

Nom : Alexandre Dufour

Matricule : p1054564

IFT3225 – Technologie de l'internet

Devoir 1

Lien internet du jeu :

<https://www.webdepot.umontreal.ca/Usagers/p1054564/MonDepotPublic/IFT3225/Devoir1/devoir1.html?uniq=-sus38s>

Technologies utilisées :

Bien sûr, à priori, HTML5 et CSS sont utilisés par défaut, quoi que beaucoup d'éléments de style sont utilisés de Bootstrap tel que le *Jumbotron* par exemple.

Bootstrap : Bootstrap fut utilisé pour la mise en page des éléments, et plus particulièrement pour les éléments de connaissance, c'est-à-dire tous ce qui concerne les notes et tonalités. Pour être efficace sur tablette et téléphone, tous les éléments de questions et réponses ont la classe `.col-sm-1`. Ainsi, sur tablette et ordinateur, il y aura 12 éléments par ligne au minimum. Pour que la mise en page soit plus efficace sur téléphone, j'ai utilisé `.col-xs-3`. Ainsi, sur les écrans très petits, tel que les téléphones, on retrouve au maximum 4 éléments par ligne, ce qui évite le problème d'avoir 1 élément par ligne, et ainsi d'avoir une page internet très longue et plate.

JQuery/Javascript : JQuery a été utilisé dans l'ensemble du code javascript pour faciliter l'acquisition des différentes valeurs, tel que les réponses soumises par l'utilisateur, mais également pour pouvoir former les questions. En effet, pour déterminer une question au hasard, le code javascript utilise à deux reprises la fonction `Math.Random()`. Dans un premier temps, on utilise le sélecteur JQuery pour déterminer le nombre de section il y a dans la partie *connaissance* de la page internet (chaque section a son propre type d'information, et donc de question). Pour ce faire, on exécute la fonction `$('selector').length`, ce qui retourne le nombre d'élément *selector*. Une fois la section trouvée par la fonction `math.random()`, on procède de la même manière pour déterminer il y a combien d'élément *question* dans la section (soit le nombre de tonalité). On détermine ensuite au hasard une tonalité, encore une fois avec `Math.random()`. Par la suite, toujours en utilisant JQuery, il est possible d'aller récupérer l'élément de réponse à la question, puis de la comparer à la réponse de l'utilisateur. Si la réponse est bonne, une fonction javascript permet de déterminer le score pour la question, soit deux fois le temps

restant pour répondre à la question. Ainsi, plus la question est répondu rapidement, plus le nombre de points attribué sera grand. Finalement, on utilise JQuery pour cacher ou montrer la section du jeu. En appuyant sur *Prêt?*, on montre la section de jeu. Dans le cas d'une mauvaise réponse et du temps écoulé, la section de jeu est caché. On remplace alors le *dernier score* par le score fait dans cette ronde en allant récupérer le *.text()* de l'élément *.score*.

Le jeu offre deux modes possibles : le mode normal et le mode **normal** et le mode **assisté**. Le mode normal correspond à une simple question, choisit au hasard parmi l'ensemble des données de connaissances, et le joueur écrit avec le clavier sa réponse dans l'encadré. Le mode assisté, quant à lui, donne au joueur un choix trois de réponse. Pour procéder, en utilisant jquery/javascript, on doit mettre dans un bouton, au hasard, la bonne réponse. Dans les deux autres boutons, on met d'autre réponse possible, également choisi au hasard, qui doit être différent des autres choix de réponse (voir le code de *devoir1.js* pour voir la procédure).

Notes : Un bug se retrouve encore dans mon jeu. Lorsqu'une mauvaise réponse est entré, ou qu'on épuise le temps alloué pour répondre à la question, si on fait de nouvelle tentative après avoir réappuyer sur le mode désiré, toutes les nouvelles réponses seront jugé comme fausse alors qu'elle sont bonnes. Le problème semble être dans le choix de la question, c'est-à-dire que la page affiche une question mais cette question ne correspond pas véritablement à celle que le programme à déterminer, et ainsi la réponse écrite ne correspond pas à la réponse auquel le système s'attend. On voit cependant que le score est calculé, donc on a donné la bonne réponse, mais pour une raison que je n'ai pas trouvée, il détecte également que la réponse donnée n'est pas la bonne.