Benjamin Mitton, Cédric Pinard, M1 MIASHS 2020/2021

Programmation de composants Angular pour la mise en œuvre de tests utilisateurs

Cahier des charges

Mitton, Benjamin. (Photomonteur). (2020, 05 12). Dom&Angular[image numérique]. Images récupérées sur https://pixabay.com/fr/

Table des matières

Introduction

Ce présent cahier des charges a pour but de fixer les objectifs ainsi que de déterminer les contraintes possibles face à l'avancement, du projet de travail encadré de recherche (TER) réalisé par Benjamin Mitton et Cédric Pinard dans le cadre de l'enseignement de la première année de Master Mathématiques et informatique appliquées aux sciences humaines et sociales (MIASHS), parcours Web, informatique et connaissances (WIC), au sein de l'Université Grenoble Alpes (UGA).
Ce projet se déroule sur une année, à raison d'un jour par semaine du 02/11/2020 au 02/05/2021 puis d'une période à temps plein du 03/05/2021 au 23/06/2021. Le projet TER porte sur le sujet suivant, Programmation de composants Angular pour la mise en œuvre de tests utilisateurs, ce dernier est proposé et encadré par Alexandre Demeure, enseignant chercheur au laboratoire IIHM.

Contexte

Le domaine dans lequel s'inscrit ce projet est la domotique et plus globalement « L'internet of things » . Les maisons connectées devraient être toujours plus présentes dans le futur, il est donc nécessaire de fournir à ces utilisateurs des outils adaptés à leurs capacités. Hors dans un but de permettre aux utilisateurs de modifier à leur souhait le fonctionnement et les règles régissant les appareils de leur maison, il est nécessaire de leur fournir un langage de programmation accessible et le plus intuitif possible. C'est sur cet enjeu que s'est construit le projet du laboratoire.

Historique

Notre projet TER s'inscrit dans une démarche de développement de tests utilisateurs en ligne, par le laboratoire IIHM. Ce dernier a réalisé en 2017 deux études sur la création d'un nouveau langage de programmation (le CCBL) qui a pour but de permettre une programmation plus simple et intuitive des environnements de domotique domestique. C'est dans le but de continuer les expérimentations sur ce langage qu'est venu la nécessité de créer des tests utilisateurs en ligne pour ce produit. Le but du projet TER est de développer des composants Angular, qui seront intégrés ou réutilisés dans la mise œuvre de ces tests utilisateurs.

Description de la demande

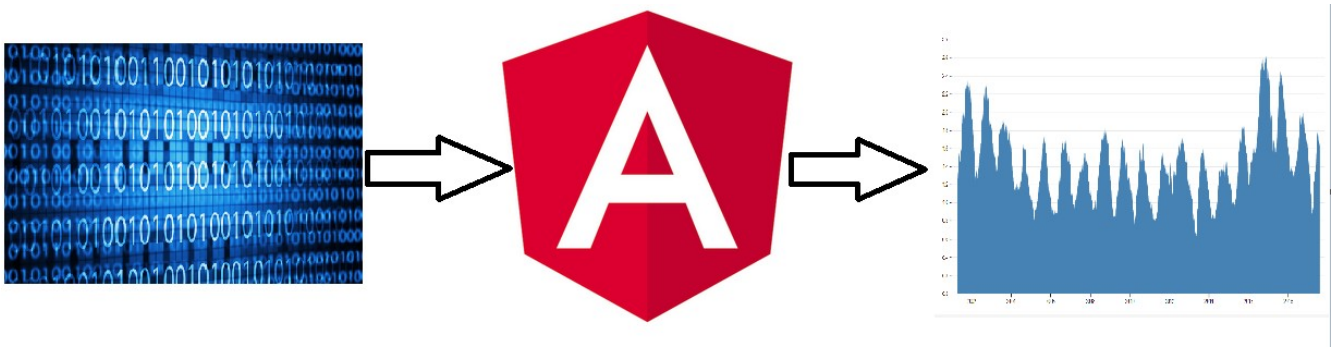
Les objectifs

L'objectif principal de ce projet est de permettre à la plate-forme de test du CCBL d'utiliser des linecharts (graphiques en ligne) afin d'afficher les données produites par les acteurs (capteurs et effecteurs) de l'environnement numérique.

