

Cycle 1.1 Etude de cas pratique

D12 Programmation/Game design

D11.1 Algo

D11.2 web

Type d'épreuve : Réalisation Pratique

Et Présentation Orale

Durée : 5 jours Et oral de 10 min

Session : Juin 2022

BAREME DE NOTATION

Dossier 1 - Web	100 Points
Dossier 2 - Programmation et Algo	125 Points
Dossier 3- Game Design	70 Points
Présentation et orthographe	5 Points
Total	300 Points

Vous aurez à rendre pour le 17/06/2022 18h :

- Git du projet, Documentation et l'archive des ressources si besoin sur l'adresse suivante :
exam_cycle1_sxb@ludus-academie.com
- Vous aurez un oral durant les semaines suivantes pour présenter vos réalisations.

LISTE DES ANNEXES

Annexe 1	Web	Page 9
Annexe 2	Algo et Développement	Page 10

Contexte de l'étude

La société KiffMySocialGame est implantée depuis un moment en Alsace et a pour but de diffuser, promouvoir et partager autour du jeu. Ses clients payent des services afin d'avoir accès à des jeux de société ou des informations autour de ceux-ci.

Vous avez été engagé pour répondre à plusieurs missions de cette société. Les tâches qui vous seront attribuées sont indépendantes. KiffMySocialGame vous souhaite bonne chance et a hâte de voir vos travaux.

Dossier 1 - WEB

La société KiffMySocialGame spécialisée dans les jeux de société et leur appréciation souhaite se débarrasser des fiches papier de ses adhérents.

Elle fait donc appel à vous et vos compétences fraîchement acquises en web statique.

Votre mission est de leur créer un outil web afin que les utilisateurs puissent s'identifier et donner des avis sur un ensemble de jeux de société.

NB : Vous ne gérez pas la mémorisation des données pour une reconnexion de l'utilisateur. Seules nous intéressent la génération des pages et la transmission des informations saisies en page 1(saisie_avis.html) vers la page 2(VotreAvis.html).

Description des attentes :

Le répertoire KiffMySocialGame avec une arborescence permettant de séparer les fichiers (html, css, images, js....).

2 pages à créer : une première de saisie et une seconde de feedback.

Page saisie_avis.html (65 pts) :

Le contenu des pages sera généré en JS DOM (JavaScript Data Object Model) ; cf. Annexe1 Web Partie DOM.

Cette page contient un formulaire ; cf. Annexe1 Web Partie formulaire, liens 1 et 2.

NB : le formulaire est généré avec du JS DOM au chargement de la page.

Le formulaire est organisé en plusieurs zones :

L'utilisateur y saisira ses informations personnelles (cf. Zone identité plus bas).

L'utilisateur pourra définir ses préférences de jeux (cf. Zone préférences plus bas).

L'utilisateur pourra donner son avis ((cf. Zone avis plus bas) :

1. Nom du jeu,
2. Commentaire (200 caractères max).
3. Note sur 5.

Les zones et leurs informations à renseigner :

Zone identité :

Genre (bouton radio),

Nom,

Prénom,

Email(2 champs dont la saisie devra être identique),

Date de naissance,

Adresse postale (Zone de texte, optionnelle),

Téléphone, optionnel.

Zone Préférences de jeux :

Checkboxes de catégories de jeux favoris,
Listes de choix (multiple) de vos jeux préférés.

Zone avis :

Liste de choix (unique) du jeu auquel l'utilisateur donne son avis (ce jeu n'est pas obligatoirement un des jeux préférés).
Commentaire (200 caractères max.),
Note (5 maximum).

Zone de validation du formulaire :

Bouton Effacer (qui va vider le formulaire),
Bouton Valider (soumission du formulaire à la page VotreAvis.html)

Indications :

- Les zones du formulaire sont matérialisées par des fieldset.
- Les champs de saisie sont précédés de labels et contiennent des placeholders.
- Les radiogroups, listes de valeurs et checkboxes sont générés dynamiquement à partir de sources de données de type tableau (déclaration de tableaux de catégories, de jeux ; puis itérations sur ces tableaux afin de créer vos éléments html).
- Il faudra au moins une catégorie de jeu cochée (pas de catégorie par défaut) pour pouvoir valider le formulaire.
- Il faudra au moins un jeu sélectionné dans la liste de choix Jeux pour pouvoir valider le formulaire.
- Les 3 champs de l'avis sont obligatoires.

Le style des champs :

Les champs obligatoires sont bordurés de rouge (bien choisir le sélecteur).
La couleur de police des champs email passe au vert quand la saisie des 2 champs est identique.

Validation du formulaire :

Si la saisie est incorrecte, rester sur la page et indiquer les erreurs à l'utilisateur (pas de console.log).
Sinon, afficher la page de feedback VotreAvis.

VotreAvis.html (35 pts) :

Cette page dont le contenu est également généré en JS DOM affiche un récapitulatif de la saisie de l'utilisateur (A vous d'en définir le style). Cf. annexes Web Formulaires 2,3,4,5.
NB : il ne s'agit plus de zones de saisie dans VotreAvis.html mais de simples conteneurs.

