

Plan de développement

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

MITTON BENJAMIN , PINARD CEDRIC

UNIVERSITÉ GRENOBLE ALPES

Sommaire

Sommaire.....	2
1. Introduction	3
1.1. Objectifs	3
1.2. Documents de référence.....	3
2. Concepts de base.....	3
3. Organisation.....	4
3.1. Décomposition en tâches.....	4
4. Planification.....	5
5. Méthodes et outils.....	5
6. Qualité.....	6
7. Références.....	6

1.Introduction

En 2017 l'IIHM produit une étude sur un langage de programmation plus approprié pour les utilisateurs de maison connecté. Ils ont besoin de tester leur langage plus efficacement et décide donc de démarrer la création de batterie de test en ligne pour simplifié les test utilisateurs. L'idée est la suivante l'utilisateur se retrouve face a une simulation d'environnement connecté et peu interagir avec différents composants de l'interface afin de vérifier si ce qu'il a écrit avec le nouveau langage correspond à ce qu'il souhaitais (cf référence 2) Le développement de ce projet a commencé en 2019, Une partie du contenu a donc déjà été produit. En 2019-2020 a été les réaliser les différents composants de l'interfaces de test, tel que les personnages ou la température des pièces et encore la météo. Ce qui signifie que ce projet s'inclue donc dans un chantier plus grand.

1.1. Objectifs

La phase de 2020-2021 a pour but de compléter le banc virtuel de test en y ajoutant le composant de timeline ainsi que les fonctionnalités nécessaires a son déploiement. Ce composant doit permettre de visualiser une démonstration de ce que peu faire le module de test en modifiant chacun des éléments par rapports a un scénario et a un temps donné. Dans un deuxième temps le but est que l'utilisateur puisse créer lui même ces scénario pour pouvoir les observer.

1.2. Documents de référence

Comme document de référence nous utiliserons donc le code précédemment produit mais pour lesquels aucun document technique n'as été créer et donc fournit par le client. (cf annexe 1)

2.Concepts de base

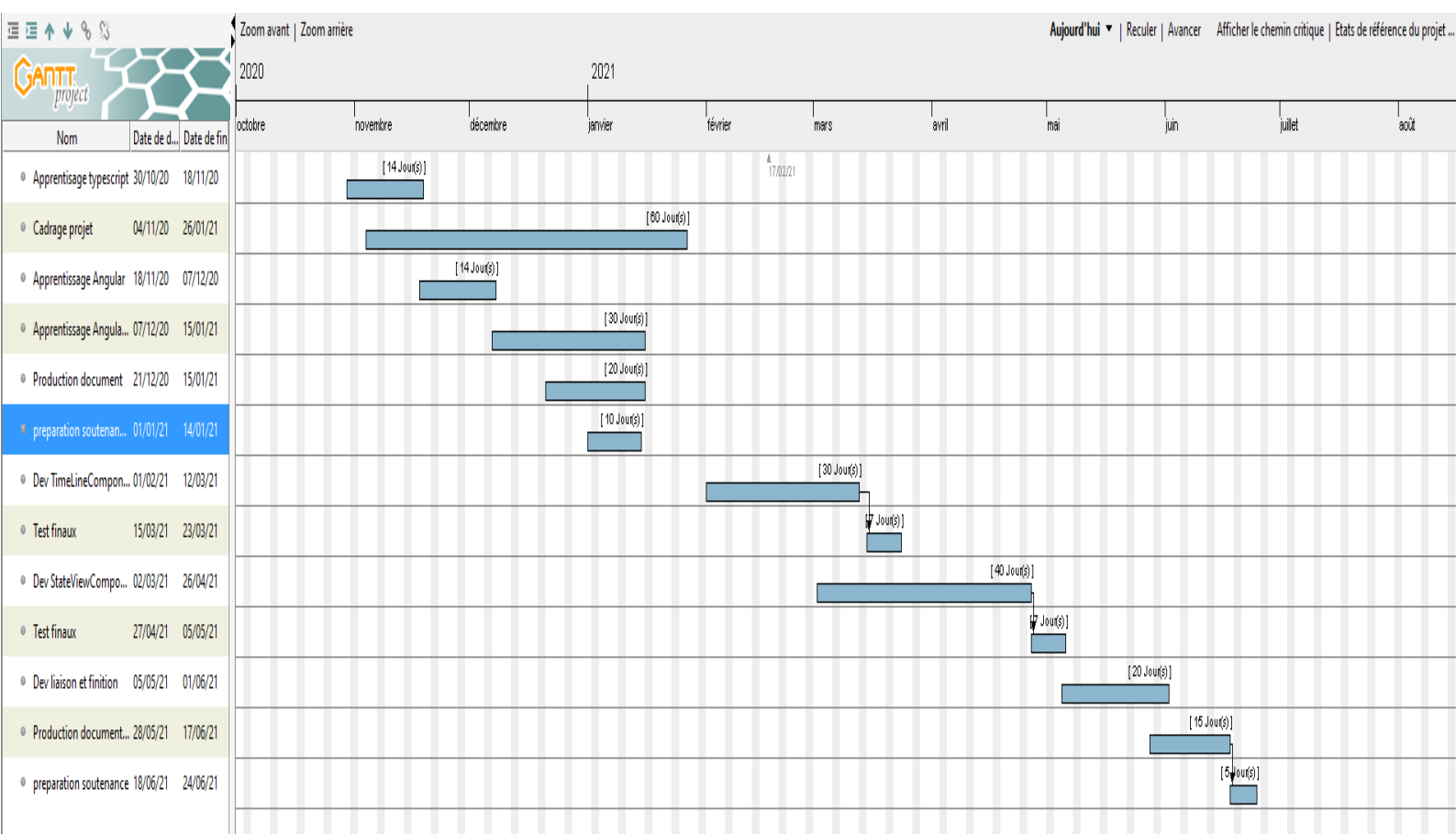
La programmation sous angular js repose sur les concepts de composants. Un composant est une partie d'une interface web qui possède son propre visuel et ces fonctions. Grâce à ce découpage des éléments d'une page on découvre le principe de single Page Application qui nous permet de ne charger qu'une seule page et de la modifier en fonction des actions de l'utilisateur.

