Programming Project

C++ - Console Game Management System

Student I: Balaș Andrei-Mario
Student II: Fazecaș Joshua-Manuell

I. Task Description

Student 2

Student 1 is responsible for game management:

- installing/uninstalling games
- viewing a list of installed games

Student 2 is responsible for profile management:

- create/delete profiles

II. Data Structures Used

The following classes will be used:

- **Game**: `string title, int size`
- **Profile**: `string name, Game[] games, int memory_space`

III. File Structure

The following files will be used:

available_games.txt

A file where the available games and their size will be stored:

```
<game title> <size GB>
...
```

profiles.txt

A file with all the profiles and their size and game lists. By default, it has a single profile named "Default":

```
Default <size GB>
<game 1>
<size game 1>
...
<profile 2>
<game 1>
<size game 1>
<game 2>
<size game 2>
...
```

IV. Interacting with Executables

Application 1 will offer the following options:

```
./app_1.exe view_installed_games
To view all installed games in all profiles

./app_1.exe install_game <profile_name> <game_title>
To install a game into a specific profile, if enough memory is available

./app_1.exe uninstall_game <profile_name> <game_title>
To uninstall a game from a specific profile

./app_1.exe view_available_games
To display all games available for installation from available_games.txt
```

Application 2 will offer the following options:

```
./app_2.exe view_profiles
To display all profiles, their memory usage, and installed games

./app_2.exe create_profile <profile_name> <memory_space>
To create a new profile with the specified name and memory space

./app_2.exe delete_profile <profile_name>
To delete an existing profile and all associated installed games

./app_2.exe view_profile_details <profile_name>
To view details (installed games, memory usage) of a specific profile
```