**WAgI (We're Against Ignorance)**

**Membrii echipei:**

Radu Vulpescu (radu.vulpescu16@gmail.com)

Răzvan Cimpoeșu (razvan.cimpoesu96@gmail.com)

Adrian Dorneanu (adrian.dorneanu96@gmail.com)

Alexandru Corfu (corfu.alexandru01@gmail.com)

**Cerințele proiectului:**

Să se imagineze un joc educativ de tip shooter ce va putea fi jucat direct în cadrul navigatorului Web. Entităţile care vor fi "anihilate" (fiecare "ţintă" valorând un anumit număr de puncte sau declanşând un eveniment specific de genul "notifică atunci când apar răspunsuri pertinente") vor putea fi întrebări "dificile" plasate pe situri de profil precum StackExchange (Stack Overflow).

Scorul acumulat în urma cumulării voturilor pozitive acordate de alte persoane va putea fi salvat local şi partajat pe o reţea socială.

**Bonus:** starea curentă a jocului va fi reprezentată grafic recurgând la diverse maniere de vizualizare a datelor.

**Bonus:** redarea stării jocului în format vectorial cu SVG şi/sau 3D (recurgând la WebGL).

**Bonus 2:** folosirea realităţii îmbogăţite (augmented reality) pe baza specificaţiei WebVR.

Maniera de interacțiune cu utilizatorul:

Un utilizator al site-ului nostru va avea următoarele opțiuni:

-**Home:** reprezintă o pagină dedicată informațiilor generale cu privire la joc (posibile update-uri și un scurt rezumat legat de conținutul lor)

-**Play:** pagina care conține jocul propiu-zis. Utilizatorul, dacă nu s-a logat, va avea posibilitatea de a introduce un username care va reprezenta alias-ul acestuia în joc. Toate punctele acumulate vor fi salvate sub acel username.

-**About:** pagină care prezintă: proiectul(cerințele acestuia) și echipa(poze & scurtă descriere a fiecărui membru)

-**Contact:** pagină unde prezintă adresele de contact a membrilor

-**Highscore:** pagină unde se salvează scorurile tuturor jucătorilor, astfel utilizator va avea posibilitatea să vadă poziția acestuia în clasament

Maniera de folosire a sistemului de management al codului-sursă:

**Arhitectură:** Am proiectat aplicația folosind o arhitectură de tip MVC. Astfel, fiecare request al utilizatorului va fi preluat de controler ce va face update-urile necesare la Modul si View. Modulul reprezintă partea logică a aplicației, care interacționează și cu baza de date, în timp ce View-ul reprezintă contactul direct cu utilizatorul.

Detalii referitoare la modelul datelor, API-uri externe, soluții de stocare, etc.

La capitolul API-uri externe, vom utiliza (conform cerinței) API-ul pus la dispoziție de StackExchange. Astfel, cu ajutorul acestuia, vom putea prelua întrebările postate pe platforma respectivă, împreună cu cel mai bun/votat răspuns aferent întrebării. Aceste date vor fi folosite mai departe la calcularea scorului final.

Ca și soluții de stocare, vom avea stocarea username-ului într-o bază de date, bază ce va fi folosită și la generarea scorului.

**Wireframe-uri:**