L'objectif secondaire est de créer un produit utilisable non seulement par le laboratoire mais aussi par la communauté utilisant Angular, en publiant le produit de notre projet sur npm (le gestionnaire de paquets officiel de Node.js) et GitHub.

Le produit du projet

Le produit sera une librairie Angular, permettant la création d'une linechart paramétrable par l'utilisateur. Le produit est destiné à des développeurs voulant intégrer une linechart dans leur projet Angular. Cette librairie devras permettre la création d'un composant prenant des données d'un certain type en paramètre d'entrée et affichant un aperçu visuel de ces données sous la forme d'une linecharts. D'autres paramètres seront pris en compte afin que l'utilisateur puisse la personnaliser à sa guise.

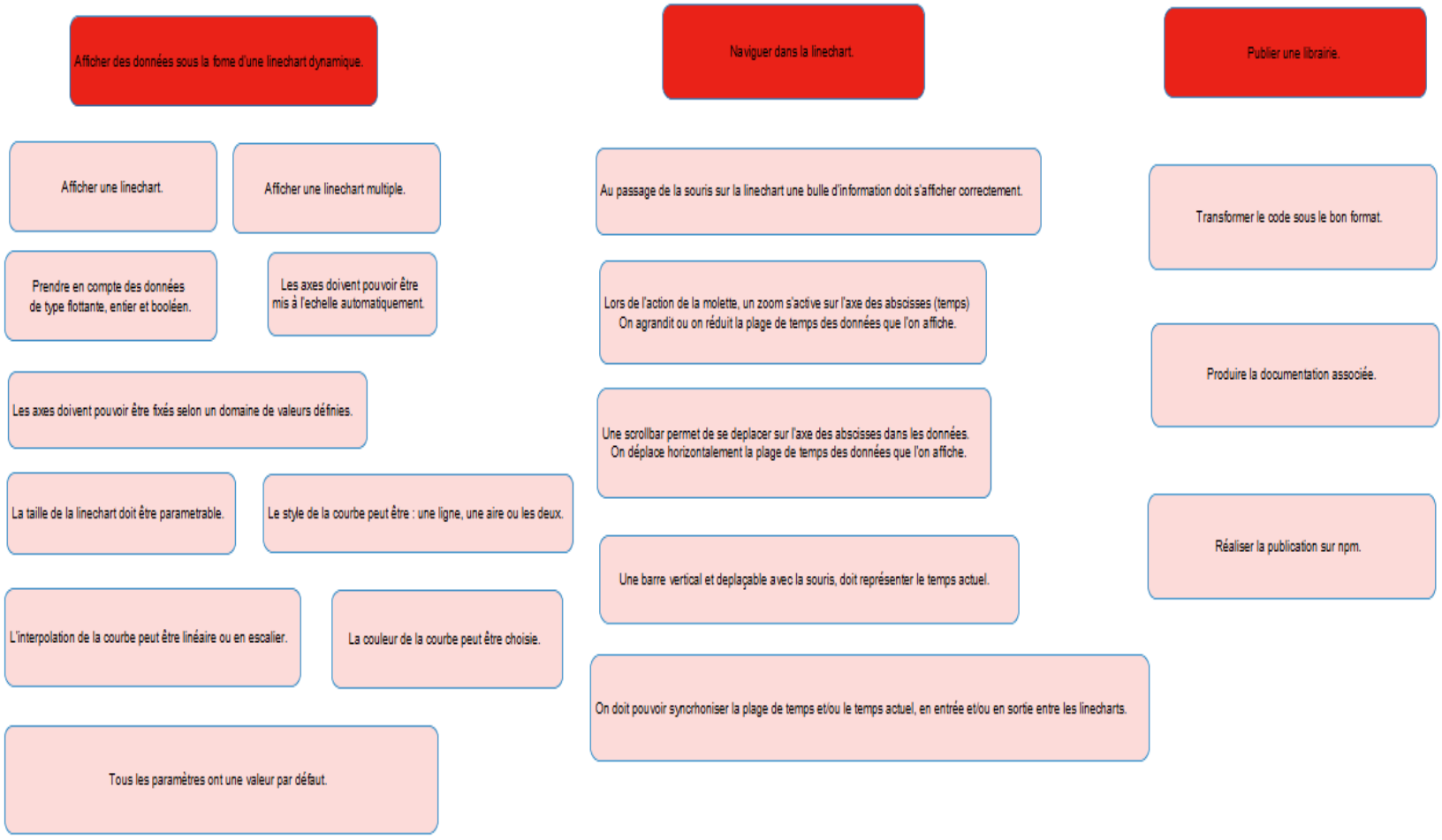


Les fonction du produits

On peut découper les fonctionnalités de notre produits en trois grande catégorie. Premièrement l'affichage de la linechart comme un élément visuel d'un site web, deuxièmement la navigation en son sein et finalement la publication de la librairie. Toutes les fonctions autour de l'affichage correspondent à la production de la linechart en elle même et sont les plus faciles à réaliser. Les fonctions rentrant dans la catégorie de la navigation pourraient être séparer à nouveau en deux autres catégories, les fonctions ne dépendant que d'une seule linechart et celles penser pour que la linechart puisse s'intégrer et se synchroniser avec d'autres composants (par exemple d'autres linecharts) au sein d'un plus grand projet. Par exemple la barre verticale représentant le temps actuel, est un élément du composant qui ne prend du sens que lorsque cet élément est synchronisé avec d'autres composants utilisant la

même notion de temps actuel. Cela permettrait de le représenter visuellement sur les données. Finalement les fonctions liées à la publication de la librairie sont les plus diverses puisqu'il y a d'une part des documents à produire et d'autres part du code à développer et à agencer sur un format de librairie déployable.

L'ensemble des fonctionnalités proposées sont disponibles dans le schéma suivant :



