



**HAL**  
open science

# Conception et évaluation d'une application Web de distributed pair programming pour les programmeurs novices

José Colin

► **To cite this version:**

José Colin. Conception et évaluation d'une application Web de distributed pair programming pour les programmeurs novices. 10èmes Rencontres Jeunes Chercheuses et Chercheurs en EIAH (RJC-EIAH 2024), ATIEF (Association des Technologies de l'Information pour l'Éducation et la Formation), Jun 2024, Laval, France. pp.96-109. hal-04682658

**HAL Id: hal-04682658**

**<https://hal.science/hal-04682658v1>**

Submitted on 30 Aug 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Conception et évaluation d'une application Web de distributed pair programming pour les programmeurs novices

José Colin, 1<sup>re</sup> année de thèse

IRIT, Université Toulouse 3 Paul Sabatier, Toulouse, France  
jose.colin@irit.fr

**Résumé** La recherche sur le *pair programming* (PP) dans l'enseignement a montré un certain nombre de résultats positifs pour les apprenants, particulièrement pour les programmeurs novices : une plus grande confiance dans la qualité du code et de meilleures compétences en matière de résolution de problèmes et de collaboration. En raison de ces divers avantages, le pair programming dans l'enseignement est de plus en plus utilisé. En outre, l'apprentissage hybride est de plus en plus répandu dans l'éducation, y compris pour l'apprentissage de la programmation. Par conséquent, le *distributed pair programming* (DPP) peut être considérée comme une bonne solution pour soutenir le pair programming dans des scénarios d'apprentissage hybride. Différentes propositions provenant à la fois de la recherche et des principaux éditeurs d'environnements de développement intégrés (IDE) mettent en oeuvre le DPP. Cependant, notre analyse de la littérature montre qu'aucune de ces propositions ne répond à l'ensemble des exigences du pair programming adapté aux novices dans des situations d'apprentissage hybride. Sur la base de ces résultats, nous présentons une nouvelle application de DPP spécialement conçue pour les novices. Elle intègre, à partir de certaines exigences identifiées dans la littérature, plusieurs fonctionnalités dédiées au DPP et à la collecte de données en vue de l'analyse de l'apprentissage. L'outil a été expérimenté en conditions réelles dans l'enseignement supérieur avec 82 étudiants, en PP et en DPP. L'expérimentation ne montre pas de différence significatives entre le PP et le DPP en ce qui concerne l'utilisabilité perçue de l'application par les étudiants, ainsi que la qualité de leurs productions.

**Keywords:** apprentissage de l'informatique, distributed pair programming, conception et évaluation d'outils, programmeurs novices

## 1 Introduction

Des travaux antérieurs ont montré de multiples impacts positifs du pair programming (PP) dans l'enseignement tels qu'une plus grande confiance dans la qualité du code, de meilleures compétences en résolution de problèmes [9], ainsi qu'une plus grande satisfaction perçue des étudiants [21]. Le PP est fréquemment étudié dans l'enseignement supérieur, mais des recherches menées sur des lycéens [16] et dans des cours d'introduction à l'informatique au niveau universitaire [31]

ont démontré des bienfaits similaires lorsqu'il est utilisé avec des programmeurs novices.

L'apprentissage hybride est aujourd'hui de plus en plus répandu dans l'éducation. Dans ce contexte, le distributed pair programming (DPP) permet aux étudiants de s'engager à distance dans des activités de pair programming. L'efficacité du DPP comparativement au PP et à la programmation en solo semble très prometteuse en ce qui concerne la qualité et la compréhension du code, ainsi que la performance académique, puisque des recherches n'ont observé aucune différence entre ces différentes pratiques de la programmation [22]. Cependant, en raison des défis soulevés par la composante *en ligne* de l'apprentissage hybride [19], les outils actuels dédiés au DPP souffrent de faiblesses affectant leur utilisabilité pour les étudiants et les enseignants, telles que le manque de certaines fonctionnalités de PP et le faible soutien pour la gestion pédagogique. Ils souffrent également de manques affectant les possibilités d'effectuer des travaux de recherche tels que la faible collecte de données empêchant le développement de *learning analytics*.

Dans cet article, nous présentons un nouvel outil pour le DPP et discutons les résultats d'une expérimentation menée dans l'enseignement supérieur sur des séances de travaux pratiques en Python. Nous explorons la capacité de l'outil à soutenir le DPP comparativement au PP classique, en analysant l'utilisabilité perçue par les étudiants et la qualité de leurs productions dans ces deux contextes. Enfin, nous présentons nos conclusions et les principales perspectives à court terme concernant l'évolution de l'outil.

