**Graphical user interface, text, application

Description automatically generated**

**MICROPROCESSOR SYSTEMS**

[**CSU23021-202122**](https://tcd.blackboard.com/webapps/blackboard/execute/courseMain?course_id=_72181_1)

**ASSIGNMENT #2**

**PROJECT GROUP #15**

**Team**

Member #1: Ananya Garg Member #3: Angad Singh Garcha

TCD ID: 21355233 TCD ID: 21354935

Member #2: Mudit Garg Member #4: Rahul Garg

TCD ID: 21355125 TCD ID: 21355038

**GitLab Repository Link**

<https://gitlab.scss.tcd.ie/gargmu/a2-morse-code-game>

**Introduction**

The project consists of a simple interactive game made using a mixture of C code and ARM assembly that teaches the player, Morse Code. Morse code is a system of electronic communication that uses dashes, dots and spaces to represent letters, numbers and punctuation. These dots, dashes and spaces are arranged to spell out a message, slashes are used as spaces to separate words.

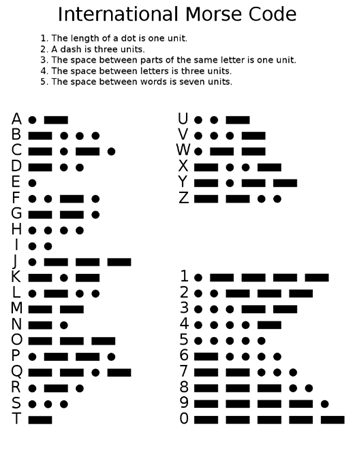
The game comprises of 4 levels whose descriptions are as follows:

**Level #1:** Individual characters with their equivalent Morse code provided.

**Level #2:** Individual characters without their equivalent Morse code provided.

**Level #3:** Individual words with their equivalent Morse code provided.

**Level #4:** Individual words without their equivalent Morse code provided.

  
The morse code translation from characters (A-Z / 0-9) in international format is given for your reference.

**Workflow**

The following individual project roles were distributed amongst ourselves at the start of the project.

* **Project Code Owner**: Mudit Garg

As the project code owner, I was responsible for ensuring that the overall application functions as expected. I got assigned the duty to integrate every function and subroutine that others made, as per their assigned duties, into the overall code to make everything errorless and enable the building of the project. Everyone in the team contributed to implementing the code required for the application and the team efforts were clear from the beginning.

* **Project Workflow Owner**: Angad Singh Garcha

As the project workflow owner, I had to look after the overall organisation of how the project should be run.  I worked with the other functional owner(s) to ensure the overall project was on target, everyone is in regular communication and the individual project tasks have all been allocated and are on track for completion by the milestone dates.

* **GitLab workflow Owner**: Rahul Garg and Mudit Garg

As the GitLab workflow owner, we were responsible for creating a new shared GitLab project repository for your team, providing access to your project team members, and managing code merge requests to the repository. The code merge requests directly went through Rahul whereas the handling of the repository was under Mudit. The other project team members cloned this repository directly so they can work on their allocate code portions individually and make regular pull requests (using Git feature branches) back to the project repository. Rahul ensured that all pull requests get correctly merged back to this repository.  At the end of the project, we shared the overview of the Git history for Ananya (the documentation owner) to include in the final project report.

The overview of Git history of our repository (it is also present in the repository as a separate file) is mentioned in the GitLab section of the report.

