

Note sur le test
de l'application
GC partie CVA
et
Authentication

I - Introduction

Je vais dans cette note aborder les technologies utilisées ainsi que les difficultés rencontrées lors de l'utilisation de chacune d'elles d'une manière détaillée et avec captures d'écran à l'appui.

Le but est de mettre en avant la difficulté de mettre en place un environnement de travail permettant de tester de manière efficace l'application Grand Cercle mais également celle de tester certains composants de notre application due principalement à l'utilisation de react-navigation.

Ainsi, je vais aborder principalement Jest et Mocha, les deux cadres de test Javascript que j'ai utilisés au cours des dernières semaines.

II – Jest

Cf: <https://jestjs.io/>

Comme on peut le lire sur le site dédié, Jest est un cadre de test consacré au Javascript qui fonctionne très bien avec React. Cependant, il a déjà plus de problèmes avec react-native.

Plusieurs tentatives infructueuses ont été tentées pour mettre en place, directement dans le dossier fourni, un environnement de travail permettant de tester l'application avec Jest de manière efficace.

L'utilisation de Jest reste pour le moment bloquée à ce stade pour ce dossier, rendant impossible la création de snapshots (instantanés):



```
FAIL src/Activity/MainActivity.snapshot.js
● Test suite failed to run

Jest encountered an unexpected token

This usually means that you are trying to import a file which Jest cannot parse, e.g. it's not plain JavaScript.
By default, if Jest sees a Babel config, it will use that to transform your files, ignoring "node_modules".

Here's what you can do:
  • To have some of your "node_modules" files transformed, you can specify a custom "transformIgnorePatterns" in your config.
  • If you need a custom transformation specify a "transform" option in your config.
  • If you simply want to mock your non-JS modules (e.g. binary assets) you can stub them out with the "moduleNameMapper" config option.

You'll find more details and examples of these config options in the docs:
https://jestjs.io/docs/en/configuration.html

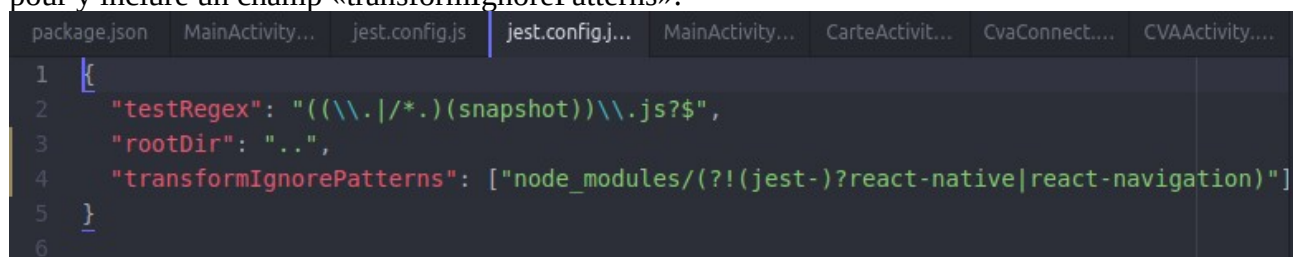
Details:

/home/halfbad/ProjetGC/node_modules/react-native-status-bar-height/index.js:1
({Object.<anonymous>":function(module,exports,require,__dirname,__filename,global,jest){import { Dimensions, Platform, StatusBar } from 'react-native';
                                                                                                                                              ^^^^^^^
SyntaxError: Unexpected token import

   3 |   import { NavigationActions } from 'react-navigation';
   4 |   import { getPartenariats, partenairesAreLoaded } from './Partenariats/DataLoader';
>  5 |   import { getStatusBarHeight } from 'react-native-status-bar-height';
     |   ^
   6 |   import { getEvents } from './Actualites/EventLoader';
   7 |   import { getAssosAndEvents, eventsAreLoaded } from './Actualites/Assoloader'
   8 |
at ScriptTransformer._transformAndBuildScript (node_modules/@jest/transform/build/ScriptTransformer.js:471:17)
at ScriptTransformer.transform (node_modules/@jest/transform/build/ScriptTransformer.js:513:25)
at Object.<anonymous> (src/Activity/MainActivity.js:5:1)
```

Image: Capture d'écran du terminal après la commande `npm run test:snapshot` qui exécute jest

Suivant les instructions de l'erreur affichée, j'ai modifié le fichier de configuration jest.config.json pour y inclure un champ «transformIgnorePatterns»:



```
package.json  MainActivity...  jest.config.js  jest.config.j...  MainActivity...  CarteActivit...  CvaConnect....  CVAActivity....

1  {
2    "testRegex": "((\\.|/*.)?(snapshot))\\.js?$",
3    "rootDir": "..",
4    "transformIgnorePatterns": ["node_modules/(?!(jest)?react-native|react-navigation)"]
5  }
6
```

Image: Capture d'écran du fichier de configuration de jest, jest.config.json

Cependant, le résultat dans la console reste le même.

