

Team Name: .....

Date: .....

GitHub URL: .....

Team Members

Name: ..... StudentID:..... Name: ..... StudentID:.....

Name: ..... StudentID:..... Name: ..... StudentID:.....

Video Details: .....

**0-No feature, 1-Attempt, 2-Working, 3-Excellent**

	Each sub-component should be given 0-3 e.g., <input checked="" type="checkbox"/> 2	0-3
Demonstrate Understanding of HTML, CSS and Javascript	<input checked="" type="checkbox"/> Working Client/Server Game <input type="checkbox"/> HTML, CSS and Javascript validates fully (no syntax errors, warning, issues) <input type="checkbox"/> Creative use of CSS styling (which lines/files) <input type="checkbox"/> Advanced CSS3 layout (which lines/files) <input type="checkbox"/> Creative use of Javascript (which lines/files why creative) <input type="checkbox"/> HTML5 semantic elements used appropriately (evidenced, how, has the HTML been run through HTML validation software W3C and included in the submission) <input type="checkbox"/> Game/website should be responsive (also responsive in a mobile/different browsers) – Code for managing client/server delays, e.g., Interpolation, prediction, AI. Good used of asynchronous Javascript (avoid stalling GUI) <input type="checkbox"/> <b>Hand written code</b> (clean, structured, commented, ..) <b>*Your Code*</b> (e.g., all files commented, license/author details top `all` files, indentation consistent across `all` files, should be the same for all team members (project coordination))	
Client/Server	<input type="checkbox"/> Manage <b>communication</b> between multiple `concurrent` <b>client/server</b> (i.e., the games should not be running `separately` but shared interactive experience) – manage concurrent/client server interaction of data (e.g., collisions/data conflicts) <input type="checkbox"/> Handle issues (delays, network issues, corruption, ) Interpolation, message pausing the game (delay is over 5 seconds), smart AI system, ...	
Database/Security	<input type="checkbox"/> Data is managed for the website effectively and efficiently (Server/Database) <input type="checkbox"/> <b>Database (working reading/writing)</b> [stores variety of information, such as high scores, ] <input type="checkbox"/> Data is automatically backed up (server side), <input type="checkbox"/> Whitelist/validation testing for data submitted to server <input type="checkbox"/> Client also tests/validates data sent from server (corruption/value ranges avoid cheating) <input type="checkbox"/> Database security checks (all SQL commands are `security checked` avoid SQL injection) <input type="checkbox"/> Data sent/received is encoded/decoded/validated somehow (e.g., hash/pattern) <input type="checkbox"/> IP tracking (prevent spamming/attack logging) <input type="checkbox"/> Login system (user can register and login/store/continue game progress) <input type="checkbox"/> Reset/restore password (maximum allowed tries, attack prevention) <input type="checkbox"/> User passwords hash encoded with salting (or similar complex encryption)	

Code Structure	<input type="checkbox"/> Clear separation of code/styling <input type="checkbox"/> <b>Tidy code organisation</b> , file and folder names (js folders/css folder, ..) <input type="checkbox"/> Clear variable names, constants for fixed values, avoid using `magic` hard coded numbers scattered around code, .. <input type="checkbox"/> Appropriate code nesting and indentation (methods with clear purposes, descriptions/comments – avoid adhoc/hack fixes), scoping and not having everything `global` <input type="checkbox"/> Informative code comments (readme, `all` files commented, every function, style, ..)	
Usability	<input type="checkbox"/> Easy to navigate and use (Website/Game) (Accessibility testing? Different browsers, screen resolution, mobile/desktop, language - `evidenced` – detailed in the documentation) <input type="checkbox"/> Clear awareness of accessibility principles (evidenced in the website, e.g., about/help page, also in the documentation/readme/code) <input type="checkbox"/> Disclaimers/warnings – e.g., photosensitive epilepsy/flashing images <input type="checkbox"/> Effective navigation at all sizes, content easy to locate Font size (scales to different screens, not just `fixed/hardcoded`), text/information not in margins, bright/animated buttons/text so easy to see/identify on the screen <input type="checkbox"/> Limitations and optimizations Profiling data statistics, jpg vs bmp, appropriate file sizes/download considerations (avoid downloading everything for anything), sending only required data to specific clients (not all data to all clients)	
Development Progress	<input type="checkbox"/> Shows even work distribution (group work) <b>(evidence on code written/bugs fixed/testing/...)</b> <input type="checkbox"/> Evidence of task development, tuning, feature refinement (week by week log of task development, i.e., not the last week or 2 weeks – over the full duration of the project) <input type="checkbox"/> Live site (Github website) (GitHub page, with readme, and access to material/source files) <input type="checkbox"/> Project management (team synergy, evidence of team working together on GitHub, helping/each other, such as, task lists, issues lists, bug fix lists available on GitHub over the duration of the project)	
Testing	<input type="checkbox"/> Testing integrated into the project from the start (as each feature was added a set of tests was defined/included in the code/documentation) <input type="checkbox"/> Validation/verification process for ensuring reliability (e.g., code, standards, ..) <input type="checkbox"/> <b>Documentation on the tests</b> (e.g., where delays added to the server to simulate poor connections, automatic errors added to the send/receive data, soak testing (left running 3+ days), larger number of instances created to simulate 100s or 1000s of simultaneous users)	

Notes/Comments (Extra Features)