* **Project Documentation Owner**: Ananya Garg

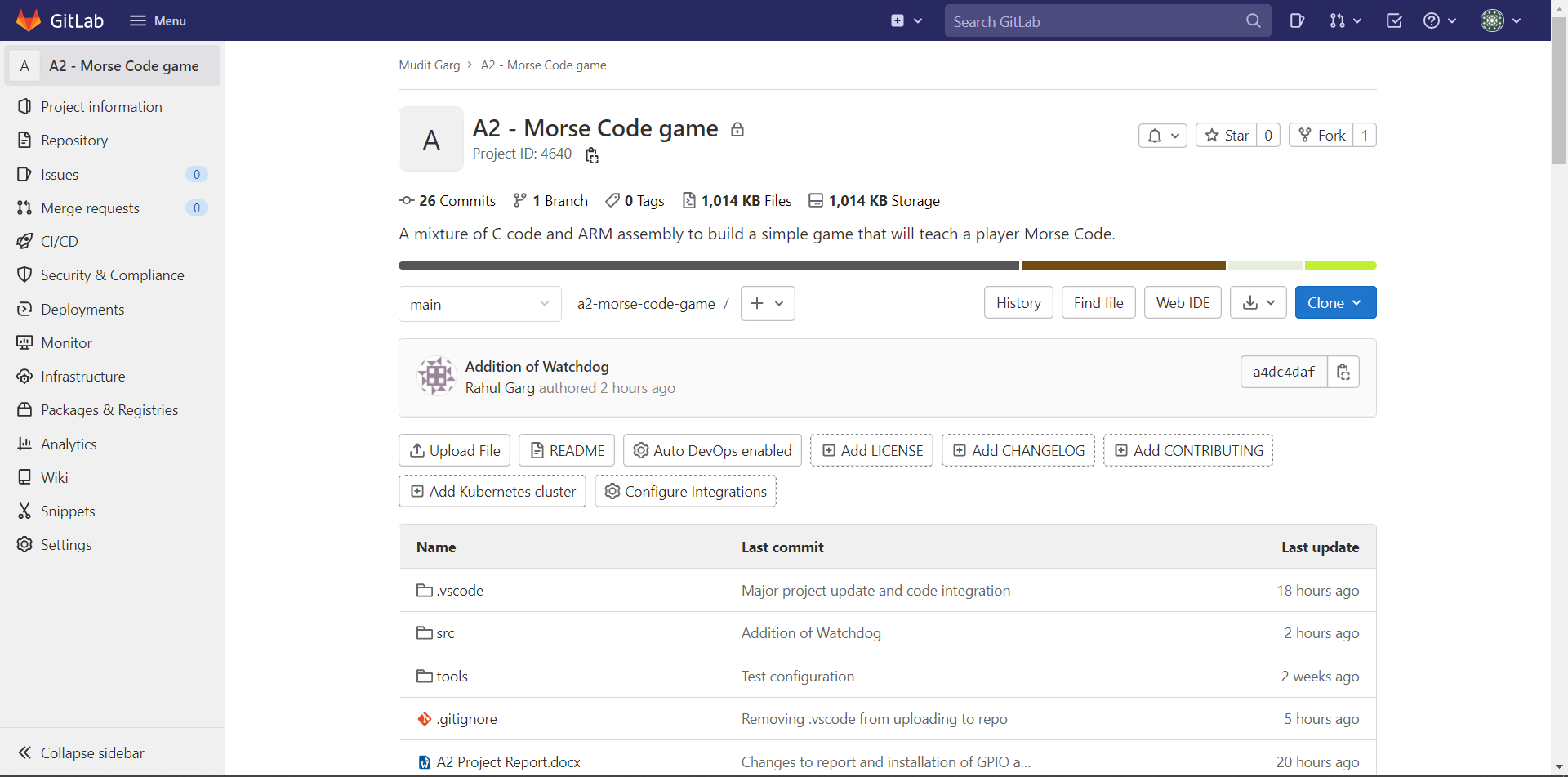
As the project documentation owner, I was responsible for compiling the required documentation and final project report from my team. Each member of the project team individually documented their contributions to the project during the whole project duration and for being a part of the report, conveyed it to me. I set up the initial template and structure of the report in the same repository as the main code project as that’s the location where the team members can and usually access frequently.