## 2 État de l'art

Cette section identifie dans un premier temps les exigences d'une application de distributed pair programming dédiée aux programmeurs novices, puis s'intéresse ensuite aux principaux outils existants afin d'exposer une synthèse de ces outils au regard de l'ensemble des critères que nous avons identifiés.

### 2.1 Exigences pour le distributed pair programming

Selon Schenk et al. [24], le DPP se déroule dans un environnement virtuel où un éditeur partagé est fourni aux utilisateurs, afin que toutes les modifications du code source effectuées par un participant soient transférées sur l'écran de l'autre participant. L'environnement virtuel garantit également qu'un seul utilisateur peut modifier le code source à tout moment. Comme dans le PP traditionnel, cet utilisateur est appelé le *Driver* et son partenaire est appelé le *Navigator*. Le rôle du *Driver* est de travailler activement sur la tâche de programmation en rédigeant le code, tandis que le *Navigator* observe et assiste le *Driver* en suggérant des améliorations et en identifiant les parties du code problématiques.

Les exigences relatives aux outils de DPP ont été étudiées dans des travaux antérieurs. Winkler et al. [30] ont développé un processus d'évaluation systématique pour les outils de DPP et ont listé différentes propriétés souhaitées pour

ces outils. Ils ont notamment identifié plusieurs exigences comme étant hautement prioritaires. La *conscience de l'espace de travail* fait référence à la capacité des utilisateurs à comprendre qui participe à la session de DPP courante, à comprendre le rôle de chacun, et à être conscient des différentes interactions au sein de l'espace partagé. Le *contrôle d'accès* concerne les restrictions imposées sur les modifications du code source. Seules les modifications apportées par le *Driver* devraient être autorisées. Le *changement de rôle* fournit aux utilisateurs un mécanisme leur permettant d'intervenir leurs rôles au cours d'une session de DPP. Dans un contexte d'enseignement, ce mécanisme peut être adapté aux besoins de l'activité pédagogique en limitant ou contrôlant son utilisation par les étudiants. Les *outils de communication* offrent des canaux de communication tels que des canaux vocaux ou des chats textuels entre les participants d'une session de DPP. Enfin, les *fonctionnalités gestuelles* aident les utilisateurs à visualiser les actions des autres sur l'espace de travail partagé. Par exemple, la fonction de partage du curseur permet aux autres participants de voir le curseur d'un utilisateur. En plus de ces caractéristiques, Schummer et al. [25] ont identifié une autre exigence notable. Le *mode spectateur* permet aux utilisateurs de voir et de suivre les interactions d'une session collaborative en cours, sans influencer les participants de cette session. Cette fonction est très importante dans un contexte d'apprentissage à distance, par exemple pour permettre à un enseignant de suivre les actions de ses étudiants.

## 2.2 Exigences pour un EIAH de programmation dédié à des programmeurs novices

Pour l'apprentissage de la programmation, il est courant de ne pas utiliser les mêmes logiciels de programmation que dans le monde professionnel, mais plutôt des environnements d'apprentissage adaptés aux besoins des apprenants. Notre analyse effectuée sur des recherches antérieures visant à soutenir les programmeurs novices a permis d'identifier un certain nombre d'exigences. Les outils pédagogiques fournissent généralement aux programmeurs novices des interfaces *simplifiées* sous la forme d'éditeurs de code simples [3,17],

afin que les apprenants puissent se concentrer sur les concepts informatiques [11] sans s'efforcer à prendre en main des interfaces graphiques complexes. Les EIAHs mettent à disposition divers mécanismes pour gérer les *activités pédagogiques* pour lesquelles ils sont conçus, et pour affecter les étudiants à ces activités [12]. En ce qui concerne la programmation, les logiciels courants proposent généralement un support de débogage très riche, mais ils exigent beaucoup d'efforts pour des débutants. La recherche a donc largement étudié la manière d'améliorer la performance des programmeurs novices en *test et débogage* [4], ainsi que dans la compréhension des messages d'erreur du compilateur [6].

Une autre caractéristique importante des outils pédagogiques informatiques est la capacité à faire du *tutorat*, qu'il soit assuré par un humain ou par une intelligence artificielle [2,14].

Enfin, de plus en plus d'EIAH s'appuient sur la *collecte de données* pour améliorer l'aide apportée aux apprenants. Ces systèmes collectent des données

résultant des interactions des utilisateurs avec le système afin de mieux comprendre leurs comportements [18,26], prédire leurs performances [15], ou leur fournir les ressources pédagogiques dont ils ont besoin [10].

