**Triviador – Profiling aplicație**

1. **Prezentarea aplicației**

După bine-cunoscuta aplicație deja existentă „Triviador”, ne-am propus să dezvoltăm o replică a acesteia în care am adus mici modificări cum ar fi existența variantelor ajutătoare. Jucătorul are acum posibilitatea de a utiliza varianta 50/50 care îi înjumătățește variantele de răspuns în cazul întrebărilor cu răspuns grilă și sugerare răspuns și alegere răspuns pentru întrebările numerice. Sugerare răspuns îi furnizează utilizatorului un răspuns apropiat de cel corect, iar alegere răspuns îi conferă posibilitatea de a alege răspunsul corect din 4 furnizate de server.

Implementarea părții de rețelistică respectă arhitectura client-server și permite desfășurarea jocului de la 2 până la 4 persoane față de varianta clasică unde erau doar 3 jucători. Pentru gestionarea utilizatorilor cât și a întrebărilor ne-am folosit de o bază de date. În această bază de date reținem toate întrebările, atât cele numerice cât și pe cele cu variante multiple alături de posibilele lor răspunsuri. Tot în această bază de date reținem și informațiile legate despre un user cum ar fi: id-ul, numele și parola acestuia. Toate acestea se găsesc în folderul **Database**.

Partea de rețelistică este implementată în folderul **Server** și ea asigură comunicarea jucătorilor cu serverul și implicit unul cu altul. Serverul este cel care furnizează întrebările jucătorilor, care reține timpii de răspuns ai acestora, dar și cel care gestionează întregul joc și ține cont de starea fiecărei regiuni.

În folderul **UI** regăsim implementată partea grafică a jocului. Aici se află toate ferestrele pe care le vede jucătorul pe parcursul unui meci precum fereastra de start, de logare sau înregistrare, fereastra în care este afișată harta sau cele în care trebuie să răspundă la întrebări. Biblioteca aleasă de noi pentru a dezvolta partea grafică a fost Qt.

1. **Unelte folosite**

* Visual Studio 2019
* CMake
* vcpkg - package manager
* sqlite - baza de date - instalată prin vcpkg
* Qt 6.4.0

Pentru studierea performanței aplicației am ales sa folosim tool-urile oferite de Visual Studio. În analiza pe care am ales să o facem am decis să studiem cum se prezintă GPU, CPU și memoria utilizată pe parcursul etapei de alegere a bazei și a selectării teritoriilor.

1. **Rezultatele după procesul de testare**

* pentru 2 jucători

Timeline

Description automatically generated

Deoarece aplicația pe care am dezvoltat-o este amplă am ales să exemplificăm mai sus performanțele obținute în urma etapei de alegere a bazei. Pe parcursul celor 3 minute se poate observa în graficul de GPU creșterea bruscă în momentul în care după ce ambii jucători își introduc răspunsurile și acestea ajung la server. Memoria utilizată pe parcursul procesului este relativ constantă deoarece întregul proces este unul uniform. Valoarea CPU utilizată este foarte mică, procesul nefiind unul costisitor.

* pentru 3 jucători

Chart

Description automatically generated

Față de varianta de mai sus unde performanțele au fost testate când aplicația era utilizată de 2 jucători, în cazul în care numărul acestora crește la trei se observă și variațiile mult mai bruște ale GPU. Atât memoria utilizată cât și procentul de CPU au crescut. Ținem să precizăm că rezultatele au fost obținute pentru testarea pe același device.