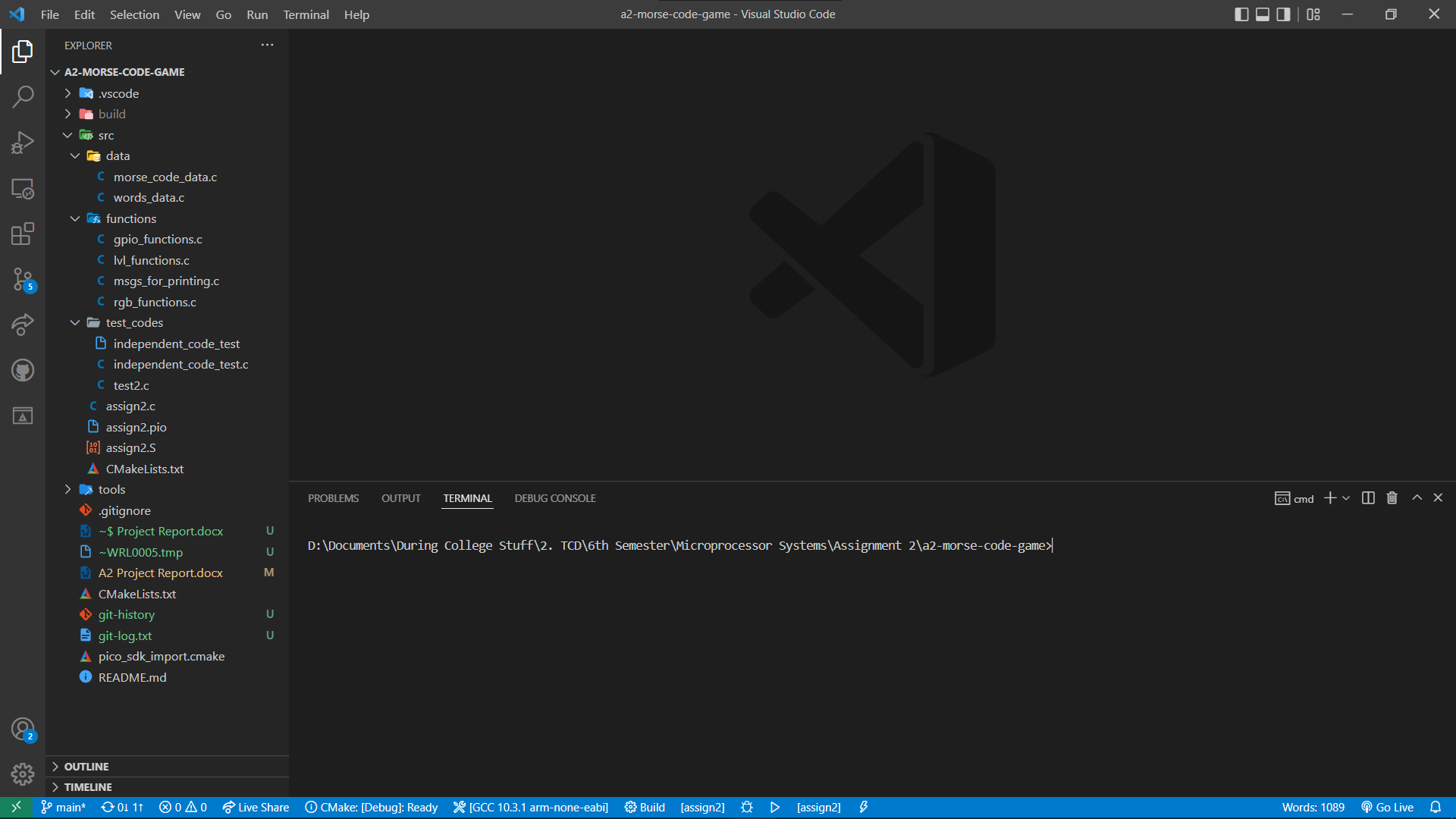
* **Project Demonstration Owner**: Angad Singh Garcha and Ananya Garg

