

## Atelier-projet 2 : 6 au 11 février 2017

### **FindYourWay:** *Un jeu amusant où tu dois suivre ton chemin sur la carte pour collecter des indices et trouver ta destination finale*

#### **Principes et règles du jeu**

FindYourWay est un jeu disponible sur ordinateur, tablette et téléphone. Le principe est le suivant : une série de points à placer sur une carte est proposée au joueur. Au fur et à mesure qu'il place ces points, un chemin est affiché sur la carte. Lorsque ces points sont bien placés, le joueur récolte un indice qui l'aidera à trouver et placer un dernier point : sa destination finale.

#### Règles du jeu :

- une partie consiste en une séquence de 5 points (ou éléments de chemin) correspondant à des villes ou des lieux à placer sur une carte,
- pour chaque point à placer, une indication textuelle ou photo est proposé au joueur,
- le joueur place les points sur la carte les uns après les autres, le chemin ainsi construit est affiché sur la carte. Lorsque un point est correctement placé (i.e. placé à une distance inférieure à une valeur fixée), un indice sur la destination finale est affiché,
- l'ensemble des indices permet de découvrir l'emplacement de la destination finale, que le joueur doit placer sur la carte. Ces indices peuvent former une adresse, ou une description détaillée permettant de placer avec précision la destination finale,
- si le joueur découvre la destination finale, il marque des points en fonction de la précision avec laquelle cette destination est placée. De plus des informations relatives à cette destination sont éventuellement affichées (url , texte, photos )

#### Règles de calcul du score :

Le score du joueur est calculé en tenant compte de la distance à laquelle il a placé la destination finale par rapport à sa position exacte :

- pour 1 destination placée à une distance  $< D$  : 10 pts
- pour 1 destination placée à une distance  $< 2 D$  : 8 pts
- pour 1 destination placée à une distance  $< 3 D$  : 6 pts
- pour 1 destination placée à une distance  $< 5 D$  : 3 pts
- pour 1 destination placée à une distance  $< 10 D$  : 1 pt

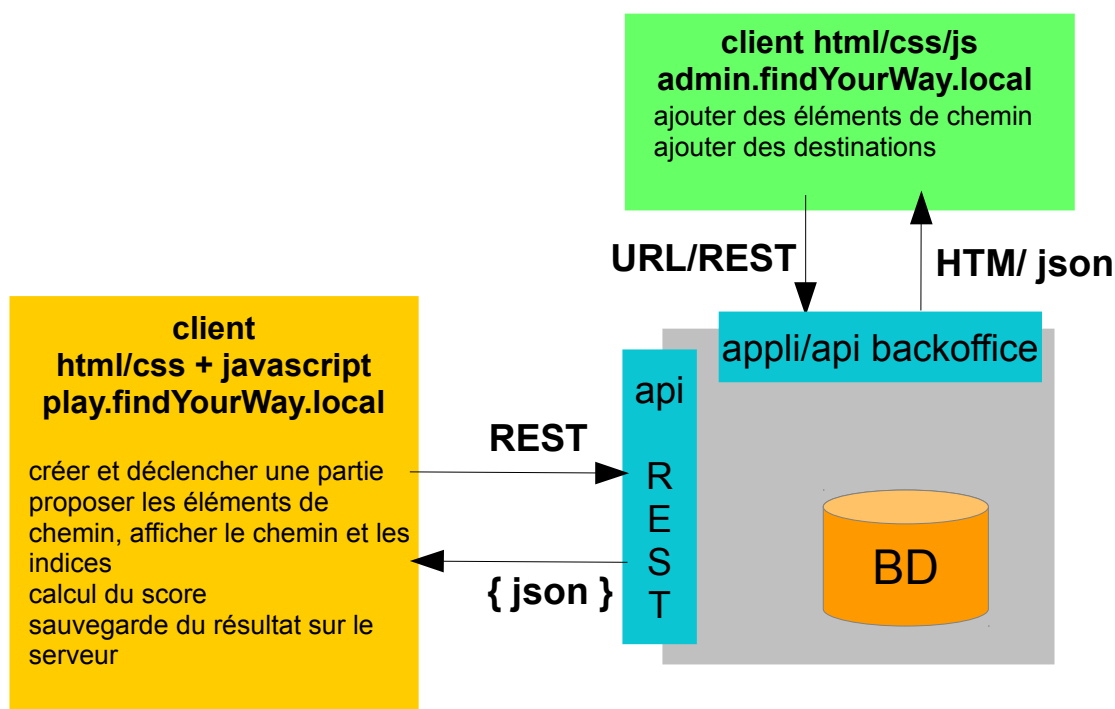
( la distance  $D$  est une valeur à choisir, qui peut être un paramètre de l'application. Elle peut être assez faible, les indices pouvant donner des indications précises).

## Architecture :

FindYourWay est constituée de 2 applications distinctes :

- **l'application de jeu** proprement dite, qui permet de dérouler des parties ; elle est entièrement programmée en javascript et dialogue avec le serveur au travers d'une API REST qui retourne des données json. Elle est accessible à l'URL <http://play.finyourway.local>
- **l'application backoffice**, qui permet d'alimenter les données nécessaires au jeu, en particulier pour :
  - ajouter et géolocaliser des éléments de chemin : indication textuelle ou photo + position sur la carte,
  - ajouter, documenter et géolocaliser des destinations finales : 5 indices + position sur la carte + informations finales.

Cette application est accessible à l'URL <http://backend.findyourway.local> . Elle est programmée au choix soit sous la forme d'une application serveur utilisant des templates et produisant des pages html, soit sous la forme d'une webapp client utilisant une api.



