

Programming Project / C++

- Hangman Game -

Student 1: Sandor Paul-Filip

Student 2: Popelca Andrei Ricardo

I. Task Description

Student 1 is responsible for managing shared resources:

- Adding, deleting, listing words in the dictionary (words.txt);
- Resetting or viewing the leaderboard and match history.

Student 2 is responsible for interacting with the game:

- Running Hangman rounds and calculating scores;
- Recording scores/ history and updating the leaderboard.

Bidirectional flow:

- App 1 modifies words.txt → App 2 immediately plays with updated words;
- App 2 records scores in leaderboard.txt & history.txt → App 1 can view them.

II. Data Structures Used by the Team

- **WordRepository** – holds `std::vector<std::string>` words; functions `load()`, `save()`, `add()`, `erase()`, `random()`.
- **Player** – members `std::string` name, `int` score.
- **Leaderboard** – keeps `std::vector<Player>` table; functions `update(name, pts)`, `sort()`, `reset()`, `load()`, `save()`.
- **HistoryEntry** – fields date, time, name, gained, total.
- **HistoryLog** – stores `std::vector<HistoryEntry>`; functions `append(entry)`, `list()`, `load()`, `save()`.
- **GameEngine** – has refs to `WordRepository` + `Leaderboard` + `HistoryLog`; main method `play_round()`, awards **points = word length**.
- **AdminTool** – same three refs; method `cli_dispatch()` to run admin commands.

Leaderboard aggregates `Player`; both front-end classes associate with the shared service classes.

III. File Structure

words.txt

Each line: <word> (lowercase, no spaces)

leaderboard.txt

Each line: <player_name> <total_score>

history.txt

Each line: <YYYY-MM-DD> <HH:MM> <player_name> <points_gained>
<total_score_after_game>

All files are plain text so both apps can append / read without binary dependencies.

IV. Interacting with Executables

Application 1: hangman_admin.exe

hangman_admin.exe list_words

hangman_admin.exe add_word <word>

hangman_admin.exe delete_word <word>

hangman_admin.exe reset_leaderboard

hangman_admin.exe view_leaderboard

hangman_admin.exe view_history

Application 2: hangman_game.exe

hangman_game.exe play # starts a game session

hangman_game.exe view_leaderboard # read-only

hangman_game.exe view_history