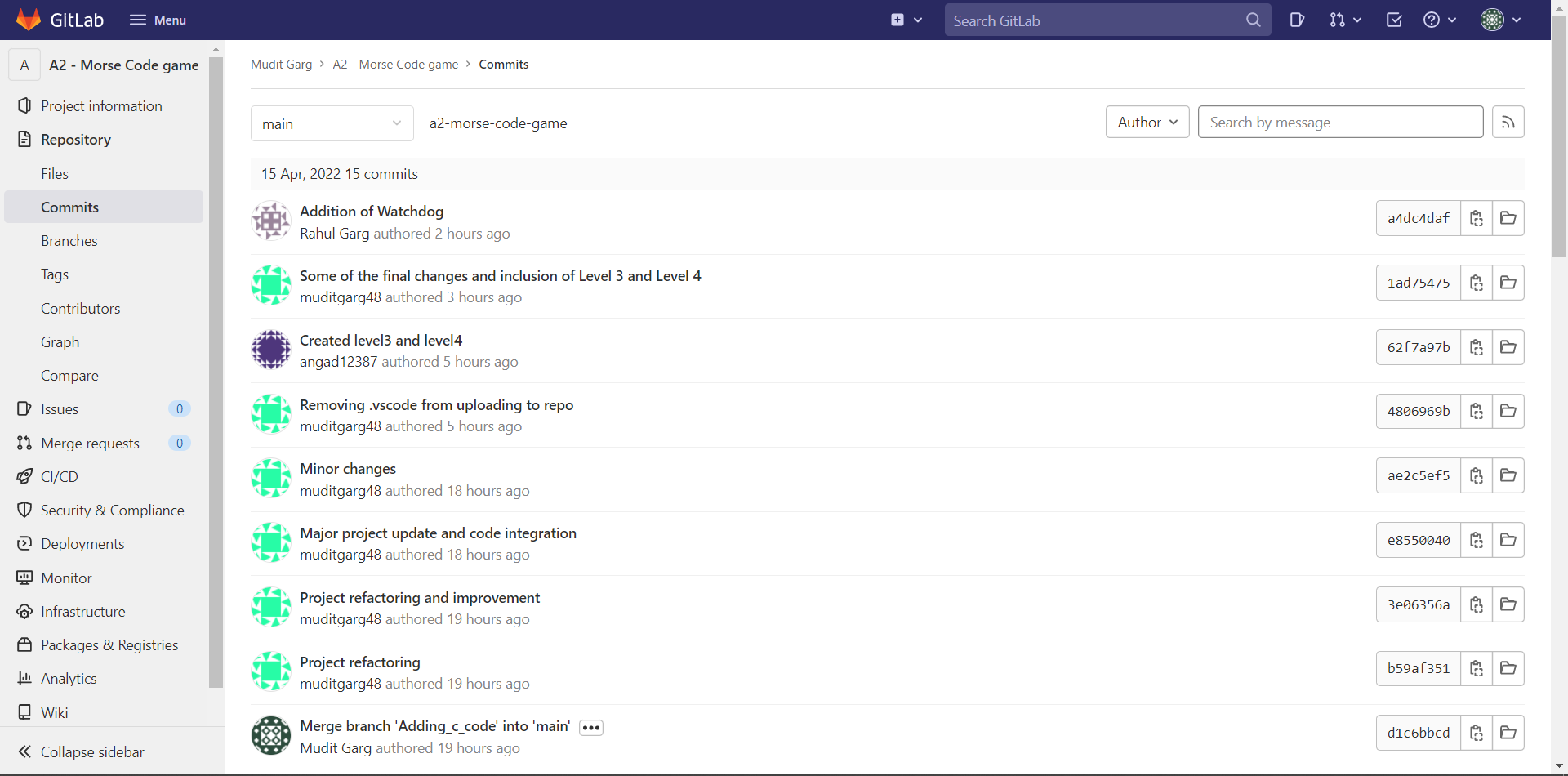
As the project demonstration owner, I was responsible for creating a short (90 sec max) video demonstration of the completed code project build. The demonstration showcased all the mentioned functionality that was decided in the project outline.

