

CityLocate : *Un jeu amusant où tu dois positionner une photo sur la carte de ta ville sans te tromper et plus vite que les autres !*

Principes et règle du jeu

CityLocate est un jeu disponible sur ordinateur, tablette et téléphone. Le principe est simple : une photo est proposée au joueur, il doit indiquer la position de ce qu'elle représente sur une carte d'une ville.

Règle du jeu :

- une partie consiste en une séquence de 10 photos à placer sur la carte d'une ville,
- toutes les photos d'une série concernent la même ville, et donc la même carte,
- chaque réponse permet de gagner un certain nombre de points, en fonction de la précision du placement et de la rapidité pour répondre,
- l'objectif pour une partie est d'obtenir le maximum de points.
- la partie est terminée lorsque les 10 photos ont été positionnées.

Règles de calcul des points :

Placement des réponses :

- pour 1 réponse placée à une distance $< D$: 5pts
- pour 1 réponse placée à une distance $< 2D$: 3pts
- pour 1 réponse placée à une distance $< 3D$: 1pts

(la distance D est une valeur à choisir, qui peut être un paramètre de l'application)

Prise en compte de la rapidité :

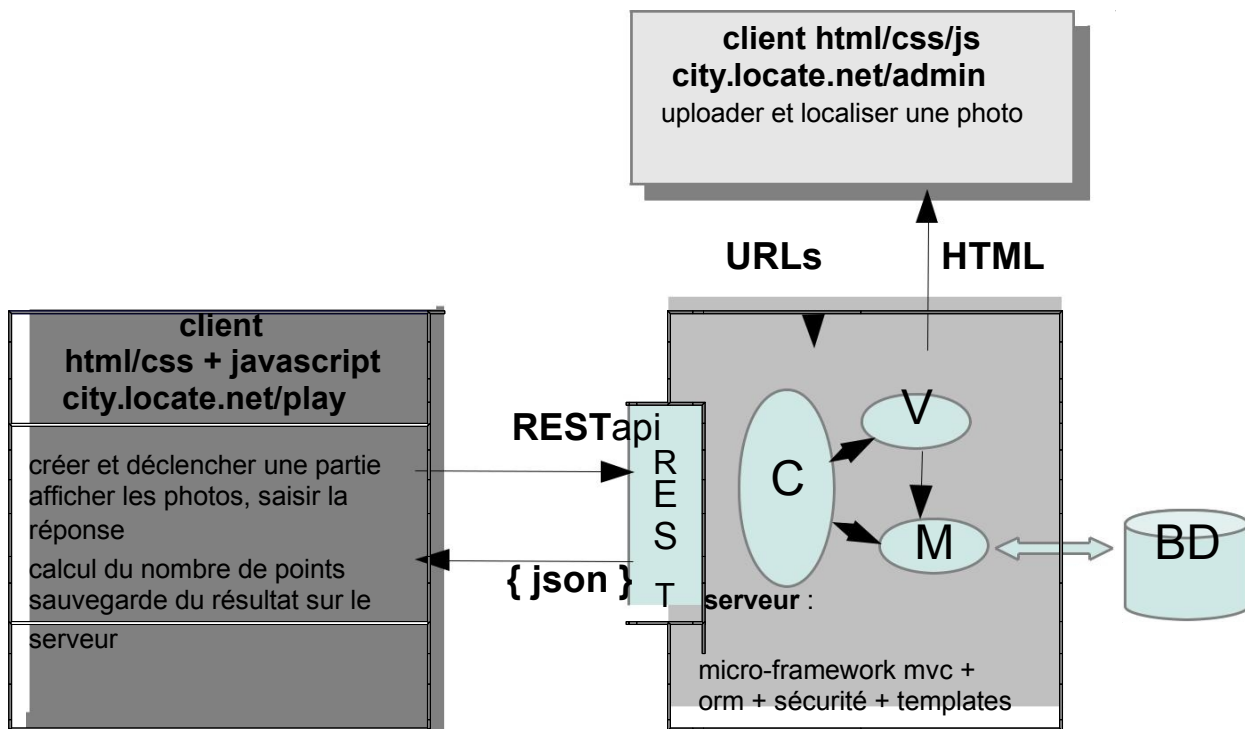
- les pts sont multipliés par 4 pour une réponse en moins de 2s
- les points sont multipliés par 2 pour 1 réponse en moins de 5s
- les points ne sont pas acquis pour 1 réponse en plus de 10s

Architecture de l'application

L'application est constituée de 2 modules distincts :

- le module de jeu proprement dit, qui permet de dérouler des parties ; il est entièrement programmé en javascript et dialogue avec le serveur au travers d'une API REST qui retourne des données json. Il est accessible à l'URL <http://city.locate.net/play> .
- le module d'administration, qui permet d'alimenter les données nécessaires au jeu, en particulier pour ajouter et geolocaliser des photos ; il est entièrement programmé sur le serveur qui produit des pages HTML grace à des templates. Il est accessible

à l'URL <http://city.locate.net/admin> .