### 2.3 Systèmes existants pour le distributed pair programming

La plupart des environnements de développement intégrés (IDE) supportent le DPP nativement, ou par l'utilisation d'extensions telles que Live Share pour Visual Studio et Visual Studio Code, Code with me pour IntelliJ, ou Code Together pour Eclipse. Ces outils offrent des fonctionnalités très complètes en matière de conscience de l'espace de travail, de fonctionnalités gestuelles et d'outils de communication. Parfois, ils mettent également en place le contrôle d'accès. Par exemple, Live Share intègre un mode lecture seule qui empêche certains utilisateurs de modifier le code source. Cependant, aucune de ces extensions n'impose une séparation stricte des rôles entre le *Driver* et le *Navigator*, comme c'est le cas en pair programming traditionnel. En outre, les IDE sont souvent considérés comme inadaptés aux programmeurs débutants en raison de leur complexité [13]. Une étude de Rigby et al. [20] tend à montrer que l'utilisation d'un IDE crée plus de frustration et moins de satisfaction chez les programmeurs novices comparativement à un outil conçu spécifiquement pour eux.

Xpairtise [25] et XecliP [28] sont deux plugins Eclipse conçus par des chercheurs. Ils sont tous deux très complets en ce qui concerne les exigences du DPP identifiées dans la Section 2.1, mais ils manquent de fonctionnalités pédagogiques et de tutorat car ils n'ont pas été spécialement conçus pour l'enseignement.

SCEPPSys [27] étend XecliP pour supporter l'utilisation du DPP en classe. Il est donc beaucoup plus complet en termes de fonctionnalités éducatives. En particulier, une interface d'administration permet aux enseignants de gérer les activités pédagogiques et la création des groupes. SCEPPSys recueille également un large éventail de données pour supporter l'évaluation des étudiants, comme le temps passé dans les rôles de *Driver* et de *Navigator*, la quantité de code produite ou l'utilisation d'outils de communication. Cependant, la collecte de données concernant l'exécution du code et les tests effectués par les étudiants semble limitée. De plus, comme Xpairtise et XecliP, cet outil est basé sur l'IDE Eclipse, ce qui n'est pas idéal pour les programmeurs novices comme expliqué précédemment.

Le Tableau 1 synthétise l'évaluation des outils mentionnés ci-dessus par rapport aux exigences identifiées dans les sections 2.1 et 2.2. Il met en évidence les principales lacunes existantes qui ont motivé la conception de notre application.

## 3 Notre application pour le distributed pair Programming

Cette section présente l'architecture de notre outil d'apprentissage du Python en se concentrant sur les caractéristiques du DPP et sur les aspects pédagogiques.

**TABLE 1.** Évaluation des outils existants sur les exigences en matière d'éducation et de DPP

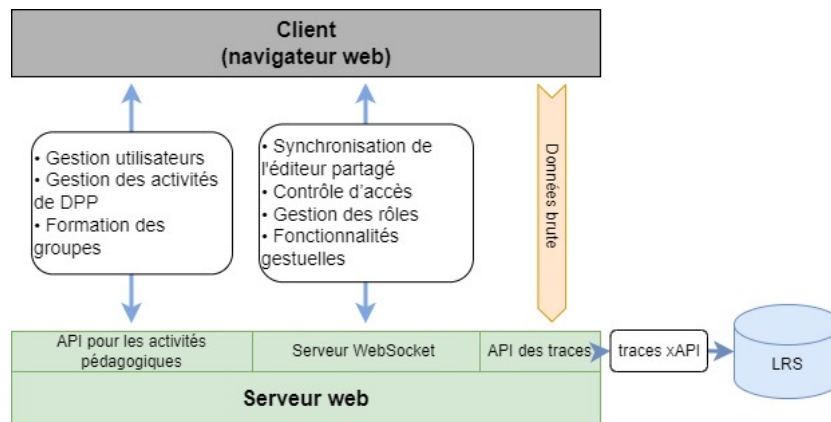
	Simplicité	Activités pédagogiques	Test et débogage	Tutorat	Collecte de données	Conscience de l'espace de travail	Contrôle d'accès	Changement de rôle	Outils de communication	Fonctionnalités gestuelles	Mode spectateur
Live Share (Visual studio and VSCode)	+	-	++	-	-	++	+	-	++	++	++
Code With Me(IntelliJ)	-	-	++	-	-	++	-	-	++	++	++
Code Together (Eclipse)	-	-	++	-	-	++	-	-	++	++	?
Xpairtise	-	-	++	+	?	++	++	++	+	++	++
XecliP	-	-	++	-	?	++	++	++	+	++	-
SCEPPSys	-	++	++	++	+	++	++	++	+	++	++

Légende : exigence entièrement prise en charge (++), partiellement prise en charge (+), non prise en charge (-), inconnue (?)

