

## **Imitation**

En repensant à mon expérience d'apprentissage d'Android et d'iOS, je me rends compte qu'une grande partie des connaissances les plus précieuses que j'ai acquises proviennent de l'imitation.

Comment imiter ? Ouvrez deux projets. Par exemple, pour imiter l'implémentation du fil d'actualité de Jack, je regarde deux lignes de son code, puis j'écris son code dans mon projet. Si je ne comprends pas, je retourne voir, et si je comprends, j'écris selon mes propres idées, en écrivant plus d'un coup.

Grâce à cette méthode, je me souviens très bien avoir appris à écrire moi-même un flux de type "waterfall" avec rafraîchissement par glissement (pull-to-refresh) sur Android, ainsi qu'à créer des interfaces utilisateur (UI) manuellement sur iOS, entre autres.

La fonctionnalité de rafraîchissement par glissement vers le bas (pull-to-refresh) sur Android m'a permis, pour la première fois, de contrôler directement une vue sans passer par le XML, et de comprendre les gestes. Cela a été une véritable amélioration. Parfois, en copiant le code de l'auteur original, j'ai eu des révélations. Certaines choses ne sont pas évidentes à première vue, peu importe à quel point on réfléchit. En effet, en lisant, on néglige souvent des éléments importants, mais en écrivant le code et en manipulant manuellement, on s'implique davantage, on se pose plus de questions, on remarque plus de détails, et finalement, on comprend mieux. Comme pour l'effet de parallaxe dont j'ai parlé dans mon précédent article. Au début, je ne savais pas que l'image de fond se déplacerait horizontalement ou verticalement, mais en reproduisant le code, j'ai remarqué beaucoup de choses et j'ai appris une méthode pour créer un effet de flou (blur).

Auparavant, j'utilisais beaucoup les xib pour iOS. J'ai essayé plusieurs fois d'écrire du code à la main, mais j'ai fini par abandonner. En imitant un projet de fil d'actualité, c'était comme si je copiais tout simplement. La plupart du temps, à part quelques différences dans les noms, tout était identique. Cependant, j'ai quand même remarqué plusieurs aspects importants de l'écriture manuelle de code. Par exemple, pour une cellule de liste, si la hauteur varie en fonction du contenu, vous devez calculer sa hauteur avant de la dessiner, puis la transmettre à la tableview. Comment calculer la hauteur ? Une méthode consiste à utiliser une largeur fixe pour calculer la hauteur. Ces détails, on les remarque en copiant le code.

Lorsque nous imitons, nous avons tendance à nous impliquer davantage. En ce qui me concerne, lire du code me donne souvent envie de dormir. D'une part, parce que mon cerveau se fatigue, et d'autre part, parce que tout mon corps est au repos, ce qui facilite l'entrée dans un état de détente. Un de mes amis, lorsqu'il lit du code, prend des notes. Sur Zhihu, quelqu'un a

mentionné qu'il résolvait des problèmes en lisant du code pour se familiariser avec un projet. Tout cela est très bien, l'essentiel est de s'impliquer, pas seulement de regarder. Bien sûr, certaines personnes peuvent simplement regarder et réfléchir pendant longtemps.

Imiter et créer par soi-même sont deux choses différentes. Imiter est plus facile, car il s'agit simplement de reproduire, voire de copier le code tel quel. Le processus de création, en revanche, est bien plus difficile. Souvent, imiter une application existante est plus simple, car on peut facilement calculer les frames en se basant sur la disposition de l'interface. Comme l'imitation est relativement plus facile, elle peut être maintenue plus longtemps, sans risque d'abandonner face à la difficulté. Cependant, les connaissances acquises par l'imitation peuvent parfois être plus riches que celles obtenues par la création.

L'imitation permet de réduire les erreurs d'essai et d'acquérir plus rapidement les connaissances correctes. Lorsqu'on apprend la programmation, on peut travailler sur des projets tout en cherchant sur Google. Cependant, cela peut entraîner de nombreuses erreurs d'essai, où l'on explore une chose ici, une autre là, et souvent, ces méthodes ne sont pas très adaptées, surtout pour les débutants. De plus, beaucoup de ces erreurs d'essai sont sans signification. Parfois, il suffit de maîtriser les bonnes connaissances et de comprendre pourquoi elles sont correctes. Souvent, lors des essais, on essaie au hasard, par exemple, pour le format HTML, si la mise en page est incorrecte, on modifie ceci, on ajuste cela, et si on réussit, on rencontre le même problème la prochaine fois. En imitant, puisque la bonne réponse se trouve dans le code des autres, on peut s'y référer. La bonne réponse est à portée de main. Ainsi, on acquiert rapidement les bonnes connaissances. Par exemple, pour le format HTML, je peux trouver des projets avec des formats HTML et copier ligne par ligne le code des autres, tout en réfléchissant pendant que je copie, ce qui pourrait me permettre d'apprendre plus vite et mieux que si je bricolais tout seul.

Imitez-le !