

Module INFO-1

TP 01 - Jeu du Morpion

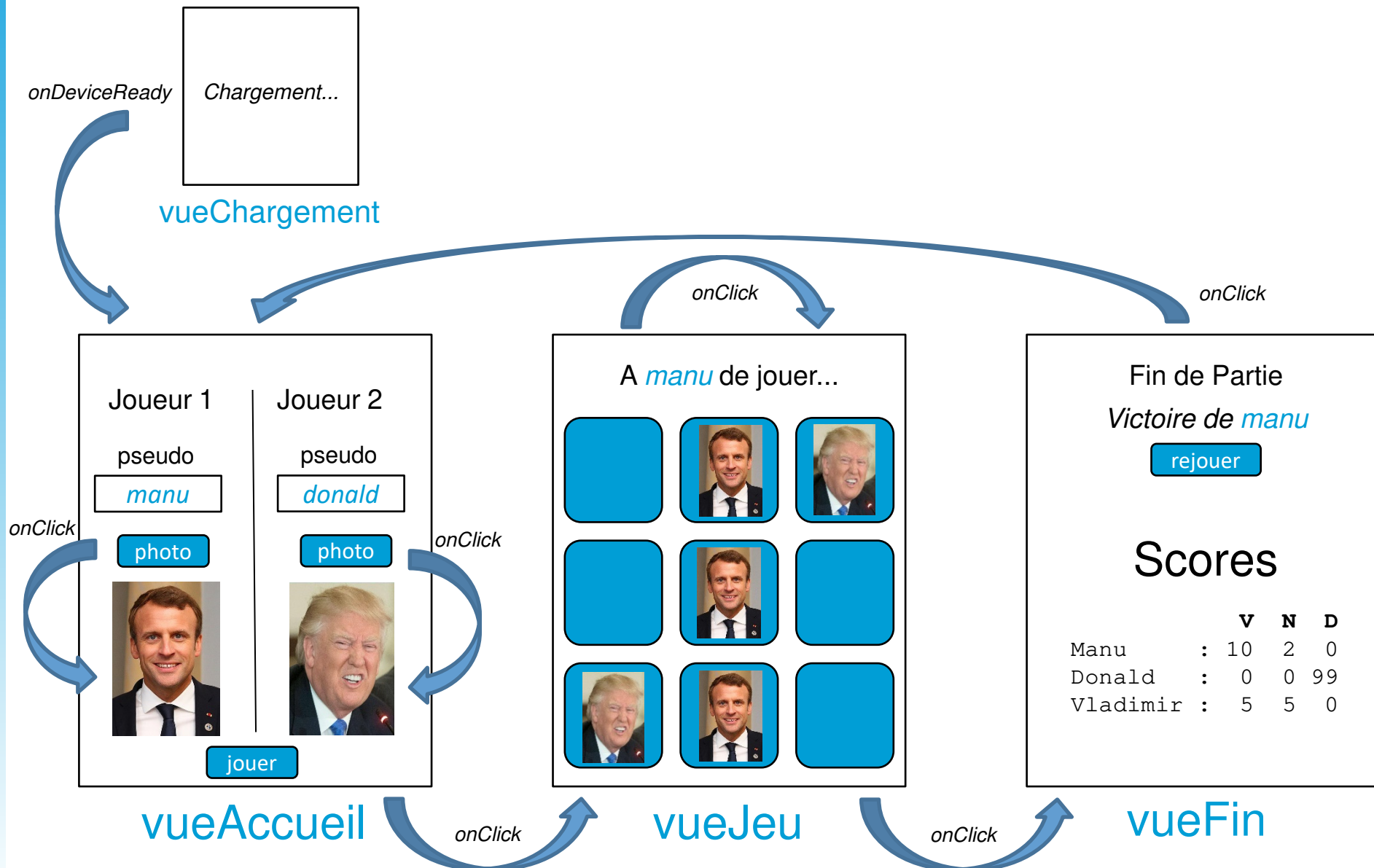
- ❑ Réaliser une application mobile MVC avec Cordova et JQuery Mobile pour jouer au jeu de morpion (Tic-Tac-Toe)
- ❑ Deux joueurs s'affrontent sur une grille carrée de 3x3 cases
- ❑ Ils doivent remplir chacun à leur tour une case de la grille avec le symbole qui leur est attribué
- ❑ Le gagnant est celui qui arrive à aligner trois symboles identiques, horizontalement, verticalement ou en diagonale



Spécifications (1)

- L'application sera composée de 4 pages (vues) JQuery Mobile :
 - Une page de chargement (**vueChargement**), affichée brièvement en attendant l'initialisation de l'application)
 - Une page d'accueil (**vueAccueil**) pour saisir le pseudo et la photo de chaque joueur
 - Une page de jeu (**vueJeu**) pour afficher la grille 3x3 et permettre le déroulement de la partie. La photo d'un joueur sera affichée dans la case qu'il vient de jouer
 - Une page de résultat (**vueFin**) qui s'affichera lorsque la partie est terminée. Elle précisera quel joueur a gagné ou s'il y a eu match nul. Elle affichera également les scores des joueurs ayant joué (nombre de victoires, matchs nuls et défaites)