Ce schéma est disponible dans l'archive du projet et dans dépôt Git, dans un format plus lisible.

Justification

Afficher une linechart :
La fonctionnalité la plus évidente du projet est celle d'afficher les données sous forme d'une courbe de données dans un composant comportant des axes.

Afficher une linechart multiple :
Une linechart multiple est un composant d'affichage de données comportant plusieurs courbes de données permettant ainsi une comparaison plus simple des données.

Prendre en compte des données de type flottante, entier et booléen :
Les données qui peuvent être transmises et comprises par le composant sont de ces trois types tout autres types de données n'est pas à prendre en compte.

Les axes doivent pouvoir être mis à l'échelle automatiquement :
Si l'on souhaite simplement transmettre ces données et non un domaine d'affichage il faut que le composant calcule automatiquement la plage d'affichage des valeurs.

Les axes doivent pouvoir être fixés selon un domaine de valeurs définies :
Il doit aussi être possible de transmettre une plage d'affichage des valeurs pour simplifier les comparaisons entre deux courbes de données.

La taille de la linechart doit être paramétrable :
Le composant doit pouvoir s'intégrer au sein de projet plus grand et donc l'utilisateur doit pouvoir transmettre la taille qu'il souhaite pour son composant d'affichage.

Le style de la courbe peut être : une ligne, une aire ou les deux :
Visuellement le choix d'une simple ligne ou d'une aire peut être intéressant il est donc nécessaire que l'utilisateur puisse choisir la façon d'afficher ses valeurs.

L'interpolation de la courbe peut être linéaire ou en escalier :
L'interpolation linéaire de la courbe permet d'avoir un graphique plus lisse sur des données flottante tandis que l'interpolation en escalier permet de garder les valeurs des données à tous moments pour tous les types de données.

La couleur de la courbe peut être choisie :
Fonctionnalité essentiellement esthétique mais peut aussi permettre de différencier les courbes entres elles.

Tous les paramètres ont une valeur par défaut :
Si un utilisateur oublie un paramètre ou ne souhaite pas le préciser, il est nécessaire qu'une configuration par défaut soit disponible.

