

Réalisation d'un site de flashcards (cartes mémoire)

Modalités du projet

1 Organisation

Équipes : Vous devez réaliser ce projet par binômes. **Aucune équipe de plus de deux personnes ne sera acceptée.** Vous pouvez le faire avec un-e étudiant-e d'un autre groupe. Les monômes sont interdits sauf exceptions¹ et ils ne seront autorisés qu'au cas par cas par les enseignants. Un forum est mis en place sur Moodle pour vous aider à trouver un binôme.

Inscription : L'inscription s'effectuera sur Moodle. Les membres d'une même équipe doivent s'inscrire dans le même binôme (identifié par un numéro) sur Moodle. Les numéros n'ont aucune signification, l'ordre de passage des groupes sera fixé plus tard.

Rendu : Vous devrez rendre votre travail dans une archive sur Moodle. La date limite vous sera communiquée ultérieurement.

Si vous travaillez sur vos propres machines, pensez à tester votre site web au Script pour le cas où vous auriez un problème avec votre ordinateur.

Soutenances : L'objectif de ce projet est de réaliser un site complet et d'être capable de présenter votre travail lors d'une soutenance de 20 minutes.

Il est important de travailler **en groupe** de manière équitable : chacun des membres du binôme devra avoir programmé une partie significative du projet. Par exemple, si un(e) étudiant(e) ne code que le HTML et le CSS et son(sa) camarade tout le reste, ce n'est pas équitable. Par ailleurs, chacun des membres du binôme doit pouvoir répondre à des questions sur chaque partie du projet.

2 Sujet

2.1 Principes des cartes mémoires

Une carte mémoire, à l'origine, est une fiche cartonnée comportant une question et la réponse au dos.

Un jeu (paquet) de ces cartes mémoire sert à accélérer et améliorer l'apprentissage et peut être utilisé pour apprendre le vocabulaire d'une langue étrangère, les formules mathématiques, les dates historiques, etc.

L'utilisation la plus classique des cartes classiques consiste à tirer les cartes, et donc les questions, une à une, à essayer d'y répondre, puis à les répartir en fonction de la justesse de la réponse ; une autre succession de tirages intervenant quelque temps après, etc., jusqu'à ce que l'ensemble des réponses soient données sans erreur.

Nous préciserons plus loin, ce qui vous sera vraiment demandé dans le projet.

Pour chaque thème d'apprentissage, ou pour chaque unité d'apprentissage, on utilise un jeu de cartes mémoire séparé (on ne mélange pas les cartes pour apprendre les dates historiques avec les cartes pour apprendre le vocabulaire anglais).

1. Si le nombre d'étudiants est impair, par exemple.

2.2 Ce que vous devez faire

Vous devrez, dans ce projet, programmer une version informatisée de ces cartes mémoires à l'aide des technologies vues dans ce cours : HTML, CSS, PHP et MySQL (et optionnellement Javascript). L'objectif de ce cours étant de vous faire manipuler directement les langages, **l'utilisation d'un framework² est formellement interdite**. Vous avez une certaine liberté pour votre site, mais, dans tous les cas, il devra impérativement comporter :

Une page d'accueil au contenu libre, mais l'ensemble du site (sauf les pages d'administration) doit être accessible depuis cette page (la page d'inscription, les jeux de cartes mémoire si jamais ils ne sont pas affichés sur cette page, la possibilité de se connecter...). Tout ne doit pas forcément être accessible directement depuis cette page, mais indirectement si. Il doit également être possible d'y retourner facilement depuis toutes les pages.

Une page d'inscription elle permettra de créer un compte sur le site, et demandera *au minimum* un identifiant (email, pseudo, ...) et un mot de passe. D'autres champs pourront être ajoutés selon votre envie.

Un formulaire pour se connecter sur le site, demandant une combinaison *identifiant / mot de passe*. En cas d'erreur (mauvaise combinaison identifiant / mot de passe) votre site doit l'indiquer à l'utilisateur, et une fois connecté l'utilisateur doit pouvoir se déconnecter.

Une page de compte où les utilisateurs peuvent modifier leurs données ainsi que supprimer leur compte. D'autres fonctionnalités pourront y être ajoutées (comme une liste des jeux de cartes mémoire utilisés).

Différents niveaux de responsabilités :

Utilisateur L'utilisateur lambda doit pouvoir consulter les jeux de cartes mémoires proposés et commencer ou reprendre un apprentissage sur un de ces jeux. (voir plus loin).

Rédacteur un utilisateur rédacteur a en plus le droit de proposer de nouveaux jeux de cartes et de modifier ceux qu'il a rédigés.

Administrateur Le rôle de l'administrateur est de supprimer des jeux de cartes ou des cartes au contenu inapproprié ou faux. Il peut aussi supprimer les comptes des rédacteurs qui mettent du contenu inapproprié ou répétitivement faux (dans ce dernier cas, ils peuvent devenir simples utilisateurs).

L'administrateur utilisera pour ce faire une page non-visible par les autres utilisateurs.

