TP - Mémo LoL (2/3)

Université Toulouse 2 - Jean Jaurès 29 janvier 2018

Présentation

Dans ce TP, il est question de développer le jeu du memory en utilisant les données du jeux League Of Legends (LoL) dont la présentation du jeu est disponible ici [1]. Ce jeu propose à 10 joueurs de s'affronter en 5 contre 5, chacun incarnant un champion et devant protéger une base tout en essayant de détruire la base ennemie. Riot Games™ (l'entreprise qui a créé ce jeu) a mis à disposition une API pour récupérer les données du jeu. Notamment les données concernant les champions disponibles. Nous proposons ici de développer le jeu du memory en utilisant ces données pour que les cartes représentent des champions du jeu LoL.

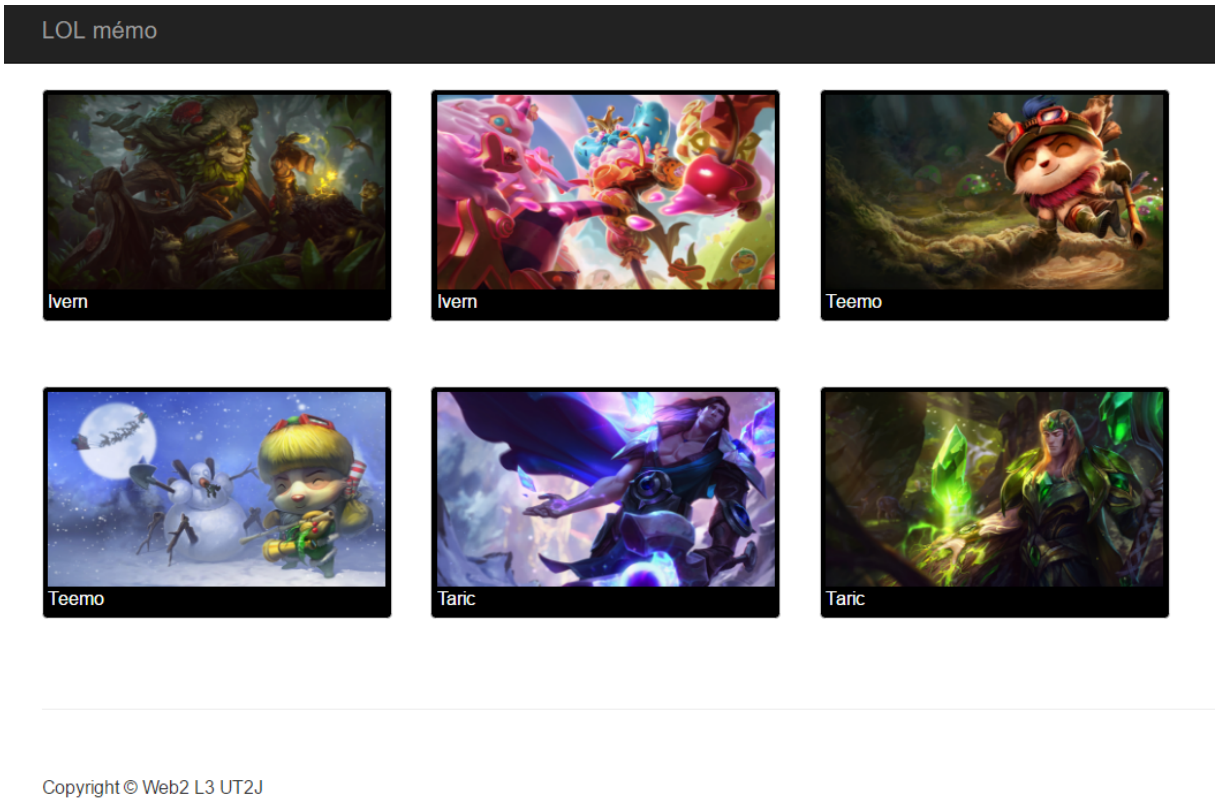
Le jeu du memory consiste à avoir un nombre paire de cartes faces cachées présentées au joueur. Celui-ci doit retrouver les paires de cartes correspondantes. Pour cela il peut retourner simultanément 2 cartes. Si celles ci correspondent (le nom du champion est le même) alors la paire reste retournée. Sinon les deux cartes sont remises faces cachées et le joueur tente de retourner une autre paire de cartes. Le jeu se terminer lorsque toutes les paires sont trouvées.

Objectif de la seconde partie de ce TP : développer toute la partie métier de l'application avec *au choix* le framework Angular 1.* ou Angular 2.*

1. `http://euw.leagueoflegends.com/fr`

1 Interface

Reprendre l'interface développée dans le TP précédent avec les frameworks CSS Bootstrap et Less Votre plateau doit ressembler à celui ci :



2 Application

Dans cette deuxième partie, vous devez développer toute la partie métier de l'application. Vous allez donc utiliser le framework AngularJS vu en cours pour faire ce développement. L'objectif est d'avoir le jeu du memory fonctionnel.