### 3.1 Aperçu général et technologies utilisées

L'application que nous avons développée est une application web, en vue d'encourager son utilisation par les enseignants et les étudiants à partir d'un simple navigateur web. La recherche de la **simplicité** est un facteur important ayant guidé la conception de notre application.

L'architecture client-serveur de notre système et les communications entre ses différents composants sont illustrées par la Figure 1. Le backend comprend plusieurs API NodeJS principalement dédiées aux fonctions éducatives, ainsi qu'un serveur WebSocket basé sur ShareDB. Ce dernier fournit une communication duplex entre le front et le backend pour mettre en oeuvre la synchronisation de l'éditeur partagé entre les utilisateurs, ainsi que de nombreuses **fonctionnalités gestuelles**. Le backend intègre également un *Learning Record Store* (LRS) conforme au standard xAPI pour stocker les traces d'interactions. Ce LRS a été développé avec NodeJS et utilise MongoDB pour la base de données. Des travaux sont en cours pour utiliser le système TraxLRS qui est une solution plus aboutie. Il est également important de noter que l'interprétation du code Python est faite directement dans le navigateur web par l'utilisation de Pyodide, ce qui facilite le passage à l'échelle de l'application en minimisant la charge du serveur.



**FIGURE 1.** Architecture globale de notre outil dédié au distributed pair programming pour l'apprentissage de Python.

### 3.2 Fonctionnalités de distributed pair programming

L'interface de l'application est illustrée par la figure 2 et comprend 5 composants. L'éditeur partagé (composant 1 de la figure 2) est synchronisé entre tous les membres d'une même session de DPP pour permettre à chacun de voir le même code source. Il implémente deux **fonctionnalités gestuelles** importantes : le curseur partagé et la sélection partagée, en utilisant des couleurs pour identifier les différents utilisateurs, et met en oeuvre le **contrôle d'accès**, puisque seul l'utilisateur ayant le rôle de *Driver* peut éditer le code source. Le composant 3 expose le problème de programmation à résoudre. Le composant 4 se concentre sur l'exigence de la **conscience de l'espace de travail** en montrant les utilisateurs actuellement connectés à la session de DPP, et en les associant à une couleur, utilisée par d'autres éléments de l'interface. Il montre l'utilisateur dans le rôle de *Driver* par le biais d'une icône spécifique. L'enseignant a accès à toutes les sessions de DPP de ses étudiants et apparaîtra dans le composant 4 s'il en rejoint une. Cela permet d'assurer l'intégration du **mode spectateur** et contribue à supporter le **tutorat** dans la mesure où cela aide l'enseignant à être conscient des difficultés rencontrées par les étudiants. Le composant 5 est consacré à l'exigence du **changement de rôle**, et permet aux utilisateurs dans le rôle de *Navigator* d'effectuer des requêtes pour changer de rôle. Le *Driver* actuel reçoit alors une notification de la requête et a la possibilité de l'accepter ou de la rejeter. En fonction de la réponse, le changement de rôle a lieu ou non. Enfin, l'interface utilisateur met en oeuvre l'exigence de **test et débogage** par le biais de la console Python du composant 2. Chaque utilisateur dispose de sa propre console, qui n'est pas partagée avec les autres participants de la session de DPP. Cette console permet aux participants d'une session d'exécuter l'ensemble du code Python présent dans l'éditeur partagé et d'exécuter n'importe quelle commande Python pour tester le code.

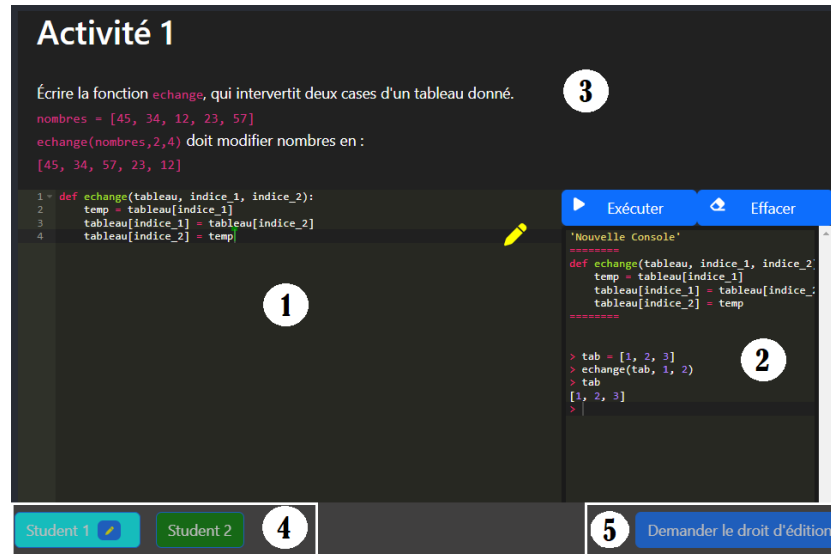


FIGURE 2. L'interface utilisateur de l'application dédiée aux exigences du DPP.