3. Organisation

3.1. Décomposition en tâches

		type de retour	temps/personne	variable	interface/librairie liée	
barreComponent.ts	avancer(i : number)	void	3h			
	activer()	void	3h			
	pause()	void	3h			
	Reinitialise()	void	1h			
barreComponent.css	css		1h			
barreComponent.html	html		1h			
butonPauseComponent.html	html		1h			Librairie : rxjs
butonPauseComponent.css	css		1h			
butonPauseComponent.ts	mettreenpause()	void	3h			interface: scenario events
butonLireComponent.html	html		1h			
butonLireComponent.css	css		1h			
butonLireComponent.ts	Activer()	void	3h			
TimeLineComponent.html	html		4h			
TimeLineComponent.css	css		1h			
TimeLineComponent.ts	ngOnDestroy()	void	1h			
	ngOnInit()	void	1h			
	lireScenario()	void	3h			
StateViewComponent.html	html		4h			
StateViewComponent.css	css		1h			
StateViewComponent.ts	actualise()	void	2h			
	ngOnInit()	void	1h			
	ngOnDestroy()	void	1h			
StateService	set	void	1h			
	get	any	1h	string etat (lecture , pause)		
	emitStateSubject()	void	1h	scenario scenario		
	ClearEvent()	void	1h			
	ExecuteEvent()	void	2h			
TimeService	clearTime()	void	1h	Number : time		
	ngOnInit()	void	1h			
	ngOnDestroy()	void	1h			
	getTime()	number	1h			

4. Planification



5. Méthodes et outils

Pour produire le notre code nous utiliserons comme environnement de développement le logiciel vs Code de Microsoft. Non seulement il structure facilement les composants mais en plus certaines de ces fonctionnalités comme les partages en live de document sont très intéressants. Quant aux aux technologies utilisées nous avons donc comme base et contrainte du projet le framework web Angular JS et la librairie rxjs qui nous permet d'avoir plus de contrôles sur notre interface web.

6. Qualité

Les critères déterminants la qualité sont les suivants :

- L'affordance de nos composants
- La facilité de déploiement sur d'autres modules
- Le nombre d'erreurs
- Le rendement (que le coût en puissance de calcul ne soit pas trop élevé)

7. Références

- 1 D. (2020, 2 novembre). AlexDmr/ccbl-gfx-angular-dev. Consulté à l'adresse <https://github.com/AlexDmr/ccbl-gfx-angular-dev/tree/master/src/app/scene-house>
- 2 Lénaïc Terrier, Alexandre Demeure and Sybille Caffiau. 2017. CCBL: A Language for Better Supporting Context Centered Programming in the Smart Home. PACMHCI / EICS'17 Paper, 1, 1, Article 13, 2017. 18 pages. DOI: <http://dx.doi.org/10.1145/3099584>