Pour le moment, malgré bon nombre de recherches Google ainsi que de fouilles de StackOverflow et GitHub, je n'ai pas réussi à surpasser ce problème.

J'ai donc décidé de partir d'un nouvel environnement de travail en récupérant celui dédié au travail avec react-native fourni par Facebook sur GitHub: <https://github.com/facebook/react-native>
Après avoir incorporé notre projet et réalisé les quelques adaptations nécessaires, j'ai pu créer mes premiers fichiers tests et les tester:

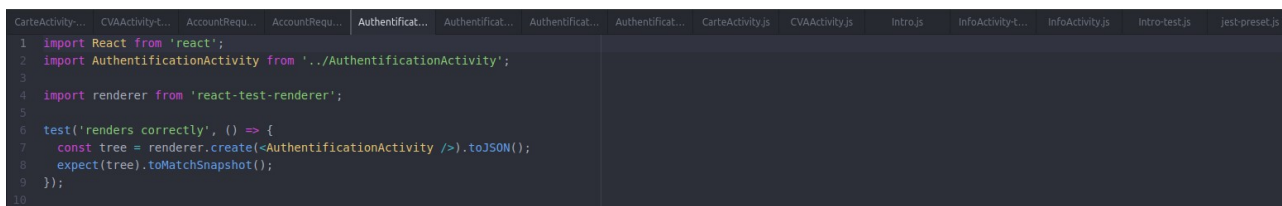


Image: Capture d'écran du fichier test, AuthenticationActivity-test.js testant le composant AuthenticationActivity

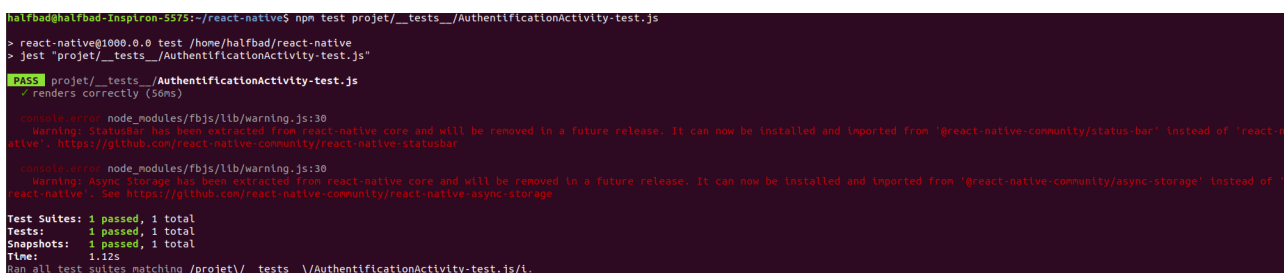


Image: Capture d'écran du terminal après la commande `npm test projet/__tests__/AuthenticationActivity-test.js` exécutant jest

Comme on peut le voir sur la deuxième image, le test passe avec succès.

Cependant, les problèmes arrivent dès qu'on veut capturer un instantané d'un composant utilisant `this.props.navigation.state` qui renvoie `undefined` dans sa méthode `render()`. Cela induit le problème suivant lors de l'exécution de jest:

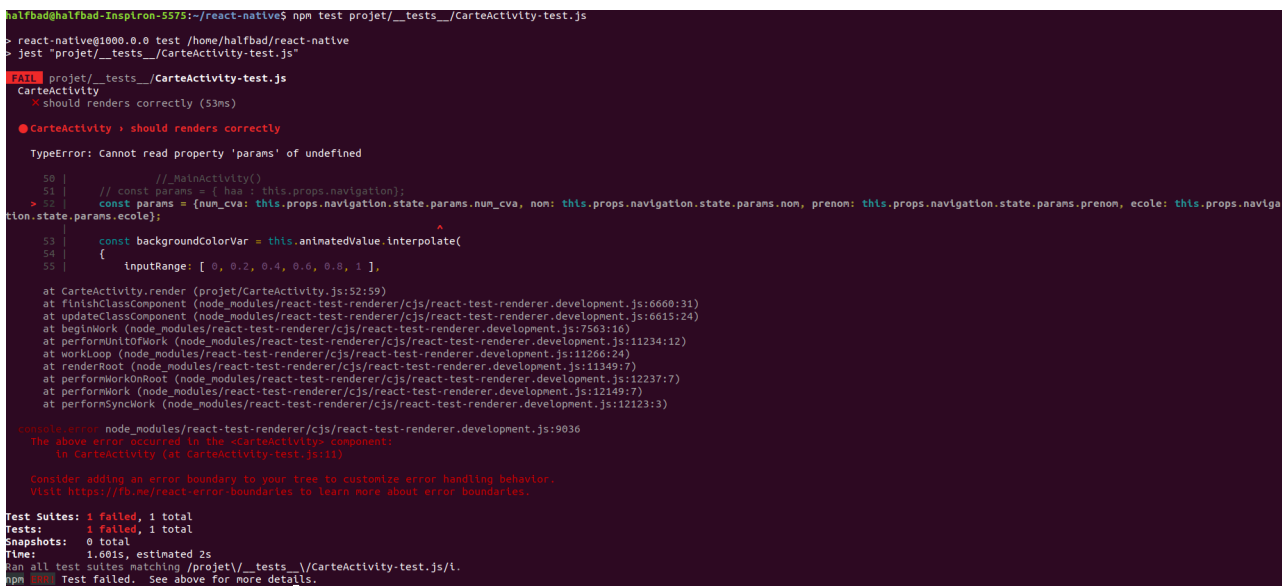


Image: Capture d'écran du terminal après la commande `npm test projet/__tests__/CardActivity-test.js` exécutant jest

Suivant les réponses sur les forums StackOverflow et GitHub, j'ai modifié mon fichier test plusieurs fois en combinant en dernier lieu les méthodes proposées dans ces liens:

- <https://stackoverflow.com/a/46114589>
- <https://github.com/react-navigation/react-navigation-stack/issues/72>

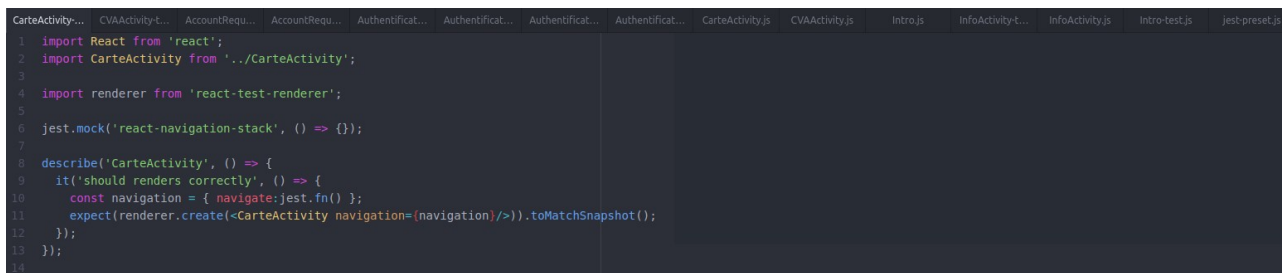


Image: Capture d'écran du fichier test, *CarteActivity-test.js* testant le composant *CarteActivity*

Cependant, cela n'a apporté aucune modification au rendu du test et s'est solvé par le même échec.

Me retrouvant bloqué, j'ai décidé de tenter de tester autre chose, en l'occurrence, les clics dans l'application et pour cela, j'ai suivi la méthode donnée dans l'article suivant:

<https://medium.com/codeclan/testing-react-with-jest-and-enzyme-20505fec4675> (partie Mock Functions).

En utilisant Enzyme, un utilitaire de test Javascript pour React, en combinaison avec Jest, on peut imiter un clic sur un composant. Cela fonctionne mais encore une fois, l'utilisation de react-navigation pose problème et empêche l'imitation de se faire correctement. Or, la majorité des clics sur l'application consiste justement à faire des changements d'écrans donc des appels à des méthodes issues de react-navigation. On obtient ainsi l'erreur suivante que je n'ai pas réussi à corriger malgré l'aide de Google et des diverses forums:

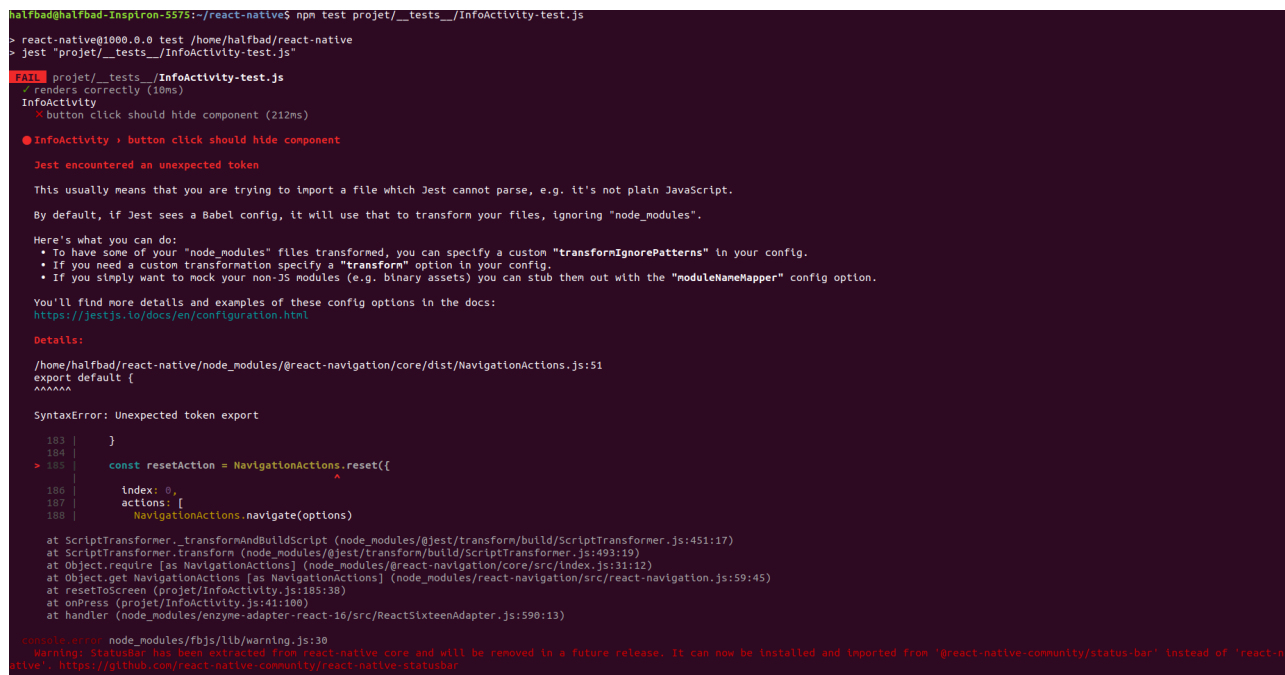


Image: Capture d'écran du terminal après la commande *npm test projet/___tests___/InfoActivity-test.js* exécutant jest

Cela conclut mon expérience de Jest jusqu'à maintenant.

Je me suis donc dirigé vers Mocha, un autre cadre de test Javascript afin de réaliser des tests unitaires.

III – Mocha and Chai

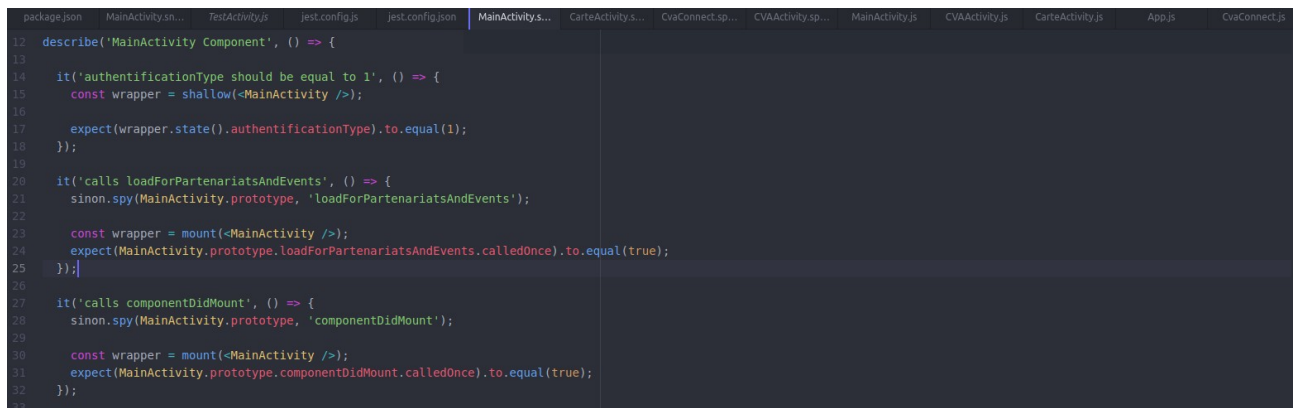
Cf: <https://mochajs.org/> and <https://www.chaijs.com/>

Après mon échec avec Jest, j'ai donc cherché une nouvelle façon de tester l'application GC et j'ai ainsi découvert Mocha, un autre cadre de test Javascript fonctionnant en combinaison avec Chai, une librairie dédiée au test Javascript.