### 3.3 Fonctionnalités pédagogiques

Les fonctionnalités pédagogiques sont principalement mises en oeuvre par le backend de l'application par le biais de deux API.

La première API est dédiée à la gestion des **activités pédagogiques**. Elle permet la création, la modification et le déploiement d'activités de DPP, qui consistent en (i) un problème à résoudre spécifié par l'enseignant, et (ii) une liste de paires d'étudiants. Cette API est accessible aux enseignants par le biais d'une interface web. Une fois l'activité créée, l'enseignant peut partager un lien d'accès à ses étudiants. Lorsqu'un étudiant utilise ce lien, le système démarre une session spécifique de l'activité pour lui et son binôme, et attribue le rôle de *Driver* au premier étudiant qui se connecte à l'activité.

L'autre API est responsable de la **collecte des données**. Elle recueille des données provenant du client et du serveur, les formate conformément à la norme xAPI et les stocke dans le LRS (voir Figure 1). Nous avons utilisé le format xAPI car il est largement adopté par la communauté de recherche en EIAH, et permet une analyse riche du comportement des étudiants. Nous collectons sept types de traces xAPI : connexion et déconnexion à une session, requêtes de changement de rôle et réponses, code écrit, exécutions du code (comprenant le résultat de l'exécution) et interactions avec la console Python. Toutes les traces comprennent un horodatage et un identifiant utilisateur. Par conséquent, à partir de ces données, il est possible de modéliser la séquence complète des interactions qui se sont produites au cours d'une session de DPP.



## 4 Expérimentation et évaluation

### 4.1 Configuration expérimentale

L'expérimentation a impliqué 82 étudiants de première année à l'université de la Réunion, dans le cadre d'un cours d'introduction à la programmation en Python.

Elle a été réalisée sur deux semaines, chacune comprenant une séance de travaux pratiques d'une heure. Au cours des deux séances, les étudiants ont été invités à résoudre des exercices sur le concept de liste bidimensionnelle ; les exercices étaient différents pour les deux séances. Notons que les étudiants n'ont pas été formés sur l'application avant l'expérimentation ; l'enseignant a simplement présenté brièvement l'application la première semaine. Les étudiants étant des novices en programmation, ils n'étaient pas familiers avec le PP au début de l'expérimentation.

Nous avons proposé deux conditions expérimentales pour évaluer notre application : la programmation en pair programming et la programmation en distributed pair programming. En PP, un seul ordinateur était connecté à l'application et les étudiants étaient assis l'un à côté de l'autre devant cet ordinateur. En revanche, en DPP, chaque étudiant d'une paire avait accès à un ordinateur et ils étaient tous deux connectés à la même session sur l'application. Ils étaient également assis l'un à côté de l'autre dans la salle de classe afin de pouvoir se parler. Cependant, ils ne pouvaient pas voir l'écran de leur partenaire, reproduisant ainsi une situation proche à celle que deux programmeurs peuvent expérimenter à distance en DPP, en utilisant un outil de communication. Dans les deux cas, les étudiants étaient libres de changer de rôle autant de fois qu'ils le souhaitaient pendant la séance. L'enseignant, en plus des interventions normales réalisées en séance de TP en regardant directement sur l'écran de ses étudiants, avait la possibilité d'utiliser le **mode spectateur** pour observer les sessions de travail.

Au cours de la première séance, 26 paires d'étudiants (groupe A) ont été placées en DPP et 15 paires d'étudiants (groupe B) en PP. Les paires d'étudiants ont été formées par ordre alphabétique au sein des groupes de travaux pratiques habituels pour ce cours. Comme nous voulions que chaque paire d'étudiants expérimente les deux conditions expérimentales, les étudiants du groupe A ont été assignés en PP pour la deuxième séance de l'expérimentation, tandis que les étudiants du groupe B ont été assignés en DPP. Les paires d'étudiants sont inchangées entre les deux séances.

### 4.2 Données collectées

L'utilisabilité perçue de l'application par les étudiants et la qualité de leurs productions ont été évaluées sur la base de données quantitatives.

Pour évaluer l'utilisabilité, nous avons demandé, de façon non obligatoire, aux étudiants de répondre au questionnaire System Usability Scale (SUS) [7] après chaque séance. Les étudiants ont ainsi donné leur perception de l'utilisabilité de l'application en PP et en DPP. Le SUS est reconnu comme un outil rapide et

fiable pour évaluer l'utilisabilité perçue d'un système [8]. En outre, des études antérieures ont montré qu'il s'agissait d'un instrument valide pour évaluer les applications web [29].

