Introduction à la programmation Web mobile

Basé sur un cours de L. Médini

Ressources restreintes — Matériel

Capacité de calcul

→ Limiter la quantité de calculs côté client

- Éviter les scripts trop gourmands
- Privilégier les traitements côté serveur

Mémoire

→ Limiter (éviter l'inflation de) la taille des objets manipulés par les scripts

Ressources restreintes — Réseau

Bande passante limitée (et payante)

Ne charger que les ressources nécessaires

- Ne pas envoyer au client ce qui ne sera pas affiché
- Réutiliser les ressources (potentiellement) déjà présentes :
  - Bibliothèques de code : CDN
  - Données statiques : cache du navigateur (cf. CM3 M1IF03)
  - Données utilisateur : Web Storage
- Charger "intelligemment" les ressources
  - En asynchrone (pour ne pas bloquer l'interaction avec la page)
  - de manière "paresseuse" (lazy) : quand l'utilisateur en a besoin (attention au temps de chargement)

Aspects contextuels

Utilisateurs "on the go”

- Tâche courante ≠ tâche de navigation
- Faciliter la tâche courante
- Adapter le comportement de l’application

Identifier le contexte

- Aspects techniques : appareil, réseau, localisation
- Préférences utilisateur : langue, profil
- Données métier : scénarios, modèles de tâches, d'activités

Aspects contextuels

Utilisateurs "on the go”

- Tâche courante ≠ tâche de navigation
- Faciliter la tâche courante
- Adapter le comportement de l’application

- Media queries (responsive design, limitation de la quantité d'information affichée)
- Device APIs (localisation, lumière ambiante, batterie, micro et caméra, orientation de l'écran, orientation de l'appareil, accéléromètre et boussole)
- Bibliothèques JS (Modernizr, Bootstrap...)
- Services externes (géolocalisation, profiling)

Mobile first

Créer des applications Web “mobile-friendly" :

- Prioriser les objectifs
- Concevoir l'application en fonction des services principaux
- L’étendre pour arriver à l'application "non mobile"
  - Déclenchement de code conditionnel
  - Utilisation des media queries de manière ascendante
  - Ne charger que les contenus nécessaires

Touch Events

Historiquement lié aux mobile

Pointer Events https://developer.mozilla.org/en-US/docs/Web/API/Pointer_events

Permet l'unification des événements de pointage

- Souris
- Stylet
- Touch

Implémenté par toutes les dernières versions des navigateurs

Pointer Events

- pointerId - a unique identifier for the pointer causing the event.
- width - the width (magnitude on the X axis), in CSS pixels, of the contact geometry of the pointer.
- height - the height (magnitude on the Y axis), in CSS pixels, of the contact geometry of the pointer.
- pressure - the normalized pressure of the pointer input in the range of 0 to 1, where 0 and 1 represent the minimum and maximum pressure the hardware is capable of detecting, respectively.
- tiltX - the plane angle (in degrees, in the range of -90 to 90) between the Y-Z plane and the plane containing both the transducer (e.g. pen stylus) axis and the Y axis.
- tiltY - the plane angle (in degrees, in the range of -90 to 90) between the X-Z plane and the plane containing both the transducer (e.g. pen stylus) axis and the X axis.
- pointerType - indicates the device type that caused the event (mouse, pen, touch, etc.)
- isPrimary - indicates if the pointer represents the primary pointer of this pointer type.

Pour aller plus loin

Du Web dans d'autres types de devices

- Microcontrôleurs (Arduino, RaspBerry Pi, Gallileo...)
- Multi-dispositifs
- Objets de tous les jours (éléctroménager, véhicule...)

Avec d'autres types d'architectures

- Protocoles applicatifs (CoAP)
- Communication orientée-messages (WebSockets, REST+Notify)
- Architectures pair-à-pair (WebRTC)