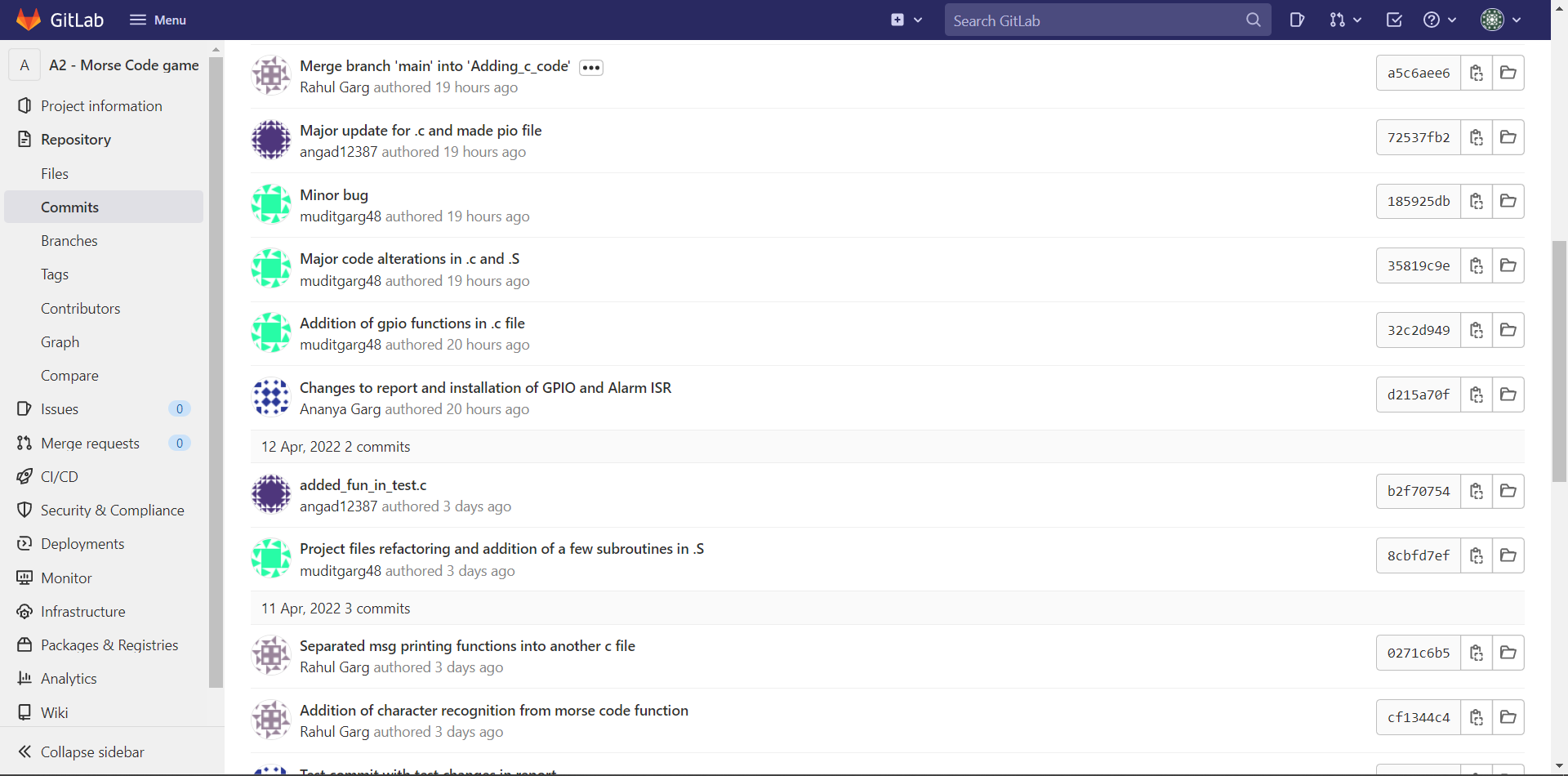
**GitLab**

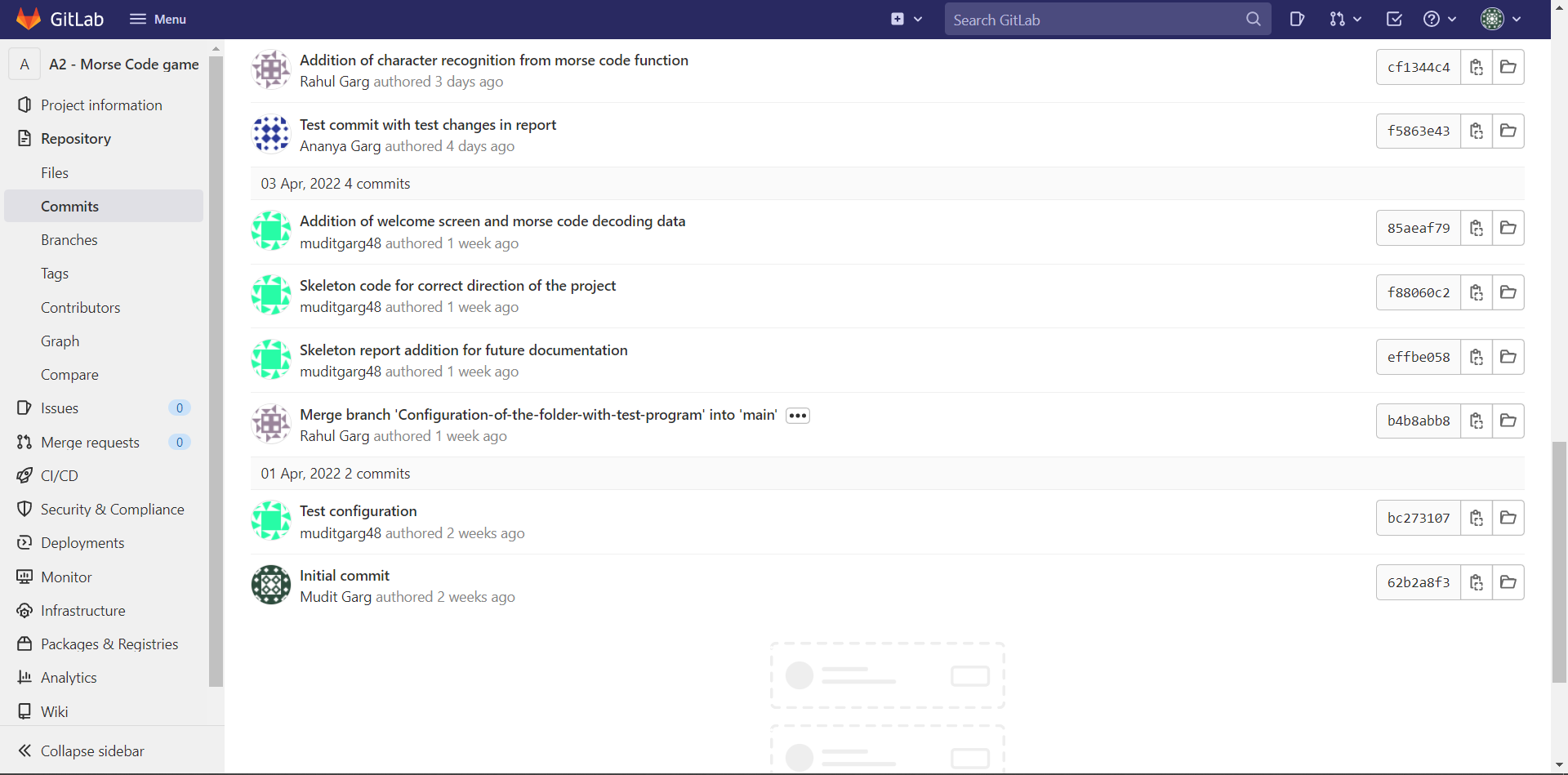
Link: <https://gitlab.scss.tcd.ie/gargmu/a2-morse-code-game>









****

**Code**

* Mudit Garg:
* I configured and designed the folder structure of the project repository for it to function properly with GitLab and CMake to build the source files.
* I was responsible to integrate the code contributions received from various team members into the project for their proper functionality. I always double checked whether the code project remains errorless and builds perfectly with return code 0.
* I was also responsible to code the ARM part of the project that was to get the input from user. I inspired it from the previous lab codes and the solution of Assignment 1 provided on Blackboard.
* I regularly restructured the project folder such that the functions are separated according to their category and the data of morse codes and words for level 3 and 4 into different .c files to reduce congestion in the main assign2.c.
* Angad Singh Garcha
* I was responsible for writing the code that made each of the levels among Level 1, Level 2, Level 3 and Level 4 work.
* I also coded the part where the program reads the number of lives left for the player to consume and display the RGB accordingly.
* I was also responsible to lookout for my fellow team mates for any errors that pop up while building the project source codes.
* Rahul Garg
* I was assigned to design the functions that was responsible to get the morse code string as input and return whether the morse code exists in the morse code database. If the following morse code exists in the database, it returns the index of the morse code and that can be used to determine whose morse code it was. If the morse code is not found, it returns -1 which means the morse code entered is not present in the database.
* I also coded the function that was responsible for recognising the string’s morse code that looped over the morse code and check which word did the user intend to input and check if that’s what word was expected or not.
* I also installed the Watchdog into the code so that to introduce a timeout such that if user does not input anything for 9s, the code is interrupted.
* Ananya Garg
* I was responsible for coding the subroutines that installed the GPIO and the Alarm interrupts and give them to Mudit to add it into the ARM part of the project.
* I was also given the task to make the message prints on the console using printf and make separate .c file for it so as to reduce congestion.