Afin de compléter les questions du questionnaire SUS, nous avons également distribué aux étudiants en DPP un questionnaire d'utilisabilité portant sur les principales caractéristiques de l'application. Il comprend quatre affirmations associées à une échelle de Likert en 5 points. Les quatre affirmations portaient respectivement sur la visualisation du curseur du partenaire (S1), la console Python (S2), la compréhension de la répartition des rôles (S3) et le mécanisme de changement de rôle (S4).

Pour évaluer le travail réalisé par les étudiants, nous avons collecté, pour chaque paire d'étudiants et pour chaque séance de l'expérimentation, la version finale de leur code source, c'est-à-dire le code source qu'ils ont laissé en se déconnectant de l'application. Ces productions ont ensuite été évaluées sur une échelle de 0 à 5 par un seul chercheur en informatique, afin d'éviter les disparités dans la notation des étudiants.

Enfin, nous avons distribué aux étudiants une question ouverte leur demandant quelles autres fonctionnalités ils aimeraient voir intégrées à l'application.

### 4.3 Analyse des données

La moyenne, l'écart-type et la médiane du score SUS ont été calculés pour chaque groupe d'étudiants, afin de comparer l'utilisabilité perçue entre les deux conditions expérimentales. En outre, pour déterminer s'il existe une différence significative entre les deux conditions, nous avons effectué un test T apparié pour chaque groupe. Nous avons également calculé la moyenne, l'écart-type et la médiane des réponses aux questions posées aux étudiants en DPP.

Pour évaluer l'impact du DPP sur la qualité des productions des apprenants comparativement au PP, nous avons comparé la moyenne, l'écart-type et la médiane des notes obtenues par les étudiants dans les deux conditions expérimentales. Cependant, contrairement à l'évaluation de l'utilisabilité, notre analyse a été effectuée indépendamment pour chaque séance de l'expérimentation, car les exercices étaient différents entre les deux séances. Nous avons également effectué un test T indépendant pour chaque séance, afin de déterminer s'il existait ou non une différence significative dans la qualité des productions des étudiants en fonction de la condition expérimentale.

## 5 Résultats et discussion

### 5.1 Utilisabilité perçue

Les résultats du questionnaire SUS sont basés uniquement sur les étudiants qui ont rempli le questionnaire pour chaque séance, c'est-à-dire ceux qui ont connu les deux conditions expérimentales. Nous avons recueilli 37 réponses dans le groupe A et 18 dans le groupe B.

Pour le groupe A, les scores moyens du SUS sont très similaires en PP ( $M = 69.19$ ,  $s.d = 16.33$ ,  $Mdn = 70$ ) et en DPP ( $M = 67.97$ ,  $s.d = 14.99$ ,  $Mdn = 67.5$ ). En ce qui concerne le groupe B, les scores en PP ( $M = 63.47$ ,  $s.d = 13.15$ ,  $Mdn = 67.5$ ) et en DPP ( $M = 65$ ,  $s.d = 10.96$ ,  $Mdn = 66.25$ ) sont plus faibles. Selon [5] et avec ces résultats, l'application peut être qualifiée par l'adjectif "OK".

Pour le questionnaire donné uniquement aux étudiants en DPP, nous avons obtenu 32 réponses du groupe A et 15 du groupe B. Les résultats donnés dans le tableau 2 sont cohérents avec ceux du questionnaire SUS. Ils indiquent que les étudiants ont tendance à être "d'accord" avec le fait que chaque fonctionnalité remplit son objectif.

Ces résultats sont encourageants pour plusieurs raisons. Tout d'abord, le score SUS moyen du groupe A, qui comprend plus d'étudiants que le groupe B, est de 68,58. Il est proche du seuil "Good" (71 selon [5]), et juste au-dessus du score SUS moyen pour les applications web (68,05), obtenu selon des recherches antérieures basées sur une évaluation empirique de 1180 résultats de questionnaires SUS existants [1]. Deuxièmement, les résultats montrent que pour les deux groupes, le score SUS moyen augmente entre la première et la deuxième séance.

Nous supposons que si les étudiants avaient été formés sur l'application avant l'expérimentation, les scores SUS moyens auraient atteint des valeurs plus élevées.

Troisièmement, les résultats des tests T appariés ne montrent pas de différence significative dans les scores SUS entre les deux conditions expérimentales pour le groupe A ( $t = 0.67$ ,  $p = 0.50$ ), et pour le groupe B ( $t = 0.46$ ,  $p = 0.64$ ). Ces résultats suggèrent qu'utiliser l'outil en DPP n'a pas d'impact négatif sur son utilisabilité perçue par rapport à une utilisation en PP.