Au passage de la souris sur la linechart une bulle d'information doit s'afficher correctement :
Pour assurer une meilleure exploration des données, une bulle d'information sur la position de la souris par rapport aux données est nécessaire.

Lors de l'action de la molette, un zoom s'active sur l'axe des abscisses (temps), on agrandit ou on réduit la plage de temps des données que l'on affiche :
Pour assurer une meilleure précision dans l'exploration des données, la possibilité de zoomer au sein des données est nécessaire.

Une scrollbar horizontale permet de se déplacer sur l'axe des abscisses, on déplace la plage de temps des données que l'on affiche :
Pour ne pas avoir à dézoomer et zoomer à chaque fois que l'on souhaite observer une autre plage de temps des données, une barre de défilement horizontale est nécessaire.

Une barre vertical représentant le temps actuel, déplaçable avec la souris ou avec des paramètres (input) :
Les données affichées correspondent à une plage de temps dans le cas d'une simulation. Il est nécessaire de créer une barre vertical correspondant au temps actuel, déplaçable dynamiquement.

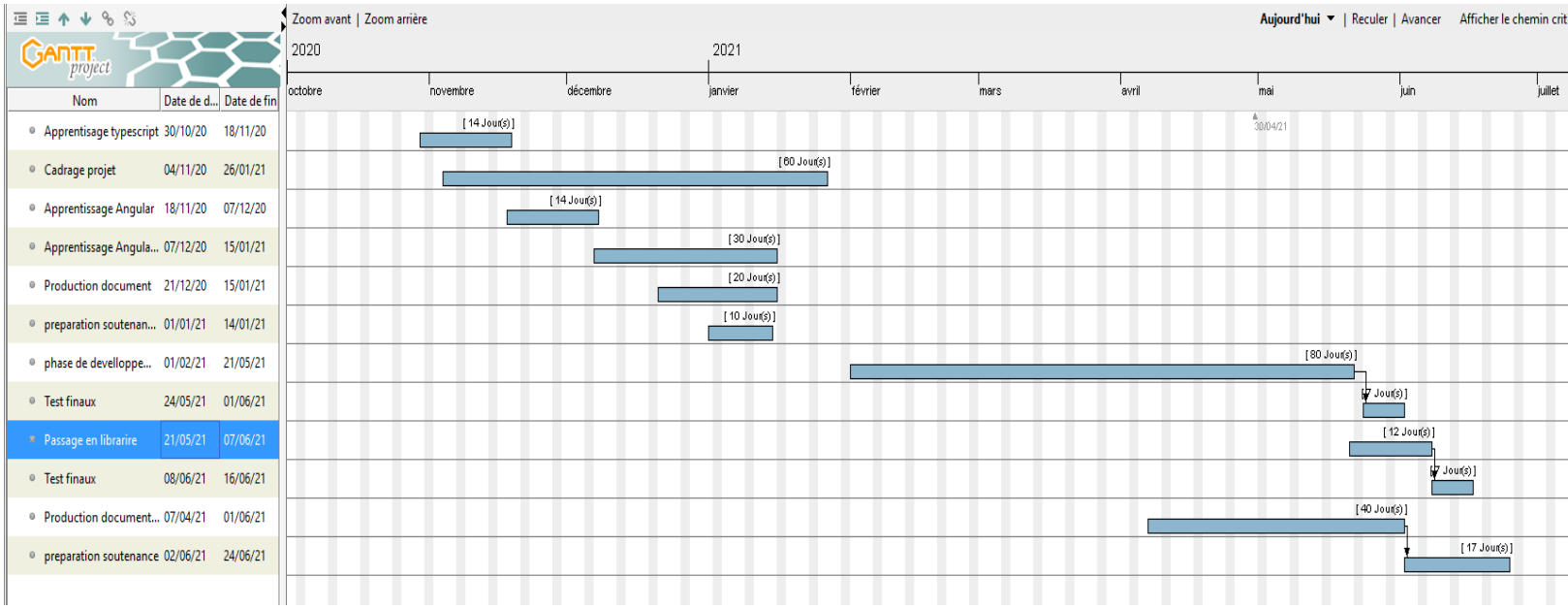
On doit pouvoir synchroniser la plage de temps et le temps actuel, en entrée et en sortie entre les linecharts et d'autres éventuels composants :
La synchronisation des informations, dans une optique d'intégration à un plus grand projet, permet de créer des interactions entre les différents composants et une plus grande diversité de possibilité d'utilisation.