Spécifications (2)



Mise en Place

- Créer un projet morpion :
create cordova Morpion
cd Morpion
- Configurer le projet :
cordova platform add android
cordova plugin add cordova-plugin-splashscreen
cordova plugin add cordova-plugin-x-toast
cordova plugin add cordova-sqlite-storage
cordova plugin add cordova-plugin-camera
cordova plugin add cordova-plugin-webpack
- Copier le code de l'application **DemoShifumi** que vous modifierez :
Remplacer le fichier config.xml existant par celui fourni dans le ZIP
Remplacer le répertoire www existant par celui fourni dans le ZIP
Copier le répertoire res et le fichier webpack.config.js fournis dans le ZIP
- Pour « compiler » et exécuter :
cordova run <plateforme>
- Pour debugger :
 - lancer chrome et taper l'URL : <chrome://inspect/#devices>
 - Cliquez sur le lien « **inspect** » correspondant à la *webview* de votre application... vous contrôlez le code qui s'exécute sur votre mobile

Travail à faire - Vues

- Vous devez coder en HTML+Jquery mobile le contenu des 3 pages [vueAccueil](#), [vueJeu](#) et [vueFin](#)
- Tout se passe dans le fichier **index.html** de votre projet qui contient le code de toutes les pages (mode « multipage » de Jquery mobile, cf cours n°1)
- Inspirez-vous du code fourni dans Shifumi
- Consultez la documentation de Jquery mobile : <http://demos.jquerymobile.com/1.4.5/>
- Pour organiser du contenu en grilles (grids), consultez la documentation : <http://demos.jquerymobile.com/1.4.5/grids/>

Travail à faire - modèle (1)

- Dans le fichier **js/modele.js**, vous devez coder un prototype **Partie** pour modéliser une partie de morpion
- Ce prototype devra comporter les attributs suivants :
 - Pseudos des 2 joueurs
 - Photos des 2 joueurs (chaînes au format Base64)
 - Tableau 2D pour représenter l'état de la grille

```
var morpion=new Array();  
morpion[0]=new Array(" ", "1", "2");  
morpion[1]=new Array(" ", "2", "1");  
morpion[2]=new Array("1", " ", " ");
```

Travail à faire - modèle (2)

- Le prototype devra également comporter des méthodes :
 - Un constructeur
 - Une méthode pour mettre à jour l'état de la partie lorsque le joueur X a joué en (x,y)
 - Une méthode pour savoir si la partie est terminée et quel en est le résultat (victoire d'un joueur ou match nul)

Travail à faire - modèle (3)

- Le modèle devra également comporter
 - une méthode permettant de récupérer la photo d'un joueur à l'aide du plugin cordova (cf cours n°3 et exemple **DemoPlugins**, méthode takePicture)
 - Un prototype ScoreDAO contenant :
 - Une méthode pour récupérer dans le « local storage » les informations d'un joueur
 - Une méthode pour sauvegarder dans le « local storage » les informations d'un joueur

Travail à faire - contrôleur (1)

- Pour chaque page, vous devez programmer, dans le fichier **js/contrôleur.js**, les méthodes du contrôleur qui seront exécutées lorsque des événements sont déclenchés.
- Pour la page **vueAccueil** :
 - Au chargement de la page (événement *pagebeforeshow*) : (ré)initialisation de la page (vider champs input et image)
 - Click sur un bouton photo : déclenchement de la prise de photo et mise à jour de la balise `` correspondante
 - Click sur bouton jouer : aller à la page **vueJeu**

Travail à faire - contrôleur (2)

- Pour la page **vueJeu** :
 - Au chargement de la page (événement `pagebeforeshow`) : (ré)initialisation de la page (vider la grille)
 - Click sur un bouton de la grille :
 - Mettre à jour la partie en fonction de la case cliquée
 - Désactiver le bouton concerné
 - Si la partie est terminée, aller à la page [vueFin](#)

Travail à faire - contrôleur (3)

- Pour la page **vueFin** :
 - Au chargement de la page (événement *pagebeforeshow*) :
 - Mettre à jour le contenu de la page en fonction du résultat de la partie en cours
 - Mettre à jour la BD
 - Mettre à jour le contenu de la page en fonction des scores
 - Click sur le bouton rejouer:
 - Aller à la page [vueAccueil](#)

S'il reste du temps (optionnel)

- Stocker aussi dans la table des scores les photos des joueurs
- Afficher les photos des joueurs dans la page [vueFin](#)
- Sur la page [vueAccueil](#), faire en sorte que si un pseudo de joueur existe déjà, sa photo soit récupérée en BD
- Mettre en place un « splash Screen » personnalisé (image affichée au démarrage de l'application en attendant son chargement)