



Cours FrontEnd (HTML,
CSS, JS)

TP Évaluation

Résumé/ Objectif

L'objectif est de réaliser une simple application web exploitant l'API de jeux vidéos **GiantBomb**.

Cette application nous permettra en premier lieu de faire une recherche parmi les milliers de jeux listés par l'API, d'afficher une liste des résultats sous forme d'aperçus, et de consulter la fiche des différents jeux affichés.

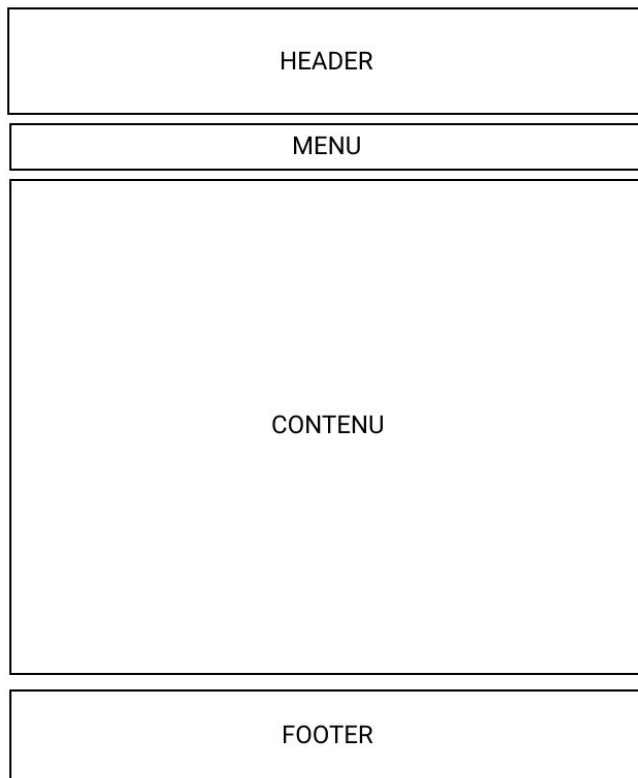
Elle permettra également d'ajouter les différents jeux consultés à une liste de favoris.

Structure et fonctionnalités de base

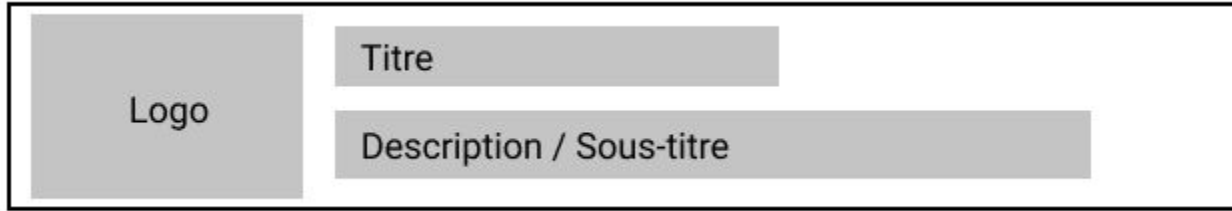
L'application attendue est une SPA (Single Page Application), dont le layout global sera constitué :

- D'un header, qui affichera le logo du site, son titre, ainsi qu'une courte description.
- D'un menu, affichant :
Un champ de recherche
Un bouton "Favoris", permettant de consulter la liste des jeux ajoutés en favoris.
- D'une section principale affichant soit les différents jeux/résultats (plateforme ou recherche), soit la fiche d'un jeu sélectionné
- D'un footer, affichant quelques liens annexes ainsi que les détails de l'auteur de l'application.

Structure générale

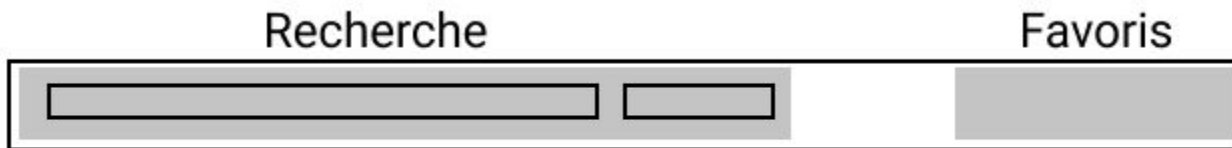


Header



- Le logo peut-être une image de votre choix, ou un des logos fournis avec cet énoncé.
- Le titre et la description sont au choix.
- La description doit représenter succinctement mais efficacement l'objectif de l'application, dans un soucis de référencement par exemple.

Menu



Le menu, dans sa version de base, contient d'un côté un champ de recherche, constitué d'un input text, et d'un bouton permettant de valider la recherche.

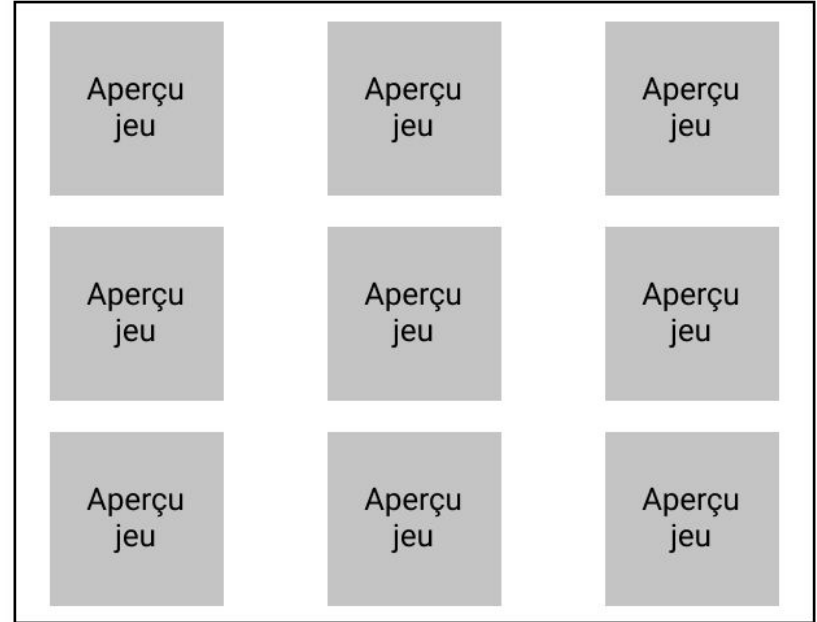
De l'autre côté, un bouton permettant d'afficher la liste des jeux mis en favoris.

Contenu : Résultats de recherche et liste des favoris

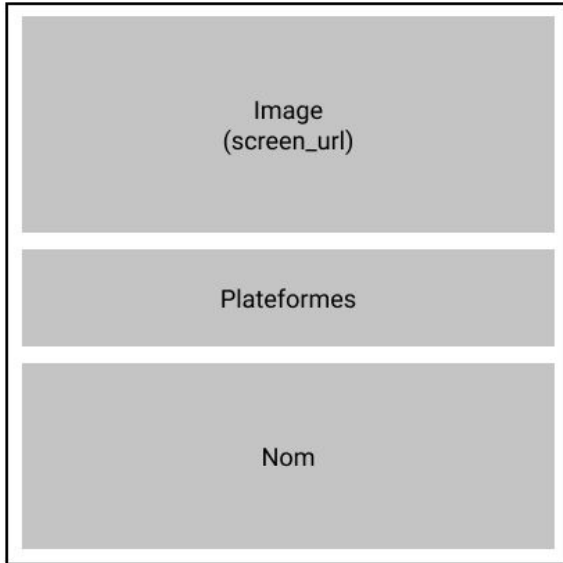
La partie principale affiche, suite à une recherche ou le clic sur le bouton favoris, la liste des jeux correspondants.

Chaque jeu est présenté via un bloc “aperçu” au style similaire, et ces blocs sont répartis équitablement.

Le nombre de blocs par ligne est libre, et s’adapte idéalement selon la largeur de l’écran.



Bloc aperçu d'un jeu



Le bloc aperçu d'un jeu est constitué des éléments suivants :

- Une image représentant le jeu. (Format “screen_url” dans les données de l’API)
- La liste des plateformes sur lesquels le jeu est accessible, affichées dans des blocs séparés visuellement, et contenant l’abréviation de la plateforme.
 - Si le nombre de plateformes est supérieur à 4, on affiche dans un 5ème bloc le nombre restant de plateformes, au format “+N” ou N est le nombre de plateformes non affichées.
- Le nom complet du jeu.

Au clic sur ce bloc, on affiche la fiche du jeu en question.

Contenu : Fiche jeu

La fiche d'un jeu contient les informations suivantes :

- Le nom complet
- L'image représentant le jeu (format "small_url" dans l'API)
- La liste des plateformes (mêmes consignes que sur l'aperçu)
- La date de sortie du jeu (entête "Sortie", OU l'année annoncée de sortie (entête "Sortie prévue"), l'entête et la date étant sur deux lignes.
- La description courte du jeu.
- La description longue du jeu.
- Un bouton "Ajouter aux favoris" si ce jeu n'y est pas déjà, sinon, "Retirer des favoris".

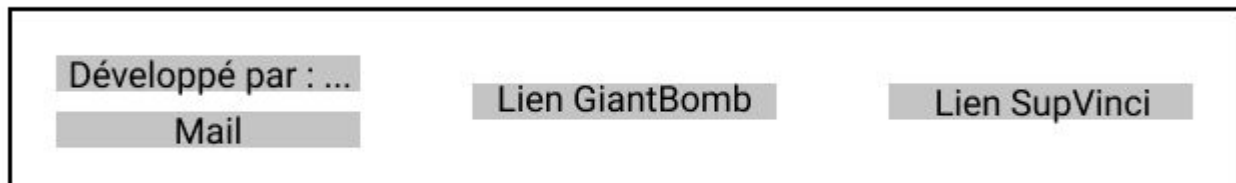
Nom	Image (small_url)
Plateformes	Date
Description courte	
Description longue	

Ajout/Retrait des favoris

Lors du click sur le bouton Ajouter ou Retirer des favoris :

- S'il s'agit d'un retrait, demander confirmation.
- Une fois l'ajout/retrait fait, une popup s'affiche indiquant que l'action a été effectuée.
- Cette popup devra être créée sur mesure en HTML/CSS (Pas via alert()), et devra suivre la charte graphique du reste de l'application.
- Idéalement, mais pas obligatoirement, elle sera développée sous forme de composant/module générique.

Footer



Le footer contiendra les informations suivantes :

- Dans un premier bloc sur deux lignes, votre nom ainsi que votre mail @supdevinci. (Le mail n'a pas besoin d'être clickable)
- Un lien vers le site GiantBomb
- Un lien vers le site Sup de Vinci

Les deux liens devront s'ouvrir dans un nouvel onglet.

Divers

- Lors du chargement des pages, un loader animé est attendu (CSS uniquement, pas de JS). Son aspect est libre.
- Les différents boutons de l'application devront réagir au hover, avec un effet de transition, même léger.
- Autant que possible, tenter d'obtenir un design responsive, même si ce ne sera pas un facteur important de l'évaluation, étant donné le peu de temps que nous avons pu consacrer au sujet.

Attentes et évaluations

Attentes techniques

- Le design est libre et ne comptera pas dans l'évaluation finale, cependant un effort minimal de stylisation est attendu.
- La liste des favoris devra être stockée localement dans le navigateur via localStorage.
- Un découpage de l'application est attendu, même si minime, sous forme de modules, au moins en terme de découpage en couches pour le DAO et les Models.
- Pas de préférence de programmation procédurale/orientée objet, mais utilisez au moins des classes pour le/les modèles représentant les types de données manipulées.
- Le projet devra être historisé avec git et hébergé sur un dépôt distant (bitbucket, github, gitlab, etc.). Le cours ne portant pas sur git en soi l'utilisation qui en est faite ne sera pas évaluée, mais je pourrais vous faire remonter des conseils/points de surveillance si besoin.

Évaluation

L'évaluation ne suivra pas un barème fixe et prendra en compte le caractère “condensé” du cours que vous avez eu, ainsi que vos différences d'expériences techniques.