Contraintes

Pour la réalisation de ce projet nous devons faire face à plusieurs contraintes. La première est l'absence de budget pour financer notre projet, en effet nous ne percevrons aucune rémunération pour la réalisation de notre projet TER et aucun investirent n'est disponible pour du matériel ou quelque autre raison. La deuxième est une contrainte temporelle, ce projet doit être terminé avant le 18/06/2021. La dernière est le format du rendu des documents, de la production d'une soutenance orale et de la librairie Angular. Le format d'une librairie Angular est bien spécifique de part son format de développement et de part la documentation qui doit lui être associée lors de sa publication.

Déroulement du projet

Planification



Voici le diagramme de Gantt estimé disponible dans l'archive du projet et dans le dépôt Git dans un format plus lisible. On pourrait découper moins précisément ce projet en 3 phase :

La première, serait la phase d'apprentissage du framework Angular et de cadrage du projet. Angular est une technologie qui nous est inconnue et que nous devons utiliser durant ce projet. Cette phase s'étend donc du début du projet jusqu'aux vacances de Noël.

La suivante, serait la phase de développement et de publication de la librairie. Le diagramme ne représente pas correctement cette phase, car nous allons fonctionner autour de sprint sur des prototypes directement validé, à chaque itération, par le client. On peut néanmoins utiliser

le diagramme pour avoir les deadlines liées au développement, un dépassement de ces dernières nous indiquerait un retard sur le projet.

La dernière, serait la phase de production des documents et des rendus nécessaires dans le cadre de l'unité d'enseignement (UE), dans laquelle s'intègre le projet

Organisation

L'organisation du projet dépend des phases de ce même projet, lors de la phase d'apprentissage et de cadrage du projet les réunions sont fixées toutes les deux semaines pour discuter et construire le projet, mais le travail d'apprentissage d'Angular n'est déterminer par aucune organisation. Le but est que chaque étudiant du groupe engrange le plus de connaissance sur le sujet mais il est impossible de déterminer les notions à apprendre. Par ailleurs cet apprentissage ce poursuivra tout au long du projet.

C'est lors de la phase de développement que l'on peut observer une organisation, l'équipe fonctionne en interne grâce à un tableau kanban mis à jour quotidiennement lors d'une réunion de 15 minutes le matin. Cette réunion a pour but de déterminer toutes les tâches à effectuer (développement, correction de bug, rédaction de documents), ainsi que les personnes en charge de ces dernières.
Ce tableau est de la forme suivante :

Semaine	Jour	Objectfis	Etat
	1	parametre les composants	En cours
		Ajout fonction	Fait
		creer composant boolean	En cours
		mettre en place le git	Fait
		rediger les cr de reunion	Nul
		mettre a jour backlog	Fait
		repondre alexandre	Fait
		reunion alexandre	Fait

Le backlog fonctionnel est réalisé hebdomadairement avec le client. Le projet fonctionne sur des sprints d'une semaine aboutissant par une réunion avec le client. Chaque réunion commence par une présentation du travail effectué par l'équipe au client, lors de laquelle, le client peut faire part de ses remarques ou de ses suggestions d'améliorations. Ensuite le client nous fait part des nouvelles fonctionnalités qu'il souhaite ajouter à la librairie. Avant la fin de la réunion, l'équipe a donc un nouveau backlog fonctionnel, contenant les fonctions à ajouter et/ou les bugs à corriger, pour la prochaine réunion. Ainsi, l'équipe peut estimer le temps nécessaire à la réalisation de ces tâches, afin de fixer la date de la prochaine réunion pour laquelle, l'ensemble des tâches devra être terminé.