Architecture de l'application CityLocate

Le module d'administration

Le module d'administration permet à des utilisateurs inscrits (inscription libre) et authentifiés de charger une photo et de la géolocaliser afin d'alimenter la base de jeu.

Le module est une application web classique dont l'interface html est entièrement produite par le serveur, en utilisant un mécanisme de templates.

Le module de jeu

Le module de jeu permet à n'importe quel utilisateur (authentification non requise) de démarrer et jouer une partie. Lorsque la partie est démarrée, la 1ère photo s'affiche, ainsi qu'une carte sur laquelle il est possible de placer un marqueur correspondant à la réponse. Dès que le marqueur est positionné, le score est calculé et affiché. L'utilisateur peut ensuite passer à la photo suivante.

Lorsque la série de photos est terminée, le score final est affiché. L'utilisateur peut transmettre ce score au serveur pour entrer dans la liste des meilleurs scores.

Le module de jeu est entièrement programmé en javascript et dialogue avec le serveur uniquement au travers d'une API REST. Cette API retourne du json qui est traité par le client pour dérouler la partie.

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 toutes 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 carte à utiliser, la valeur de la distance D à utiliser pour le calcul des points, le nom de la ville concernée, l en nombre de photo prévues dans la partie

- la liste des photos, avec la position exacte de chacune d'elle.

L'API REST est définie ainsi :

méthode	uri	effet
POST	.../play/games/	création d'une partie ; données : pseudo de l'utilisateur retourne : l'identifiant de la nouvelle partie + 1 token associé à cette partie
GET/play/games/:id?token=.....	retourne la description d'une partie : la carte à utiliser pour le positionnement, autres informations utiles
GET/play/games/:id/photos/?token=.....	retourne la liste des 10 photos à placer pour dérouler la partie, avec la position exacte de chaque photo sur la carte
PUT/play/games/:id?token=.....	mise à jour d'une partie : - score final de l'utilisateur - état de la partie : en cours / terminée

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

1. POST /play/games/ : création d'une nouvelle partie avec les paramètres transmis (pseudo, ville éventuelle, niveau ...) ; en retour : id de la partie, token
2. GET /play/games/id : récupération de la ressource décrivant la partie : références de la carte, nb de photos ...
3. GET /play/games/id/photos : récupération de la liste des photos de la partie ; avec chaque photo : position exacte + infos utiles
4. Déroulement du jeu : les photos sont affichées les unes après les autres, calcul des points au fur et à mesure de l'avancement,
5. PUT /play/games/id : mise à jour de la partie, le score de l'utilisateur est enregistré, la partie est marquée comme achevée.

Fonctionnalités minimales:

- une seule ville : Belfort (par exemple),
- distance D pour le calcul des points identique pour toutes les parties,
- pas d'authentification pour la partie admin,
- 10 photos par partie,

Extensions :

- inscription et authentification pour accéder à la partie administration,
- plusieurs villes possibles : le choix de la ville est fait à la création de la partie ; l'ajout de photo permet d'indiquer la ville associée,
- possibilité de suspendre 1 partie entre 2 photos pour la reprendre plus tard. Stockage des

données local (LocalStorage),

- différents niveaux de jeu, qui se différencient par le nombre de photos à localiser dans la partie et par la distance D utilisée pour calculer les points. Par exemple :
 - niveau 1 : D
 - niveau 2 : $\frac{3}{4}$ D
 - niveau 3 : $\frac{1}{2}$ D etc...
- accès au tableau des meilleurs scores

Priorités de réalisation :

1. implantation de l'API, avec une base de test contenant quelques photos
2. réalisation du moteur de jeu sur un ensemble de photos fixe
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 photos au travers de l'interface d'administration,
5. différentes extensions dans la liste suggérée, dans l'ordre voulu.

Rendu :

- lien vers un dépôt git public (bitbucket, github, ...), ou privé à condition que les enseignants concernés se voient attribuer le droit de cloner.
- une url pour tester l'application (serveur IUT ou ailleurs)
- toute donnée nécessaire pour le test (noms d'utilisateurs, mdp ...)
- un rapide CR de ce qui a été réalisé
- **un tableau de bord du projet indiquant l'état, les priorités et les auteurs des stories identifiées.**

Rappels :

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()`).

Trouver les coordonnées d'un point de la ville :

Etape 1 : trouver l'adresse du point recherché : <https://www.google.fr/maps>

Etape 2 : trouver les coordonnées GPS de l'adresse : <https://www.coordonnees-gps.fr/>

Etape 3 : rentrer dans votre Base de donnée les coordonnées Longitude et Latitude du point.

Exemple de point sur Belfort : IUT Belfort, Lion de belfort, Gare de Belfort, Gare TGV, Base de loisir du Lac du Malsaucy,, Bowling de Belfort...

La distance D sera initialisé en javascript au début du code (pour les plus confirmé, une requête ajax vers l'appli backend PHP, récupérer le paramètre D de la base de donnée dans une table params)

Dès que le jeux est commencé (dès que le bouton jouer est cliqué (onClick ou autre...)) on démarre le chronomètre de temps.

Mode simplifié : initialisation des emplacements directement en javascript (tableau json)