40%)	Implementation Quality (25%)	Source Control (10%)	Documentation (25%)
A	Excellent range of features that combine to build a near complete game. The submission is a well featured demo and there is evidence of experimentation with features beyond what is covered in class	Near flawless implementation that makes broad use of C# and Unity features, contains very few issues, and is clear and easy to parse. Comments are used to increase clarity. Classes and variables are sensibly named.	Excellent use of Version Control which shows regular commits and sensible commit logs. Branches are self-contained and used appropriately.	Insightful exploration of the chosen topics that show a deep engagement with game development practices and strong problem solving skills.
B	Very good selection of features that goes beyond the brief requirements and are fully realised within the engine. They combine to make well featured demo.	Very good implementation that makes broad use of C# and Unity features, contains few issues, and is generally clearly written. Comments are used to increase clarity. Classes and variables are sensibly named.	Very good use of Version Control which shows regular commits and sensible commit logs. Branches are used to separate work on features.	Very good exploration of the chosen topics that show engagement with game development practices and strong problem solving skills. There may be flaws in some of your reasoning.
C	Good selection of features that either go beyond the brief requirements or are fully realised within the engine.	Overall, good implementation that explores C# and Unity features but could be further improved. It may contain some flaws in logic, or other issues. Comments are used to increase clarity.	Good use of Version Control which shows mostly regular commits and sensible commit logs. Some comments may be lacking in detail. Branches are too expansive or sparsely used.	Good exploration of the chosen topics that show engagement with game development practices and developing problem solving skills. There may be flaws in some of your reasoning or your understanding of problems may not be clear.
D	Satisfactory selection of features that meets the brief requirements.	Satisfactory implementation that makes basic use of C# and Unity features. It may also contain significant, but non-critical issues. The code might be confusing in places or poorly commented.	Satisfactory use of Version Control which shows semi-regular commits. Logs may be lacking in detail. Branches are very sparse or not present.	Satisfactory exploration of the chosen topics that show engagement with game development practices and some attempt at problem solving. There may be serious flaws in some of your reasoning or your understanding of problems may not be clear.
MF	Features do not meet the requirements of the brief, or are non-functional.	Overall, poor implementation. It may fail to build, or have other critical issues. The code might be confusing or poorly commented.	Poor usage of Version Control which shows sparse commits, poor commit logs, or very limited usage of branching.	There is little or no attempt elaborate on your development decisions or there is no discussion of problems encountered and solved.
F	Performance well below the expected threshold and does not meet the requirements of the brief.			
NS	Submission either missing entirely, does not evidence sufficient work or contains work unrelated to the assessment.			

% Weightings are indicative