ENCE260 Embedded Systems Assignment: Demonstration Marking Guide **Group #** _______
Member 1 Name: ________________________ **Member 2 Name**:________________________

A. Readiness & documentation:

Marks [0] Not ready to demonstrate, or group member names not in Makefile & source files.

Marks [1] Ready and group member names in Makefile & source files.

B. git usage:

Marks [0] No source files on gitLab server.

Marks [1] Source files on gitLab, but not multiple commits by both partners or object/hex/bin files are also included.

Marks [2] Multiple commits by both partners, and only source files, the Makefile and README included.

C. git README (including AI statement of use)

Marks [0] The README is insufficient to be able to understand how to play the game or AI statement of use is not included.

Marks [1] The README is sufficient to be able to understand how to play the game and AI statement of use is included.

D. Program compilation and download to the UCFK4:

Marks [0] Program fails to compile.

Marks [1] Program compiles and downloads but issues warning messages.

Marks [2] Program compiles and downloads without warning messages.

E. Communications:

Marks [0] No Infrared communications.

Marks [1] Infrared communications but only in one direction.

Marks [2] Infrared communications in both directions but intermittent or unreliable operation.

Marks [3] Reliable infrared communications.

F. Interactivity & Complexity:

Marks [0] None: A UCFK4 game has not been created.

Marks [2] Single player: A game has been created, but has no interactivity between players.

Marks [4] Basic: one-time interaction between players, e.g. Rock-Paper-Scissors game.

Marks [6] Good: some interaction between players (turn based), e.g. Noughts and Crosses.

Marks [8] Very good: highly interactive between players, e.g. a ball/missile based game.

Marks [10] Excellent: highly interactive, and highly original.

G. Operation

Marks [0] None: an operational UCFK4 game has not been created.

Marks [2] Basic: some functionality, but the game crashes or is not intuitive to play.

Marks [4] Very good: moderate functionality, and is moderately intuitive to play.

Marks [6] Excellent: extensive functionality, and is intuitive to play.

TOTAL: ______/25 Semester 2 2025 Marked by:

ENCE260 Embedded Systems Assignment: Source Code Marking Guide

A. Repository Usage

Marks [0] From the files in the git repository, the program does not compile.

Marks [2] From the files in the git repository, the program compiles, but there are warnings, or there are API functions in the repository, or there are dependency errors that needed correcting.

Marks [4]. From the files in the git repository, the program compiles without warnings. Only files written by the group are in the repository (ie no API files).

B. Formatting

- Is the program indented by the same amount for each block?
- Is whitespace used consistently? Are braces used consistently?

Marks [0] Minimal care taken.
Marks [1] Meticulous (apart from 1 or 2 whitespace inconsistencies).

C. Commenting

- Is there a banner at the top of the file listing the authors' names and what it does?
- Does each function have a comment explaining its purpose?
- Are the comments well formatted, consistently formatted, relevant and meaningful?
- Is the program over-commented? For example, do most lines have a comment?
- Are there any inappropriate comments? For example, 'add one to i'?

Marks [0] No comments.
Marks [1] Only a few comments or many inappropriate comments.
Marks [2] Good attempt at comments but with poor format.
Marks [3] Good, well formatted comments.
Marks [4] Excellent, well formatted comments.

D. Naming

- Are the variables/functions/constants named consistently?
- Do the variables/functions/constants have meaningful names?

Marks [0] Random or meaningless names.
Marks [1] Some variables, functions, and constants have consistent meaningful names.
Marks [2] Most, functions, and constants have consistent meaningful names.
Marks [3] Almost all variables, functions, and constants have consistent meaningful names.
Marks [4] All variables, functions, and constants have consistent meaningful names.

E. Constants

- Does the program use unnamed constants (magic numbers)?

Marks [0] No use of named constants.
Marks [1] Minimal use of named constants.
Marks [2] Good use of named constants.
Marks [3] Very good use of named constants.
Marks [4] Excellent use of named constants.

F. Structure

- Can you quickly figure out how to use the module? Are there nested while(1) loops?
- Does each module have high cohesion. Is there low coupling between modules?
- Are things (functions, constants) that should be private but are public?

Marks [0] No attempt at using a module.
Marks [2] An attempt at using a module but of no use to anyone.
Marks [4] The module may be useful but is either trivial or hard to use.
Marks [6] The module is not trivial and is easy to use.
Marks [8] There are multiple modules that are easy to use.

TOTAL: ______/25 Semester 2 2025 Marked by: