

Adaptation des interfaces

Rendu n°1

I Le groupe

Vincent Forquet forquet@polytech.unice.fr Technologie traitée : Foundation

Yann Guidez guidez@polytech.unice.fr Technologie traitée : Bootstrap

Shafiq Daniel Bin Sabari bs200009@etu.unice.fr Technologie traitée : Xamarin

Christophe Cappé christophe.cappe@etu.unice.fr Technologie traitée : Angular 2

II L'application

Notre application serait un site web de e-commerce basé sur les sites suivants : <http://store.steampowered.com/?l=french> <https://www.gog.com/> (sites très similaires, e-commerce pour les jeux vidéos avec beaucoup de visuels) Ces sites réussissent très bien en version desktop, mais mal sur les petits écrans. Le site web de Steam a même certains éléments qui ne passent pas du tout à l'échelle et restent en version desktop. Ces sites présentent par ailleurs beaucoup d'éléments à traiter, des listes, des tableaux, des carrousels, etc., d'où l'intérêt de les traiter.

Notre application aurait donc pour fonctionnalités :

- La consultation d'un catalogue d'articles, où chaque article posséderait sa propre fiche détaillée.
- L'achat d'un article (celui-ci serait simulé)
- La possibilité de créer un compte utilisateur permettant de répertorier les articles achetés
- La mise en place d'offres promotionnelles selon le jour/ la semaine, etc...

III Liens entre l'application et les adaptations

Résumé

Deux technologies responsive ont été choisies : Bootstrap et Foundation. L'objectif de ces deux technologies est d'adapter l'interface au contexte de visualisation, et plus particulièrement à la taille de l'écran et au mode d'interaction (tactile). Il est possible d'aller un peu plus loin en proposant des options de colorisation adaptées au contexte lumineux et/ou à l'utilisateur (daltonisme etc.).

Concernant l'utilisation des technologies orientées Web components, nous avons choisi Angular 2. Cette technologie nous permettra également d'adapter notre application selon les dispositifs, tout en proposant une structure réutilisable de composants dont l'affichage change selon le contexte d'utilisation.

Pour le cross platform, la technologie choisi est Xamarin. Le but est de proposer une application mobile native avec des interfaces adaptées pour chaque platform (Android et iOS) et des services qui sont disponibles pour mobile (ex: Push Notification).

Foundation

Utiliser Foundation pour réussir à rendre un site de e-commerce de jeux vidéos complètement responsive. Ces sites implémentent peu de comportements complexes, aussi un framework comme Foundation a de bonnes chances de bien fonctionner. Le défi avec Foundation sera de gérer les visuels et leur taille, car certaines captures d'écran et certaines jaquettes passent mal à l'échelle. Garder une page légère en limitant l'utilisation de mécaniques comme le cacher/montrer en fonction de la taille de l'écran peut aussi se montrer difficile lors de la création d'un site responsive avec ce genre de frameworks. Avec Foundation, il s'agit d'adapter le site web au contexte de visualisation : smartphone, tablette, laptop, desktop, et peut-être d'ajouter une option de contraste/luminosité/couleur pour gérer les conditions de lumière et/ou les pathologies type daltonisme etc.

Xamarin

Cette technologie permet de développer une application cross-platform, donc avec une seul base de code, nous pouvons créer une application compatible avec Android et iOS par exemple. Xamarin.Forms est une nouvelle fonctionnalités dans cette environnement de développement, qui génère les composants natives pour chaque platform. Donc, nous aurons une interface native pour chaque platform. Les composants de l'affichage sont gérée avec les éléments comme Pages et Views, qui peut être adaptée pour différent tailles d'écrans.

Puisque le développement sur Xamarin est pour le mobile, donc l'adaptation visée est pour les différents platforms mobile, et de proposer une application avec des interfaces natives pour chaque platform. Cela présente une avantage par rapport à la solution web qui donne la même interface pour tous l'utilisateur, en adaptant seulement par rapport à la taille d'écran. Une autre fonctionnalité qui est intéressant pour l'application mobile est de pouvoir recevoir des notifications sans ouvrir l'application. Pour notre application de e-commerce pour jeux vidéos, cela représente un intérêt pour l'utilisateur afin qu'il ne rate pas les offres et promotions qui ne dure pas longtemps.

Exemple de scénario : Bob est un hardcore gamer avec un PC équipé de dernière version de carte graphique. Bob est parti en vacance pour 1 semaine. Un jour, pendant les vacances, il a reçu une notification disant qu'il y a une réduction de 80% pour un jeu dans sa liste de souhaits, et l'offre dure juste pour 24 heures. Il n'a pas son PC avec lui, mais grâce à l'application mobile, il peut faire l'achat et commencer le téléchargement à distance. Quand il rentre chez lui, le jeu est déjà téléchargé sur son PC, et il peut jouer toute de suite.

Angular 2

Cette technologie orientée Web components permettrait de rendre l'application très modulaire (par principe des Web components). En effet, chaque fonctionnalité pourrait être traitée à part, sans qu'elle n'empiète sur une autre : par exemple, il y aurait un module de paiement, un module pour afficher la liste des articles, etc...

Cela permettrait également une grande réutilisabilité de l'application : le module de paiement pourrait par exemple être utilisé pour une autre application sans que l'on ait à le modifier en profondeur.

L'adaptation visée concernerait donc les dispositifs : il faudrait pouvoir réutiliser tous les composants (consultation de catalogue, de connexion au site, d'achat d'articles, et d'affichage d'offres promotionnelles) selon différentes tailles d'écrans sans que l'on ait à toucher leur comportement métier. Angular2 permet très bien de faire ce genre de choses grâce aux directives qui permettent de changer l'affichage d'un composant selon le contexte dans lequel il se trouve. Pour un même composant "Article" par exemple, si on le trouve dans une liste (un catalogue) il faudrait afficher une image ainsi que le titre, mais lorsqu'on souhaite consulter sa fiche, il faudrait également afficher une description. De plus, si la taille de l'écran diminue, alors l'affichage de l'article devrait changer : on pourrait par exemple afficher sa description d'une autre façon pour empêcher qu'elle ne prenne toute la place de l'écran, ceci rendant la navigation peu agréable.

Le défi de cette technologie sera donc de bien cerner le concept de Web component et de réaliser des composants modulaires et réutilisables.

Bootstrap

Cette technologie est utile à la création de sites et d'applications webs. C'est un outil qui contient des codes HTML et CSS, des formulaires, des boutons, etc... ainsi que des extensions JavaScript. Bootstrap utilise une grille afin de placer les différents composants sur la page Web, cette disposition est clairement adaptée au type d'application que nous voulons développer, étant donné que celle-ci va avoir besoin d'afficher des listes d'articles (jeux vidéo). Bootstrap permet aussi d'adapter l'affichage en fonction de la taille de l'écran sur lequel est visualisé le site Web, cela va donc nous permettre d'avoir un affichage différents en fonction de l'appareil utilisé (desktop, tablette, smartphone, ...)

Le contexte visé par l'adaptation avec Bootstrap est celui des dispositifs, le responsive design de cette technologie va nous donner la possibilité d'adapter les différentes pages du site desktop (liste des articles, page de connexion, page d'achat d'article, etc...) à des dispositifs mobiles.