
Quiz Panic

Introduction

Ce TP n'est pas facile, il prendra sans doute deux séances. La méthodologie à adopter est de chercher dans les documentation ou plus généralement sur internet les fonctions que l'on ne connaît pas, car c'est en pratique toujours comme cela que l'on apprend un langage. Ainsi, dans les exercices, je liste les éléments à utiliser, qui n'ont pas forcément d'exemples d'utilisation dans le cours, il faut chercher sur les docs officielles et autres références/tutoriels.

TP

Le but de ce TP est de créer un clone, très rudimentaire et non multijoueur, du jeu : Quiz Panic. Le jeu dans votre version sera un simple quiz où l'on a un temps limité pour répondre à une question en cliquant sur une des quatre réponses possibles.

De plus, ce TP sera à faire à deux ! Le but est de se familiariser avec l'utilisation de git pour travailler à plusieurs.

Exercice I (Mise en place : HTML et CSS)

- 1) Déterminez qui sera appelé Albert et qui sera appelé Berthe dans la suite du TP. Sur son interface web du serveur Gog de l'IUT, Albert crée un nouveau repository, et ajoute Berthe comme contributeur.
- 2) Albert et Berthe clonent le dépôt sur leur machine : ils ont chacun une version en local.
- 3) Albert et Berthe travaillent en parallèle, l'un · e pour créer un fichier html, et l'autre pour créer une feuille de style. Dès que terminé, uploadez votre travail sur le serveur git en utilisant git add, commit puis push. Une interface très simple est suffisante, il faut un block pour contenir la question posée, quatre blocks pour contenir les réponses et un block (qui se déplacera) pour représenter le choix de la réponse. Ajoutez aussi un block pour le compte à rebours.

Les deux exercices à venir seront différents pour Albert et Berthe.

Exercice II (Javascript, Albert)

Albert va s'occuper du timer. Créez un timer qui compte à rebours de 10 à 0 secondes, et devient rouge pour les trois dernières secondes. Lorsque le temps est écoulé, le timer s'arrête et une alerte s'affiche.

Les fonctions à utiliser pour cet exercice sont les suivantes :

- 1) alert() pour faire une alerte,
- 2) setInterval() et setTimeout(), qui permettent de lancer une fonction à intervalle régulier ou à la fin d'un temps donné,
- 3) getElementById pour récupérer un élément de l'arbre HTML
- 4) innerHTML, qui permet de modifier le contenu HTML d'un élément récupéré,
- 5) element.style.color qui peut être modifié pour changer la couleur de "element"...

Exercice III (Javascript, Berthe)

Berthe va s'occuper de la sélection de la réponse. Le but est le suivant : lorsque l'on clique sur une réponse, son fond devient rouge et celui des autres réponses gris clair. De plus, le carré représentant le personnage est déplacé sur la réponse. Bonus : si vous vous sentez à l'aise, faites que le carré se déplace de manière continue. Pour cela, vous devriez utiliser les fonctions suivantes :

- 1) getElementById pour récupérer un élément de l'arbre HTML,
- 2) les propriétés de style d'un élément, modifiables pour changer son apparence,
- 3) l'association d'une fonction "réflexe" à un événement comme un clic de souris, soit à l'aide de "onClick" soit à l'aide de addEventListener,

- 4) revoir le positionnement du block “joueur” pour qu’il se déplace sur une réponse,
- 5) setInterval() et setTimeout() si vous voulez faire une animation...

À la fin de chaque exercice, poussez vos modifications sur le serveur. S’il y a des conflits, faites un pull –rebase et résolvez les conflits, puis repoussez.

Exercice IV (Conclusion, et polissage)

Ajoutez chacun de votre côté des améliorations parmi les suivantes (répartissez-vous le travail!) :

- 1) Un bouton pour démarrer le timer et afficher la question et les réponses,
- 2) la vérification de la réponse à la fin du décompte, en coloriant de différents couleurs les bonnes et mauvaises réponses,
- 3) la question est tirée au hasard (parmi au moins deux questions!)
- 4) un décompte de points,
- 5) améliorez l’interface au niveau du css,
- 6) toute amélioration qui vous vient à l’esprit !

Si vous vous débrouillez comme il faut, la mise en commun se fait le plus souvent facilement. Mais même dans ce cas, il peut y avoir des conflits à résoudre !