J'ai ainsi pu réaliser de premiers tests unitaires en utilisant principalement les deux sites suivants:

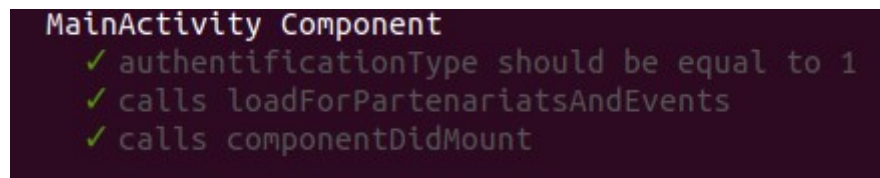
- <https://www.robinwieruch.de/react-testing-tutorial/#react-mocha-chai-test-setup>
- <https://www.robinwieruch.de/react-testing-tutorial/#react-mocha-chai-test-setup>

Ainsi, j'ai pu, avec le fichier de test suivant, vérifier la valeur initiale d'un paramètre et la réalisation d'appels de fonction:



```
12 describe('MainActivity Component', () => {
13
14   it('authenticationType should be equal to 1', () => {
15     const wrapper = shallow(<MainActivity />);
16
17     expect(wrapper.state().authenticationType).to.equal(1);
18   });
19
20   it('calls loadForPartenariatsAndEvents', () => {
21     sinon.spy(MainActivity.prototype, 'loadForPartenariatsAndEvents');
22
23     const wrapper = mount(<MainActivity />);
24     expect(MainActivity.prototype.loadForPartenariatsAndEvents.calledOnce).to.equal(true);
25   });
26
27   it('calls componentDidMount', () => {
28     sinon.spy(MainActivity.prototype, 'componentDidMount');
29
30     const wrapper = mount(<MainActivity />);
31     expect(MainActivity.prototype.componentDidMount.calledOnce).to.equal(true);
32   });
33 });
```

Image: Capture d'écran du fichier test, MainActivity.spec.js testant le composant MainActivity



```
MainActivity Component
✓ authenticationType should be equal to 1
✓ calls loadForPartenariatsAndEvents
✓ calls componentDidMount
```

Image: Capture d'écran de la partie du terminal correspondant au composant MainActivity après la commande `npm run test:unit` qui exécute mocha

Comme on peut le voir sur la deuxième image, les différents tests passent avec succès.