Je porterais notamment attention :

- au respect des conventions “classiques” de développement.
- au nommage des variables, fonctions, classes, etc.
- à la lisibilité du code.
- Côté HTML, à l'utilisation de balises sémantiques appropriées.
- Au respect du cahier des charges (une fonctionnalité non terminée vaut mieux qu'une fonctionnalité qui ne réponds pas à la demande)

Fonctionnalités “Bonus”

Si le cahier des charges de base ne vous donne pas suffisamment de fil à retordre, où si j'ai sous-estimé la charge du TP, je vous propose ici plusieurs fonctionnalités “Bonus” qui compléteront le comportement de l'application.

Il n'y a pas d'ordre de préférence si vous souhaitez en ajouter une, ou plusieurs.

Bonus: Bouton retour sur la fiche jeu

- Ajouter un bouton “retour” sur la fiche jeu, qui nous renverra vers la page de recherche, ou de favoris, qui nous a amené vers cette fiche.

Bonus: Recherche par plateforme

- Ajouter dans le menu un menu déroulant listant toutes les plateformes que propose l'API.
- Lors du clic sur une de ces plateformes, afficher la liste des jeux disponibles sur cette plateforme.
- ET/OU, permettre d'afficher la liste des jeux disponibles sur une plateforme dès que l'on clic sur un des encarts plateforme affiché sur l'aperçu d'un jeu ou sur sa fiche.

Bonus: Pagination

- Les résultats de l'API lorsque l'on fait une recherche de jeux sont paginés, et par défaut limités à 100 résultats.
- Ajouter en bas de chaque page de résultat :
 - Un bouton permettant d'accéder à la page précédente (si applicable)
 - Un bouton permettant d'accéder à la page suivante (si applicable)
 - Un bouton permettant d'accéder à la première page (si plus d'une page)
 - Un bouton permettant d'accéder à la dernière page (si plus d'une page)

Bonus de bonus: Pagination, encore

- Si vous mettez en place la pagination, vous avez également la possibilité de la compléter avec deux options :
- Entre les boutons précédent/suivant, un menu déroulant permettant de sélectionner directement un numéro de page parmi les résultats.
- Proposer également un menu déroulant permettant de paramétrer le nombre de résultat par page.

Bonus: tri

- Ajouter dans les pages de résultat la possibilité de changer le tri d'origine.
- Il faudra permettre de sélectionner parmi au moins deux propriété (par exemple nom, date de sortie)
- Et il faudra permettre de sélectionner l'ordre de tri (ascendant ou descendant)

Bonus: filtre par date

- Permettre de filtrer les résultats d'une recherche par date, en indiquant une année de départ (incluse) et une année de fin (incluse)

Bonus: Compléments d'informations sur fiche jeu

En plus des informations présentes de base sur la fiche d'un jeu, ajouter :

- Le ou les développeurs ayant travaillés sur le jeu
- La franchise du jeu si applicable
- Le nombre ou la liste (avec noms) de DLCs disponibles pour ce jeu.

Bonus: Listes multiples

Changer le système de favoris par un système de listes personnalisables :

- Le bouton favoris devient “Mes listes”
- La page liste contiendra un menu supplémentaire affichant les différentes listes, à gauche ou à droite de la partie affichant le contenu de la liste sélectionnée.
- Une première liste par défaut sera présente : “Favoris”
- Un bouton en bas de la liste de listes permettra de créer une nouvelle liste, qui devra être nommée, le nom devant être unique.
- Chaque liste exceptée la liste “Favoris” doit pouvoir être supprimée. Dans ce cas, demander une confirmation.
- Le bouton “Ajouter aux favoris” sur les fiches jeu deviendra “Ajouter à une liste”. Il affichera une popup permettant de sélectionner une liste.
- Si le jeu est présent dans une liste, le bouton “Ajouter à une liste” deviendra “Retirer de la liste [Nom de la liste]”.

Bonus: Cache

- Mettre en place un système de cache pour les jeux dont la fiche a déjà été consulté.
- Le cache n'aura pas besoin d'être persistant.