

LLENGUATGE DE MARQUES: PROJECTE FINAL

El projecte s'ha d'acompanyar d'un dossier que expliqui:

- a. Quina funcionalitat té la nostra aplicació web i explicació del seu ús.
- b. D'on extreu la informació i com està estructurada.
- c. Possibles millores de l'aplicació.

INTRODUCCIÓ

Per aquest projecte vaig tenir diferents idees per portar a terme.

En primer lloc i amb el que vaig passar la major part del temps va ser fer una pàgina web que em permetís buscar tot tipus de tornejos atp i wta durant l'any així com també als meus jugadors preferits, desgraciadament no vaig poder aconseguir quadrar l'API (la gran majoria no eren gratis) i la que vaig aconseguir no em va servir per fer el que tenia en ment (només vaig aconseguir unes tarjetes que mostraven els diferents jugadors atp i wta, però res més).

Per això vaig explorar altres opcions i vaig trobar la idea de fer un trivial interactiu, que més tard vaig remodelar perquè s'assemblés al famós joc ¿Quién quiere ser Millonario?. Aquest és el resultat.

Anàlisi de la meva aplicació web "Who Wants to Be a Millionaire"

He desenvolupat aquesta aplicació web com una versió interactiva d'un trivial inspirat en el famós concurs televisiu *Who Wants to Be a Millionaire?*. L'objectiu principal és que l'usuari respongui correctament a una sèrie de preguntes per acumular premis en efectiu virtual. A continuació, explico com funciona, d'on extreu la informació i quines millores podria implementar en futures versions.

Funcionalitat i ús de l'aplicació

Quan un usuari accedeix a la meva pàgina web, es carrega la interfície amb el títol del joc, una taula on apareixen les preguntes i opcions de resposta, un indicador dels guanys acumulats i un botó per avançar a la següent pregunta. El procés de joc segueix aquests passos:

Carregament de preguntes: L'aplicació realitza una petició a l'API *Open Trivia Database* (*OpenTDB*) mitjançant la funció *fetch()*, obtenint un conjunt de deu preguntes aleatòries amb les seves respostes.

Visualització de la pregunta actual: La primera pregunta s'insereix dins la taula de la pàgina i es mostren les possibles respostes en files individuals. Aquestes respostes es barregen perquè no apareguin sempre en el mateix ordre.

Selecció d'una resposta: Quan l'usuari clica sobre una de les respostes, es comprova automàticament si és correcta. Si ho és, el color de la casella es torna verd, apareix un missatge de confirmació i es multiplica el premi. Si és incorrecta, el color es torna vermell i es mostra un avís d'error. Un cop s'ha seleccionat una resposta, es desactiven els clics per evitar que l'usuari pugui triar més d'una opció.

Passar a la següent pregunta: Quan l'usuari prem el botó "*Següent Pregunta*", es carrega la pregunta següent. Si ja s'han respost totes les preguntes disponibles, apareix un missatge informant que el joc ha finalitzat i es torna a començar des de zero.

Sistema de puntuació: L'usuari comença amb 0€ i, si respon correctament, guanya 100€ en la primera pregunta. A cada resposta correcta, el premi es duplica (100€, 200€, 400€, 800€...). Aquest marcador s'actualitza i es mostra en temps real en el contenidor de guanys.

Origen de la informació i estructura de les dades

Les preguntes i respostes de l'aplicació provenen de l'API pública *Open Trivia Database* (*OpenTDB*). L'URL utilitzat és:

<https://opentdb.com/api.php?amount=10&difficulty=easy>

Aquesta API retorna un JSON amb un array de preguntes i les seves opcions de resposta. La informació està estructurada de la següent manera:

```
{  
  "category": "Science",  
  "type": "multiple",  
  "difficulty": "easy",  
  "question": "What is H2O commonly known as?",  
  "correct_answer": "Water",  
  "incorrect_answers": ["Oxygen", "Hydrogen", "Helium"] }
```

"question" → Conté la pregunta en format HTML, que haig de decodificar perquè es mostri correctament.

"correct_answer" → La resposta correcta.

"incorrect_answers" → Un array amb les respostes incorrectes.

A la meva aplicació, la pregunta s'assigna a la secció corresponent dins de la taula. Després, combino la resposta correcta i les incorrectes, les barrego i creo elements HTML dinàmicament per mostrar-les dins la taula. Quan l'usuari selecciona una resposta, comparo el seu contingut amb "correct_answer" per determinar si és correcta o no.

Possibles millores per a l'aplicació

Encara que l'aplicació ja funciona correctament, crec que es podrien afegir algunes millores per fer-la més atractiva i interactiva. Aquestes són algunes de les millores que tinc en ment:

Millorar l'experiència de l'usuari (UX)

Ara mateix, faig servir *alert()* per indicar si una resposta és correcta o incorrecta, però això pot ser molest. Seria millor mostrar aquests missatges dins de la pròpia pàgina, per exemple, amb un `<div id="message"></div>` que canviï de color segons si la resposta és correcta o incorrecta.

Afegir un comptador de temps per cada pregunta

Actualment, l'usuari pot pensar tant com vulgui abans de respondre. Per fer el joc més emocionant, podria afegir un comptador regressiu de 10 segons per cada pregunta utilitzant *setInterval()*. Si el temps s'esgota abans que respongui, la pregunta es donaria per fallada i es passaria a la següent.

Permetre triar categories i dificultats

Ara mateix, totes les preguntes són de dificultat "easy" i de categories aleatòries. Crec que seria interessant afegir un menú desplegable `<select>` on l'usuari pugui triar la dificultat i el tema de les preguntes (Història, Ciència, Geografia, etc.). Això faria l'aplicació més personalitzable.

Millorar l'estructura visual i accessibilitat

Actualment, faig servir una taula per mostrar les preguntes i respostes, però això no és ideal per a mòbils. Una millora seria substituir la taula per targetes (`<div class="card">`) amb un disseny més modern i responsive. Així, les respostes podrien aparèixer com a botons grans, fent que sigui més fàcil jugar des d'un mòbil o una tauleta.

Afegir un sistema de puntuacions guardades

Ara mateix, quan es refresca la pàgina, es perd el marcador de punts. Una millora seria utilitzar *localStorage* per guardar la puntuació més alta, així l'usuari podria intentar superar el seu propi rècord.

Conclusió

Estic satisfet amb el resultat d'aquesta aplicació perquè compleix els requisits bàsics: consumeix una API pública, mostra la informació de manera dinàmica i gestiona esdeveniments d'usuari. A més, el disseny visual s'assembla al format original del concurs.

Tot i això, sé que hi ha marge per a la millora, especialment en l'experiència d'usuari, el disseny responsive i l'afegit de funcionalitats com el temporitzador o la selecció de categories. En el futur, m'agradaria implementar aquestes millores per fer que el joc sigui encara més divertit i atractiu.