

Fiche de lecture : IDEvelopar : A programming Interface to enhance code understanding in Augmented reality.

Matthys Gaillard

7 décembre 2023

1 Pourquoi ce choix ?

J'ai choisi ce document [Kreber *et al.*, 2022], car il ne proposait pas en tant que tel un nouveau paradigme pour comprendre les bases de code, mais plutôt un nouvel environnement pour coder et de par là, augmenter nos chances de comprendre le code produit. Cet environnement est un environnement en réalité augmentée. C'est-à-dire que les éléments sont ajoutés, superposés à la réalité. De plus, il était interactif contrairement aux autres articles déjà recherchés. Donc, une action entreprise par l'utilisateur via le casque se répercutait réellement sur le code source.

2 Analyse du document

2.1 Contexte

Les chercheurs ont remarqué que les développeurs passaient un temps considérable sur différentes tâches durant le processus de développement comme la navigation, l'identification d'éléments¹ importants dans le code. De base, cette compréhension s'effectue à travers un environnement de développement intégré comme la suite de JetBrains. Malgré qu'ils permettent d'entreprendre énormément de choses liées au code, ils ne sont pas prévus pour faciliter la compréhension de celui-ci une fois écrit.

C'est pourquoi et après avoir analysé différents paradigmes de visualisation de code comme Code-Bubble, ils ont décidé d'aller dans la direction d'être dans un environnement entièrement visuel et d'écrire leur application de compréhension de code : IDEvelopar.

2.2 Objectifs

L'objectif est ici de montrer que le fait de ne plus avoir de restriction dans l'espace de codage et l'ajout d'outil de visualisation permis par l'utilisation de la VR et de la RA permet de bien mieux comprendre le code. Vu que l'objectif n'est pas de démontrer formellement la compréhension du code écrit, mais plutôt de montrer que l'environnement est plus propice à la compréhension, ils ont décidé de ne pas s'appuyer sur un paradigme particulier, mais plutôt sur un enchaînement cohérent des éléments de codage.

2.3 Méthode

Pour cela, ils ont développé un interface en réalité augmentée permettant de coder et d'agencer notre espace de codage sur l'espace infini que propose réalité augmentée. De plus, ils ont conçu l'application de telle sorte que l'on peut coder en direct et que si on clique sur un élément tel que l'appel d'une fonction ou d'une classe, on ait un historique de notre passage de l'endroit antérieur où nous nous trouvons.

Ils ont voulu également que l'outil puisse faire de A à Z tous les étapes du processus de création d'une application donc, ils ont préféré que le code soit hébergé sur un ordinateur et qu'en ouvrant IntelliJ et un plugin spécifique reliait l'ordinateur au casque. Cela permettait également de compiler le code directement sur l'ordinateur.

Pour mesurer leur étude, ils ont fait appel à 8 étudiants spécialisés en informatique. Aucun était familier avec l'AR. Ils ont dû réaliser 3 tâches différentes et répondre à des questionnaires. Ils ont récolté des données quantitatives² et qualitatives. Ils ont également testé sur eux-mêmes le système en utilisant le système cognition dimension framework.

1. Comme les codes smells

2. Ils ont utilisé les questionnaires : SUS, UEQ

2.4 Résultats

Les résultats qualitatifs ont montré que les étudiants ont eu une meilleur compréhension du code malgré les quelques problèmes d'utilisabilité de l'outil.

Références

[Kreber *et al.*, 2022] KREBER, L., DIEHL, S. et WEIL, P. (2022). Idevelopar : A programming interface to enhance code understanding in augmented reality. *In 2022 Working Conference on Software Visualization (VISSOFT)*, pages 87–95.