Zone Identité :

En plus de la date de naissance saisie, afficher l'âge calculé.

Zone préférences :

Si vous avez stocké des images correspondant aux catégories, les afficher, sinon afficher le nom de la catégorie ; idem pour les jaquettes des jeux.

NB : s'il vous manque du temps, contentez-vous de lister les valeurs textuelles de catégorie et de nom des jeux choisis.

Zone Avis :

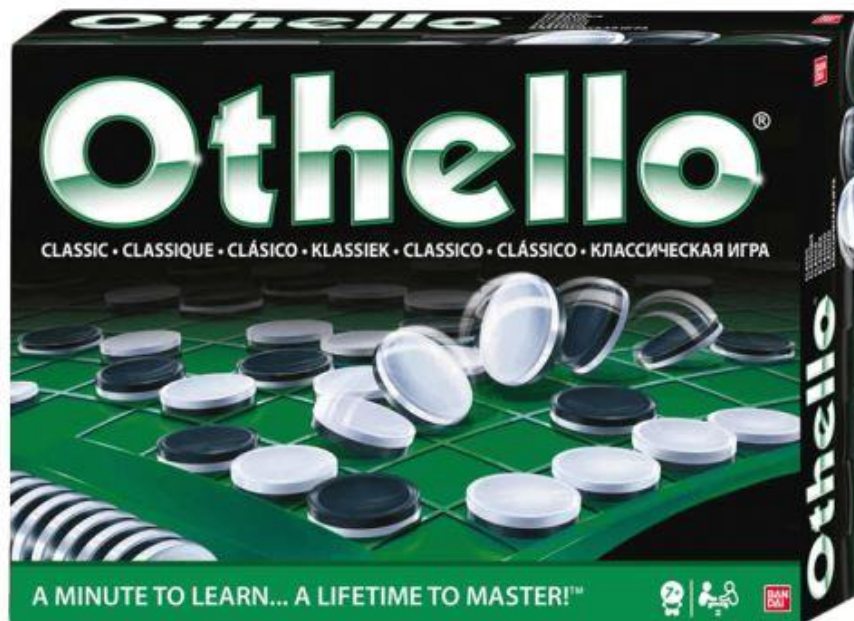
Afficher les avis et notes des jeux sous forme de tableau stylisé.

Les commentaires ainsi qu'une bonne indentation du code sont indispensables.

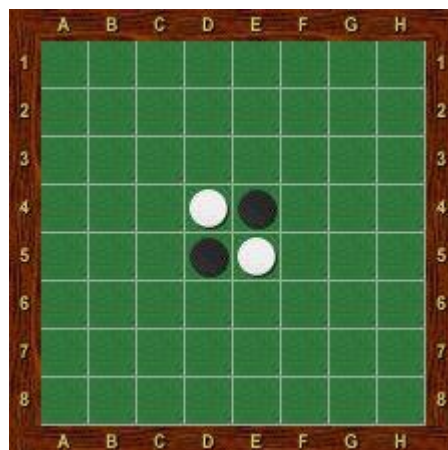
Dossier 2 – Algorithmie/Programmation

La société KiffMySocialGame fête ses 10 ans et pour l'occasion, elle a l'intention de sortir un jeu d'époque (daté de 1880 puis plus tard remodernisé dans les années 1970). Ce jeu s'intitule : « Reversi » ou également « Othello ».

On vous charge de vous occuper de la partie Algo et programmation C de ce dernier.



Ce jeu se joue sur un plateau possédant 64 cases. Il se base sur des pions réversibles (blanc d'un côté et noir de l'autre). L'image ci-dessous montre une configuration de départ (prêt à jouer).



Il y a assez peu de règles (voir le lien suivant : <https://bit.ly/3a6bWxU>).

PARTIE 1 : Algorithmie

Question 1 : 15 pts

Veillez créer un algorithme reprenant la logique complète du jeu. Autrement dit, on vous demande d'écrire l'ordre des procédures => en partant du début de jeu jusqu'à la fin. Ceci inclus surtout la boucle principale du jeu.
(Pour cette question vous ne devez pas faire l'algorithme de chaque fonction, uniquement montrer l'ordre d'appel de celles-ci).

Question 2 : 30 pts

Maintenant que vous possédez la logique principale de traitement, veuillez faire l'algorithme pour chaque fonction distincte. (Plusieurs fonctions sont nécessaires pour définir les différentes tâches du jeu, on vous demande de les détailler sous forme algorithmique).

PARTIE 2 : Développement

Question 1 : 10 pts

Affichez le titre, un menu ainsi que le plateau de jeu en mode console.
Le menu doit proposer les options suivantes :

1. Affronter un ami
2. Affronter l'ordinateur
3. Quitter

Question 2 : 20 pts

Créer une boucle de jeu principale, contenant une fonction permettant de placer des pions et calculer quel(s) pion(s) adverse(s) il faut retourner en cas de capture(s).

Question 3 : 10 pts

Utilisez les concepts d'allocation dynamique.