Cependant, de nouveau, on obtient de le problème des composants utilisant `this.props.navigation.state` qui renvoie `undefined` dans leur méthode `render()` et on se retrouve ainsi avec l'impossibilité de créer correctement le composant à tester dans le fichier test. Cela se traduit par une sortie dans le terminal comme on peut le voir dans l'image suivante:

```

2) <CarteActivity />
   calls componentDidMount:
     TypeError: Cannot read property 'state' of undefined
       at CarteActivity.state [as render] (src/Activity/CVA/CarteActivity.js:52:52)
       at finishClassComponent (node_modules/react-dom/cjs/react-dom.development.js:14741:31)
       at updateClassComponent (node_modules/react-dom/cjs/react-dom.development.js:14696:24)
       at beginWork (node_modules/react-dom/cjs/react-dom.development.js:15644:16)
       at performUnitOfWork (node_modules/react-dom/cjs/react-dom.development.js:19312:12)
       at workLoop (node_modules/react-dom/cjs/react-dom.development.js:19352:24)
       at renderRoot (node_modules/react-dom/cjs/react-dom.development.js:19435:7)
       at performWorkOnRoot (node_modules/react-dom/cjs/react-dom.development.js:20342:7)
       at requestWork (node_modules/react-dom/cjs/react-dom.development.js:20090:7)
       at scheduleWork (node_modules/react-dom/cjs/react-dom.development.js:19911:5)
       at scheduleRootUpdate (node_modules/react-dom/cjs/react-dom.development.js:20572:3)
       at updateContainerAtExpirationTime (node_modules/react-dom/cjs/react-dom.development.js:20600:10)
       at updateContainer (node_modules/react-dom/cjs/react-dom.development.js:20657:10)
       at ReactRoot.render (node_modules/react-dom/cjs/react-dom.development.js:20953:3)
       at /home/halfbad/ProjetGC/node_modules/react-dom/cjs/react-dom.development.js:21090:14
       at unbatchedUpdates (node_modules/react-dom/cjs/react-dom.development.js:20454:14)
       at legacyRenderSubtreeIntoContainer (node_modules/react-dom/cjs/react-dom.development.js:21086:5)
       at Object.render (node_modules/react-dom/cjs/react-dom.development.js:21155:12)
       at render (node_modules/enzyme-adapter-react-16/src/ReactSixteenAdapter.js:378:26)
       at fn (node_modules/enzyme-adapter-react-16/src/ReactSixteenAdapter.js:305:37)
       at Object.batchedUpdates$1 [as unstable_batchedUpdates] (node_modules/react-dom/cjs/react-dom.development.js:20439:12)
       at Object.act (node_modules/react-dom/cjs/react-dom-test-utils.development.js:1161:27)
       at act (node_modules/enzyme-adapter-react-16/src/ReactSixteenAdapter.js:305:13)
       at Object.wrapAct [as render] (node_modules/enzyme-adapter-react-16/src/ReactSixteenAdapter.js:364:16)
       at new ReactWrapper (node_modules/enzyme/build/ReactWrapper.js:134:16)
       at mount (node_modules/enzyme/build/mount.js:21:10)
       at Context.mount (src/Activity/CVA/CarteActivity.spec.js:11:21)

```

Image: Capture d'écran de la partie du terminal correspondant à CarteActivity après la commande `npm run test:unit` qui exécute mocha

J'ai donc essayé de réutiliser la méthode vue pour Jest, c'est-à-dire de créer une *const navigation* à l'aide de `jest.fn()`:

```

package.json  MainActivity.spec.js  TestActivity.js  jest.config.js  jest.config.json  MainActivity.spec.js  CarteActivity.spec.js  CvaConnect.spec.js  CVAActivity.spec.js  MainActivity.js  CVAActivity.js  CarteActivity.js  App.js  CvaConnect.js
1  import React from 'react';
2  import CarteActivity from './CarteActivity';
3
4  import renderer from 'react-test-renderer'
5
6  describe("<CarteActivity />", () => {
7
8    it('calls componentDidMount', () => {
9      sinon.spy(CarteActivity.prototype, 'componentDidMount');
10     const navigation = { navigate: jest.fn() };
11     const wrapper = mount(<CarteActivity navigation={navigation}/>);
12     expect(CarteActivity.prototype.componentDidMount.calledOnce).to.equal(true);
13   });
14
15

```

Image: Capture d'écran du fichier test, `CarteActivity.spec.js` testant le composant `CarteActivity`

Cependant, j'obtiens alors l'erreur suivante m'informant que `jest` n'est pas défini:

```

2) <CarteActivity />
   calls componentDidMount:
     ReferenceError: jest is not defined
       at Context.jest (src/Activity/CVA/CarteActivity.spec.js:10:36)

```

Image: Capture d'écran de la partie du terminal correspondant à CarteActivity après la commande `npm run test:unit` qui exécute mocha

J'ai, de nouveau, tenté de tester les clics en suivant la méthode proposée à cette adresse: <https://www.tangentsolutions.co.za/2017/07/03/unit-testing-react-native-with-mocha-chai-sinon-enzyme-and-istanbul/>. Cependant, je n'ai pas eu plus de succès que la fois précédente. Je ne peux pas montrer le code ou la sortie terminal, le code ayant été perdu lors d'un mauvais merge.

Cela conclut mon expérience de test Javascript avec Mocha.

IV – Conclusion

En conclusion, outre la proportion du temps passé à configurer plutôt qu'à écrire des tests, l'utilisation de react-navigation ainsi que l'échec d'utiliser Jest dans un fichier de test Mocha m'empêchent de pousser plus loin le test de l'application GC.

Le temps investi n'étant pas à la hauteur des résultats attendus et les difficultés consécutives rencontrées étant un trop grand frein à l'avancement du projet, je prends la décision d'abandonner le test Javascript de l'application .

Je vais ainsi me rediriger vers la programmation de fonctionnalités de l'application, ce qui me permettra de rendre utiles les heures de travail investies.