L'apprentissage avec les cartes mémoire La fonctionnalité minimale est la suivante : un utilisateur doit pouvoir choisir un jeu et le tester en entier, sans forcément que le site se rappelle ce qu'il a fait. Les questions peuvent être des questions ouvertes, des questions style QCM, etc. On peut imaginer, soit que l'utilisateur vérifie lui-même sa réponse soit que l'ordinateur le fasse.

On peut ajouter les fonctionnalités suivantes (pas forcément toutes en même temps) :

- Le site se rappelle quel est le dernier jeu sélectionné par l'utilisateur.
- Le site se rappelle quelles cartes du dernier jeu l'utilisateur doit refaire ou faire (s'ils ne les a pas faites).
- Le rédacteur donne à chaque carte un niveau de difficulté et ce sont les plus simples qui sont proposées d'abord.
- On peut aussi envisager de donner un niveau de difficulté globale à chaque jeu de cartes.
- ...

2. comme Zend, Symfony, Bootstrap ...

Autres fonctionnalités possibles n'ayant pas de rapport avec l'apprentissage

- un (ou deux) bouton(s) permettant de signaler à l'administrateur des contenus inappropriés ou faux.
- Un système de vote augmentant la visibilité des jeux appréciés par les utilisateurs.
- Un système de changement de style avec lequel l'utilisateur peut choisir différents styles d'affichage sur une page de configuration. (Il faudra bien sûr créer plusieurs feuilles CSS pour réaliser ceci).
- Un système de catégories (histoire, langues, ...) pour les jeux. La création de nouvelles catégories pourra être réservée à l'administrateur.
- Un système de scores donné à l'utilisateur pour l'apprentissage d'un jeu donné. On peut imaginer un score calculé comme un pourcentage de réponses justes, ou donner un nombre de points différents selon les cartes, ou encore prendre aussi en ligne de compte la rapidité des réponses. On peut aussi imaginer un best-off des meilleurs scores pour ce jeu, soit anonyme, soit sous réserve de l'accord des utilisateurs concernés.

Objectif : Vous devez implémenter **au moins l'une** des fonctionnalités proposées (ou une autre équivalente, si elle a été discutée avec votre chargé·e de TP au préalable). Vous pouvez bien sûr en programmer plusieurs, mais ça ne doit pas être au détriment du reste.

3 Recommandations

Sur la conduite du projet : Le projet fait appel à des contenus du cours qui n'auront pas tous déjà été vus au moment où cet énoncé est donné. Vous ne pourrez donc pas, a priori, travailler sur tous les aspects du projet dès le début.

Que ceux qui veulent étaler leur effort se rassurent : c'est tout à fait possible. Les pages HTML et CSS peuvent être écrites avant d'y intégrer du PHP, puis la base de données. Le projet est aussi une opportunité de réviser son cours en le mettant en pratique au fur et à mesure que celui-ci avance.

Sur la sauvegarde : **Pensez à faire des sauvegardes régulières** et à toujours avoir une version qui fonctionne sous la main. Parfois, en voulant corriger un bug, on casse autre chose et on est très soulagé d'avoir une ancienne version qui marche depuis laquelle repartir. Pensez également à faire des sauvegardes sur plusieurs supports : sur vos ordinateurs, au SCRIPT, sur une clé USB, sur un git ou svn... en cas de panne de l'un de ces supports, vous aurez toujours les autres en remplacement.

Sur l'objectif du projet : C'est un cours d'informatique, pas de graphisme. Concentrez-vous sur le code (HTML, CSS, PHP, MySQL) de votre site avant son esthétique. Le Javascript ne doit pas remplacer le PHP, vous serez avant tout notés sur la maîtrise des quatre langages du programme du cours.

Sur la sécurité du site : Vous ferez attention aux failles de sécurité : les formulaires devront être protégés contre les injections SQL et les mots de passe ne doivent pas être stockés en clair. Le cours abordera ces sujets.

Sur la lisibilité du code : Votre code doit être clair et contenir des commentaires qui permettent de le comprendre. Il doit être indenté et respecter les standards W3C.

Sur la planification : Particulièrement pour la partie en lien avec la base de données, réfléchissez aux extensions que vous souhaitez mettre en place **avant de commencer à coder**. Si vous ne vous y prenez pas en amont, certaines d'entre elles vous demanderont d'adapter lourdement votre

base de données, ce qui vous fera perdre beaucoup de temps... Pour vos mises en page HTML et CSS, faites un schéma du résultat désiré et identifiez ses sous-parties avant d'essayer de le coder.

Rapport On ne vous demande pas de rapport, mais préparez pour la soutenance un schéma de la base de données, il peut être manuscrit, il ne vous est pas demandé de faire un schéma UML, donnez juste les tables, leurs colonnes et si une colonne d'une table correspond à une colonne d'une autre, indiquez-le. Si vous aviez prévu un schéma pour plus d'extensions que réalisées, indiquez les parties qui ne servent pas, nous tiendrons compte de cette partie du travail.

4 Questions

Si vous avez des questions ou des suggestions, parlez-en à votre chargé(e) de TP. Un forum sera également mis en place sur Moodle.