Ces concepts vous permettent d'optimiser votre jeu et d'allouer de la mémoire uniquement lorsque cela est nécessaire. Veuillez utiliser les primitives malloc()/calloc(), realloc() et free() pour les variables temporaires. Par exemple : un pion.

Question 4 : 10 pts

Créer une fonction pour détecter si le joueur doit passer son tour ou non.
(voir règles de jeu en annexe)

Question 5 : 10 pts

Créer une fonction permettant de calculer les points en fin de partie et déterminer quel joueur a gagné.

Question 6 : 15 pts

Permettre au(x) joueur(s) (dépend du mode de jeu) de choisir une couleur personnalisée (autre que noir ou blanc) et d'insérer son nom au lieu d'avoir un « Joueur 1 » ou « Joueur 2 ».

Question 8 : 5 pts

Trouvez une méthode pour jouer de la musique au menu et durant une partie (de préférence 8bit).

Dossier 3 - Game Design

Pour terminer, la société KiffMySocialGame aimerait que vous proposiez de nouveaux concepts de jeu pour favoriser la création ludique alsacienne.

Question 1 : 40 pts

Proposez un concept de jeu de société original pour alimenter la bibliothèque de jeu de KiffMySocialGame. Vous êtes en concurrence avec beaucoup d'autres concepts et votre commanditaire aimerait que vous proposiez un projet qui se démarque du reste.

Vous avez carte blanche mais l'entreprise aimerait que vous lui fournissiez les documents nécessaires pour comprendre le projet dans son intégralité :

- Dos de boîte
- Règles du jeu
- Game concept

Question 2 : 30 pts

Afin de valider et de promouvoir l'univers de votre jeu de société. KiffMySocial aimerait que vous proposiez un **game concept d'un jeu vidéo** reprenant les éléments d'univers (monde, personnages et/ou histoire) de votre jeu de société sans en reprendre le gameplay. L'équipe sera particulièrement attentive à votre analyse de la cible du jeu ainsi que vos intentions qui doivent correspondre au projet et vos buts personnels.

Rappelez-vous que les éléments les plus attrayants de votre concept de jeu de société doivent se trouver dans votre concept de jeu vidéo.

Annexe 1 – Web

Listes de liens de documentation sur le DOM et les formulaires

Partie 1 : Le DOM

Le DOM est un standard qui permet d'accéder, modifier, ajouter ou supprimer des éléments HTML en JS.

Tous les éléments sont des objets.

On peut setter et getter les attributs.

On déclenche des actions grâce aux méthodes (ajouter des événements à un élément)

Les méthodes permettant de récupérer les éléments du document sont à votre disposition ; Bien les choisir.

Il est possible de récupérer un élément par :

- Son identifiant,
- Sa balise,
- Le nom de sa classe,
- Son sélecteur CSS.

Après avoir ajouté un élément au DOM, il est possible de lui setter ses attributs.

Pour qu'un élément créé soit affiché, ne pas oublier de l'ajouter au body du document, au bon endroit par rapport aux autres éléments.

<https://fr.javascript.info/dom-nodes>

<https://openclassrooms.com/fr/courses/5543061-ecrivez-du-javascript-pour-le-web/5543068-comprenez-ce-quest-le-dom>

https://fr.wikipedia.org/wiki/Document_Object_Model

https://javascript.developpez.com/tutoriels/introduction/?page=page_5

Partie 2 : Les formulaires

Le formulaire est l'objet HTML permettant à l'utilisateur de saisir des données et de les poster (les soumettre et les transmettre via l'URL (method='GET' action='VotreAvis.html') dans notre cas.

Ne connaissant pas encore les traitements côté serveur, la page des avis que vous développez sera alimentée par les données du formulaire soumises et stockées dans la chaîne `documentation.location.search` que vous traitez en JS (travail sur les chaînes de caractères).

1. https://www.w3schools.com/html/html_forms.asp
2. https://developer.mozilla.org/fr/docs/Learn/Forms/Sending_and_retrieving_form_data
3. <https://developer.mozilla.org/en-US/docs/Web/API/Location/search>
4. <https://www.sitepoint.com/get-url-parameters-with-javascript/>
5. https://www.w3schools.com/jsref/prop_loc_search.asp

Annexe 2 – Algorithmie/Développement

Partie 1 : Algorithmie

Règles du jeu Othello :

<https://bit.ly/3H9I3KH>

Boucle de jeu (concept de macro-boucle) :

<https://www.savoirdanslavie.com/gameplay-loop-gaming/>

<http://mathsoup.xyz/mathsoup.xyz/content/ProjetJV/game-design/boucle-de-jeu.html>

Partie 2 : Développement

L'allocation dynamique :

[https://fr.wikibooks.org/wiki/Programmation_C/Gestion_de_la_mémoire](https://fr.wikibooks.org/wiki/Programmation_C/Gestion_de_la_m%C3%A9moire)

Savoir organiser son code en plusieurs fonctions et/ou fichiers :

https://zestedesavoir.com/tutoriels/755/le-langage-c-1/1042_les-bases-du-langage-c/4297_decouper-son-projet/

+N'hésitez pas à utiliser toutes les notions vues en cours (théorie + exercices TP).