## L'application backoffice

L'application permet à des utilisateurs inscrits (inscription libre) et authentifiés d'alimenter la base de jeu en créant :

- des éléments de chemin : il s'agit d'un point à placer sur la carte. Il faut donc fournir une indication qui sera proposé au joueur et la position de ce point sur la carte,
- des destinations finales : décrites par une série de 5 indices, une position sur la carte et des informations complémentaires dévoilées lorsque la destination a été trouvée : texte, photos, url.

Ces données sont utilisées pour alimenter le moteur de jeu : pour chaque partie, un chemin aléatoire est généré à partir des éléments de chemin enregistrés dans la base, et une destination finale est choisie au hasard à partir des destinations stockées dans la base.

## L'application de jeu

L'application de jeu permet à n'importe quel utilisateur (authentification non requise) de démarrer et jouer une partie. Lorsque la partie est démarrée, une carte et une liste de points à placer apparaît. On peut au choix afficher les points un par un ou tous à la fois, mais le joueur doit les placer dans l'ordre.

L'utilisateur place un point en cliquant sur la carte. Un marqueur est alors positionné, et le point est relié au précédent pour afficher le chemin. Si le point est bien placé (à une distance inférieure à un seuil à déterminer), un indice est affiché.

Lorsque les 5 points sont placés, l'utilisateur peut alors placer la destination finale. Le score et les informations finales sont alors affichés. L'utilisateur peut demander à stocker son score sur le serveur.

## Déroulement technique

Pour démarrer une partie, il faut d'abord la créer sur le serveur. Cette création retourne notamment un **token** identifiant la partie qui doit être ensuite ajouté à toute les requêtes concernant la partie, afin d'éviter que d'autres utilisateurs utilisent cette partie.

Sur une partie, on peut obtenir :

- des informations générales pour toute la partie : la valeur de la distance D à utiliser pour le calcul du score, le token d'identification de la partie
- la liste des points de chemin à placer, avec pour chacun, l'indication de placement et la position exacte,
- la destination finale, avec notamment les indices et sa position exacte.

Le **déroulement classique** d'une partie correspond ainsi à l'enchaînement suivant :

1. création d'une nouvelle partie avec les paramètres transmis : pseudo du joueur, niveau de jeu éventuel ; la webapp reçoit en retour l'id de la partie et le token associé,
2. récupération de la liste des éléments de chemin : indication +position exacte,
3. récupération de la destination finale : indices + position exacte
4. **Déroulement du jeu** : les éléments de chemin sont affichés, les indices sont fournis, le score est calculé. **Aucune interaction avec le serveur ne doit être nécessaire pendant le déroulement du jeu.**
5. Lorsque la partie est terminée sur le client : récupération et affichage des informations complémentaires sur la destination ; mise à jour de la partie sur le serveur ; le score de l'utilisateur est enregistré, la partie est marquée comme achevée.

Une partie peut donc être dans 3 états : créée, en cours de jeu , terminée.

### **Fonctionnalités minimales:**

- application jouable, avec 5 éléments de chemin par partie, 1 seul niveau de jeu,
- distance D pour le calcul des points identique pour toutes les destinations,
- backend permettant d'ajouter des éléments de chemin et des destinations ; pas d'authentification pour la partie backend,
- les éléments de chemin et les indices sont textuels : pas d'images,

### **Extensions proposées :**

- inscription et authentification pour accéder à la partie backend,
- possibilité de mettre des images dans les descriptions d'éléments de chemin,
- possibilité de mettre des images dans les indices,
- décompte du temps pour limiter le délai accordé pour 1 réponse (chemin ou destination finale),
- possibilité de suspendre 1 partie pour la reprendre plus tard. Stockage des données local (LocalStorage),
- différents niveaux de jeu, qui peuvent se différencier par :
  - la difficulté pour obtenir un indice (distance) lors du placement d'un élément de chemin,
  - la distance D pour valider la destination finale,
- accès au tableau des meilleurs scores
- etc ...

### **Priorités de réalisation :**

1. implantation de l'API, avec une base de test,
2. réalisation du moteur de jeu
3. connexion du moteur de jeu à l'API : création de la partie, accès aux données de jeu au travers de l'API
4. ajout de nouvelles données au travers de l'interface de gestion backoffice,
5. différentes extensions dans la liste suggérée, dans l'ordre voulu.

### **Rendu :**

Dans l'espace Arche prévu à cet effet, date limite de dépôt le 11/02/2017, 12h :

- lien vers un dépôt git public (bitbucket, github, ...) contenant le code complet des applications, les scripts de création de base de données, les éléments de configuration d'une VM permettant d'exécuter l'application (Vagrantfile, dockerfile) etc ...
- le dépôt DOIT contenir un README précisant les directives d'installation de l'application,
- un rapide CR de ce qui a été réalisé, contenant notamment toute donnée nécessaire pour le test (noms d'utilisateurs, mdp ...),

- une documentation de l'API permettant de l'utiliser et de la tester en dehors de l'application cliente,
- **un tableau de bord du projet indiquant l'état, les priorités et les auteurs des stories identifiées.**

### **Rappels :**

- travail en groupe de 4,
- le rendu sur le client doit être réalisé avec un framework css à choisir parmi :
  - bootstrap/sass, foundation 5 ou foundation for apps, gumby, materialize, semantic-ui,
  - intégration du framework dans le projet basée sur un processeur css (type sass/scss),
- la librairie leaflet.js vous propose toutes les fonctionnalités utiles pour afficher une carte. Il est notamment possible de récupérer les coordonnées d'un point de click (voir MouseEvent), et de calculer la distance entre 2 points (méthode LatLng.prototype.distanceTo() ).