



W3 - Tests

W-WEB-310

my_tcg

Bootstrap



my_tcg

language: javascript
build tool: no need here



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

PREMIÈRE PARTIE : RENSEIGNEMENTS.

Pour commencer ce Bootstrap, nous vous demandons de vous renseigner sur les outils suivant :

- gulp
- babel
- browserify

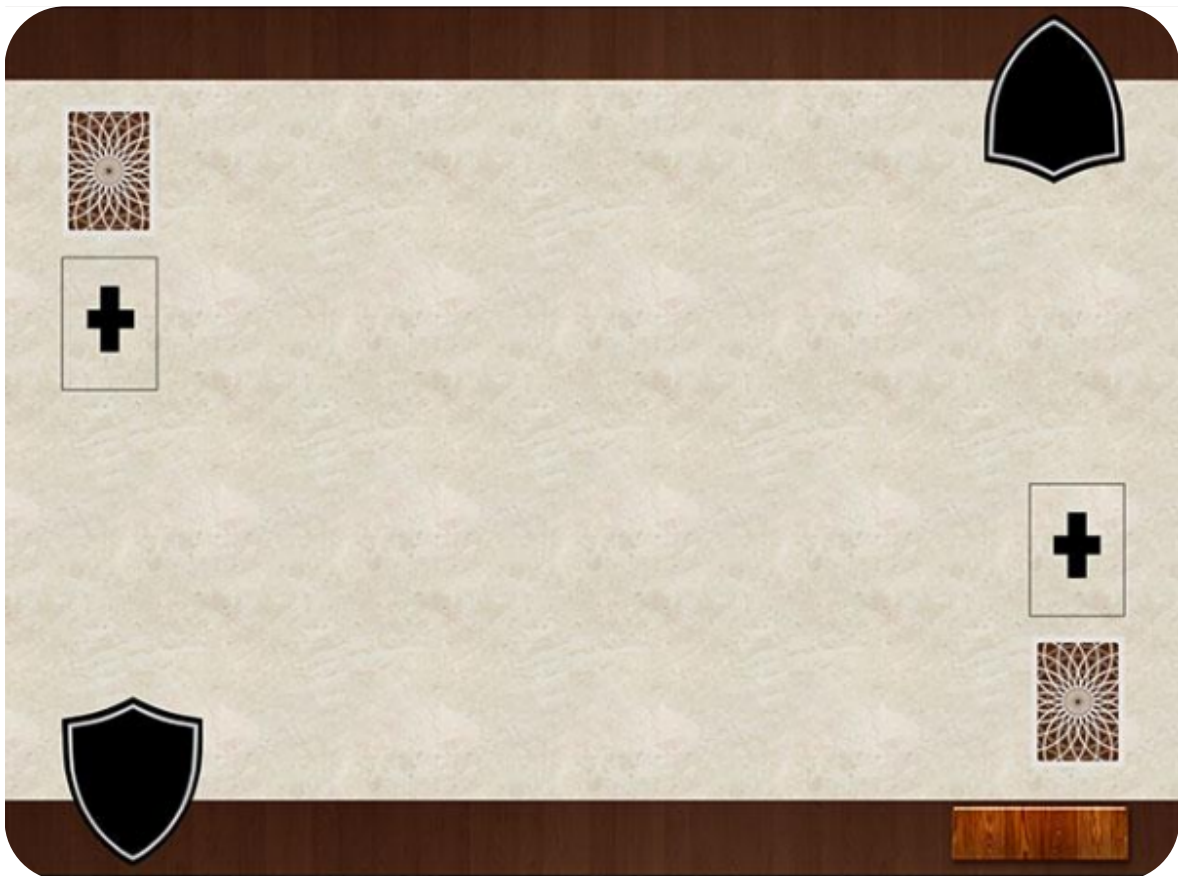


Vous irez donc vous renseigner sur le rôle et l'utilisation de chacun.

DEUXIÈME PARTIE : MISE EN PLACE

Le développeur en charge de la partie visuelle vous a livré un pack de scripts (jointe au sujet). Mais une fois décompressé, en ouvrant le script `index.html` (dans le dossier public), vous n'obtenez qu'une page blanche... Vous êtes pourtant convaincu de la fiabilité de la personne qui vous a livré le code. Vous avez donc sans doute oublié de faire quelque chose pour que l'interface s'affiche (il n'y a aucune ligne de code à toucher à cette étape...).

Voici ce que doit afficher la page `index.html` une fois l'installation du projet terminée :



`package.json` est un fichier de configuration lié à npm. Savez-vous installer les dépendances de projet liées à npm ?



Savez-vous à quoi sert Gulp ? Comment l'exécuter ?



Il est inutile de continuer si vous n'avez pas réussi à afficher la page `index.html`.

TROISIÈME PARTIE : MES PREMIERS TESTS UNITAIRES

L'une des pires choses qui puisse arriver à un développeur (toutes technologies confondues), c'est de provoquer des régressions (de nouveau bugs dans une partie d'application qui fonctionnait jusqu'alors). Il faut re-faire (débuguer) un travail que l'on a déjà fait, c'est ennuyeux et parfois le code à déboguer n'est plus tout jeune (on ne s'en souviens plus) ou pire encore, on est pas la personne à l'origine du code en question. Pour se prémunir au maximum de ce genre de complications, et pour simplifier le débogage quand c'est inévitable, les développeurs ont inventé les tests unitaires.



Renseignez-vous sur la philosophie des tests unitaires, pourquoi unitaires ? Qu'est-ce que cela implique ?

Pour vous avancer, dans le début de projet qui vous a été livré, l'environnement de test est déjà prêt (nous utiliserons le framework de test « Jasmine »). Vous devez trouver la commande (à taper dans la console) qui exécute les tests. Vous devriez obtenir quelque chose ressemblant à ceci :

```
Terminal
~/W-WEB-310> ?
> my_hearthstone@0.0.1 test /mnt/c/wamp/www/my_tcg
> babel-node -presets es2015 spec/run.js

Spec started

Executed 0 of 0 specs SUCCESS in 0.017 sec.
```