

Fiche de lecture : UML-based live programming

Matthys Gaillard

30 novembre 2023

1 Pourquoi ce choix ?

J'ai choisi ce document [Kučečka *et al.*, 2022], car il proposait un environnement entièrement en réalité virtuelle dans lequel tu pouvais à la fois coder et avoir une visualisation du code grâce à des diagrammes UML. De plus, à chaque modification et à la sauvegarde du code, le diagramme du code se mettait à jour automatiquement.

2 Analyse du document

2.1 Contexte

La compréhension du code et le développement de grandes applications sont des tâches difficiles et coûteuses en temps pour le développeur. La visualisation de code a pour but d'aider les développeur à mieux comprendre la structure de leur code, et ainsi de mieux naviguer à l'intérieur de celui-ci.

En matière de visualisation logiciel, la réalité virtuelle a déjà fait ses preuves, mais peu de recherches ont été faites en matière de programmation "instantanée".

2.2 Objectifs

Le but est donc ici de présenter un environnement de programmation en réalité virtuelle, dans lequel le développeur peut coder et visualiser son code en même temps. Toutes les étapes sont incorporés directement dans le logiciel, de la création du projet à sa compilation. Une fois, le code compilé, la visualisation se met directement à jour.

2.3 Méthode

Dans cette étude, les auteurs ont créé un environnement de programmation entièrement en réalité virtuelle. Cet environnement était divisé en 3 parties :

1. Le diagramme uml de l'application montrant les interactions
2. Le code accessible via une tablette virtuelle.
3. Un jeu illustrant l'application compilée.

Pour pouvoir tester cet environnement, ils ont fait appel à 20 étudiants en informatique. 16 sont issus du milieu universitaire et les 4 autres de l'industrie.

Pour prouver que leur environnement était efficace, ils ont fait tester le programme compilé (un jeu en 3 étapes) sur 2 environnements différents¹ et puis, il leur ont fait remplir 2 questionnaires².

2.4 Résultats

Les résultats montrent que l'environnement virtuel est bien mieux en terme d'utilisabilité et d'expérience utilisateur, mais on garde des problèmes inhérents à la VR. Un environnement classique reste plus approprié en terme de rapidité d'écriture de code.

1. VR IDE et Unity 3D

2. UEQ et Mann-Whitney

Références

[Kučečka *et al.*, 2022] KUČEČKA, J., VINCÚR, J., KAPEČ, P. et ČIČÁK, P. (2022). Uml-based live programming environment in virtual reality. pages 177–181.