**TABLE 2.** Moyenne, écart-type et médiane aux réponses du questionnaire donné uniquement aux étudiants en DPP.

	Groupe A (N = 32)			Groupe B (N = 15)			Groupe A et B		
	M	s.d	Mdn	M	s.d	Mdn	M	s.d	Mdn
S1	3.59	1.32	4.0	3.73	1.29	4.0	3.64	1.31	4.0
S2	3.78	1.11	4.0	3.47	1.15	4.0	3.64	1.13	4.0
S3	3.84	1.12	4.0	3.73	0.85	3.0	3.81	1.04	4.0
S4	3.63	1.36	4.0	4.07	0.85	4.0	3.77	1.24	4.0

## 5.2 Qualité des productions des apprenants

Les résultats des analyses des productions des étudiants montrent sur la séance 1 de meilleurs résultats pour les paires d'étudiants en DPP ( $M = 2.57$ ,  $s.d = 1.02$ ,  $Mdn = 2.25$ ) comparativement au PP ( $M = 2.43$ ,  $s.d = 0.89$ ,  $Mdn = 2.0$ ). Sur la séance 2, cette tendance entre le DPP ( $M = 2.23$ ,  $s.d = 1.70$ ,  $Mdn = 2.0$ ) et le PP ( $M = 1.29$ ,  $s.d = 1.48$ ,  $Mdn = 0.5$ ) est encore plus remarquable. Pour la séance 2, cette différence est attribuée à la difficulté de l'exercice de

programmation, qui a contraint 10 paires d'étudiants du groupe A à consacrer un temps excessif à la recherche d'une solution, sans succès.

Toutefois, ces résultats sont tempérés par les résultats des tests T indépendants, qui ne montrent pas de différence significative entre les deux conditions expérimentales, pour la séance 1 ( $t = 0,44$ ,  $p = 0,66$ ) et la séance 2 ( $t = 1,81$ ,  $p = 0,08$ ). Ils suggèrent une fois de plus que l'utilisation du DPP n'a pas eu d'impact négatif sur la qualité des productions des étudiants.

Parmi les 47 réponses à la question ouverte, 6 étudiants ont demandé l'intégration d'un chat textuel, ce qui souligne le besoin d'une communication textuelle même lorsque les étudiants peuvent communiquer à l'oral.

## 6 Limitations

Les principales limites du travail présenté sont relative à l'expérimentation. Pour la condition expérimentale en DPP, nous avons cherché à reproduire la situation réelle de deux apprenants engagés dans un apprentissage à distance. Toutefois, cette situation n'était pas véritablement "distribuée", car les étudiants étaient assis les uns à côté des autres. Il faudrait pour cela réaliser de véritables expérimentations à distance, en intégrant du DPP dans des MOOCs par exemple. Pour les résultats d'utilisabilité, nous avons recueilli un nombre limité de réponses pour les deux questionnaires, ce qui rend difficile la mise en évidence des caractéristiques de notre outil nécessitant des améliorations et/ou l'identification du besoin pour de nouvelles caractéristiques. Pour faciliter ces expérimentations à grande échelle et collecter une plus grande quantité de données, nous allons améliorer l'infrastructure de l'application et étendre son utilisation à d'autres institutions et lycées.

## 7 Conclusion et travaux futurs

Cet article propose une nouvelle application de distributed pair programming destinée aux programmeurs novices. Elle met en oeuvre toutes les fonctionnalités identifiées dans notre état de l'art, à l'exception d'un outil de communication. L'expérimentation que nous avons menée avec 82 participants en utilisant l'outil en condition de DPP n'a pas montré d'impact négatif sur l'utilisabilité perçue par les étudiants et la qualité de leurs productions, par rapport à son utilisation en PP traditionnel. Ces résultats sont encourageants en ce qui concerne l'utilisation du DPP dans l'éducation et pour nos futures utilisations de l'outil.

Dans nos travaux futurs, nous nous concentrerons sur l'implémentation d'outils de learning analytics se basant sur les données collectées par l'application, comme par exemple des indicateurs liés à l'engagement individuel et collectif des étudiants. Pour les étudiants, être conscient de leur taux de participation individuel au sein du binôme pourrait les aider à équilibrer leur participation globale [23], tandis que connaître le niveau d'engagement de chaque paire devrait aider les enseignants à intervenir en conséquence. Par la suite, nous prévoyons d'explorer la formation automatique de paires sur la base des données collectées

par notre application. En effet, des recherches suggèrent que la compatibilité et la formation des paires ne sont pas souvent analysées dans les études de PP [22], [21]. Enfin, nous prévoyons de publier l'application en open source afin de permettre aux enseignants de l'utiliser dans d'autres contextes, ainsi qu'aux chercheurs de contribuer à son amélioration.