En utilisant l'API proposée par Riot Games™ [2] vous récupérerez les informations des champions du jeu League Of Legends. Pour cela il est nécessaire d'avoir un compte sur les serveurs de Riot. C'est le même compte que celui utilisé dans le jeu, sinon il est facile d'en créer un rapidement à cette URL :
`https://signup.euw.leagueoflegends.com/fr/signup/index?realm_key=euw`
Vous remarquerez que cette API propose un grand nombre de *webservices* qui permettent de récupérer toutes les informations relatives au jeu, aux parties en cours et aux comptes. N'hésitez pas à explorer cette API et à proposer des idées d'applications innovantes.

2. `https://developer.riotgames.com/`

2.1 Génération aléatoire du plateau de jeu

Jusqu'ici les champions apparaissant sur le plateau de jeu de notre memory étaient prédéterminés (Ivern, Teemo et Taric). Nous allons maintenant interroger l'API de Riot pour sélectionner 3 champions de manière aléatoire parmi tous les champions disponibles. Pour cela nous utiliserons le *webservice* suivant :
`https://global.api.pvp.net/api/lol/static-data/euw/v1.2/champion?locale=fr_FR&api_key=VOTRE_CLEF` [3]

Ce webservice retourne un objet JSON respectant le modèle suivant :

- "data"
  - nom du champion :
    - "id" : identifiant du champion
    - "title" : titre du champion (nom long)
    - "name" : nom du champion
    - "key" : nom du champion en CamelCase [a]

a. `https://fr.wikipedia.org/wiki/CamelCase`

Listing 1 – "Exemple de JSON"

```
{
   "data": {
      "Aatrox": {
         "id": 266,
         "title": "the Darkin Blade",
         "name": "Aatrox",
         "key": "Aatrox"
      },
      "Thresh": {
         "id": 412,
         "title": "the Chain Warden",
         "name": "Thresh",
         "key": "Thresh"
      },
      ...
    }
}
```

Dans cette liste vous sélectionnerez 3 champions aléatoirement et utiliserez l'attribut "key" pour obtenir les deux cartes correspondantes :

- `https://ddragon.leagueoflegends.com/cdn/img/champion/splash/{{KEY}}_0.jpg`
- `https://ddragon.leagueoflegends.com/cdn/img/champion/splash/{{KEY}}_1.jpg`

2.2 Retourner une paire de cartes

L'animation du retournement d'une carte s'effectue, jusqu'à présent, lors du survol de la souris sur celle-ci. Vous modifierez l'évènement de retournement d'une carte pour que l'animation apparaisse lors du clique de l'utilisateur sur la carte.

De plus vous ne donnerez la possibilité de n'avoir que 2 cartes retournées en même temps. Pour cela, lorsque la deuxième carte est retournée, vous les laisserez retournées pendant 1sc [4] et ensuite vous retournerez ces deux cartes face cachée. Attention à ce que pendant ce temps, aucune autre carte ne puisse être retournée.

2.3 Déroulement de la partie

Pour pouvoir déterminer si l'utilisateur a gagné ou non vous implémenterez les fonctionnalités du déroulement de la partie.

3. Pensez à remplacer "VOTRE_CLEF" par la clef fourni pour votre compte, la documentation étant disponible ici `https://developer.riotgames.com/api/methods#!/1055/3633`
4. Indice : `https://docs.angularjs.org/api/ng/service/\protect\T1\textdollartimeout#!`

Tout d'abord, lorsque 2 cartes sont retournées, il faut vérifier si ces deux cartes correspondent au même champion (même "key"). Si c'est le cas alors ces deux cartes restent face visible (et ne sont donc plus cliquable). Le joueur ayant trouvé une paire, il devra trouver les autres paires du plateau.

Chaque fois qu'une paire est trouvée (2 cartes du même champion retournées) il faut vérifier s'il reste des cartes face cachée sur le plateau. Si ce n'est pas le cas alors l'utilisateur a gagné. Dans cette situation, vous afficherez un message indiquant au joueur qu'il a gagné et vous relancerez une nouvelle partie (en redéterminant les champions).

3 Partie bonus

Il est possible d'améliorer ce jeu de divers manières. Il vous est proposé ici 2 améliorations que vous pouvez implémenter.

3.1 Nombre dynamique de cartes

Jusqu'ici le plateau contient 6 cartes en tout (2 pour les 3 champions selectionnés). Afin de déterminer le niveau de difficulté du jeu il est interessant de proposer au joueur la possibilité de choisr le nombre de carte à afficher. Ajoutez une interface avant le lancement du jeu pour demander à l'utilisateur combien de cartes il souhaite sur le plateau.

3.2 Calcul des points

Il est possible de comptabiliser les points d'une partie en considérant un nombre de point initial et en décrémentant ce score pour chaque paire de carte retournée. Par exemple nous pouvons commencer la partie avec 40 points, si nous retournous 17 paires de cartes avant de trouver toutes les paires associées (et donc finir la partie), notre score sera de 23 points. Si ce score atteint 0 alors le joueur a perdu.

3.3 Nouvelles idées

Comme précisez précédement, il existe de nombreuses façon d'améliorer ce jeu. Vous avez probablement plein d'idées alors n'hésitez pas à proposer de nouvelles améliorations (et aussi à les implémenter !).


League of Legends?
You mispelled Draaaaaaaaven!
quickmeme.com