## Références

1. Aaron Bangor, P.T.K., Miller, J.T. : An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction* **24**(6), 574–594 (2008)
2. Ahmed, U.Z., Srivastava, N., Sindhgatta, R., Karkare, A. : Characterizing the pedagogical benefits of adaptive feedback for compilation errors by novice programmers. In : *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering : Software Engineering Education and Training*. pp. 139–150 (2020)
3. Allen, E., Cartwright, R., Stoler, B. : Drjava : a lightweight pedagogic environment for java. *SIGCSE Bull.* **34**(1), 137–141 (feb 2002)
4. Ardimento, P., Bernardi, M.L., Cimitile, M., Ruvo, G.D. : Reusing bugged source code to support novice programmers in debugging tasks. *ACM Transactions on Computing Education (TOCE)* **20**(1), 1–24 (2019)
5. Bangor, A., Kortum, P., Miller, J. : Determining what individual sus scores mean : Adding an adjective rating scale. *Journal of usability studies* **4**(3), 114–123 (2009)
6. Becker, B.A. : An effective approach to enhancing compiler error messages. In : *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. p. 126–131. *SIGCSE '16* (2016)
7. Brooke, J. : Sus : a “quick and dirty” usability. *Usability evaluation in industry* **189**(3), 189–194 (1996)
8. Brooke, J. : Sus : a retrospective. *Journal of usability studies* **8**(2), 29–40 (2013)
9. Faja, S. : Pair programming as a team based learning activity : A review of research. *Issues in Information Systems* **XII**, 207–216 (01 2011)
10. Fu, X., Shimada, A., Ogata, H., Taniguchi, Y., Suehiro, D. : Real-time learning analytics for c programming language courses. In : *Proceedings of the Seventh International Learning Analytics & Knowledge Conference*. p. 280–288. *LAK '17* (2017)
11. Hempel, B., Lubin, J., Lu, G., Chugh, R. : Deuce : a lightweight user interface for structured editing. In : *Proceedings of the 40th International Conference on Software Engineering*. p. 654–664. *ICSE '18* (2018)
12. Karkare, A., Kar, P. : Prutor : an intelligent learning and management system for programming courses. *Communications of the ACM* **65**(11), 62–64 (2022)
13. Karvelas, I. : Investigating novice programmers' interaction with programming environments. In : *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*. p. 336–337. *ITiCSE '19* (2019)
14. Keuning, H., Heeren, B., Jeurig, J. : A tutoring system to learn code refactoring. In : *Proceedings of the 52nd ACM technical symposium on computer science education*. pp. 562–568. *Association for Computing Machinery, New York, NY, USA* (2021)

15. Leinonen, J., Castro, F.E.V., Hellas, A. : Time-on-task metrics for predicting performance. *ACM Inroads* **13**(2), 42–49 (2022)
16. Papadakis, S. : Is pair programming more effective than solo programming for secondary education novice programmers? a case study. *International Journal of Web-Based Learning and Teaching Technologies* **13** (01 2018)
17. Payne, J., Cavé, V., Raman, R., Ricken, M., Cartwright, R., Sarkar, V. : Drhj : a lightweight pedagogic ide for habanero java. In : *Proceedings of the 9th International Conference on Principles and Practice of Programming in Java*. p. 147–150. PPPJ '11, Association for Computing Machinery, New York, NY, USA (2011), <https://doi.org/10.1145/2093157.2093180>
18. Pereira, F.D., Oliveira, E.H., Oliveira, D.B., Cristea, A.I., Carvalho, L.S., Fonseca, S.C., Toda, A., Isotani, S. : Using learning analytics in the amazonas : understanding students' behaviour in introductory programming. *British journal of educational technology* **51**(4), 955–972 (2020)
19. Rasheed, R.A., Kamsin, A., Abdullah, N.A. : Challenges in the online component of blended learning : A systematic review. *Computers & Education* **144**, 103701 (2020)
20. Rigby, P.C., Thompson, S. : Study of novice programmers using eclipse and gild. In : *Proceedings of the 2005 OOPSLA Workshop on Eclipse Technology EXchange*. p. 105–109. eclipse '05 (2005)
21. Salleh, N., Mendes, E., Grundy, J. : Empirical studies of pair programming for cs/se teaching in higher education : A systematic literature review. *IEEE Transactions on Software Engineering* **37**(4), 509–525 (2011)
22. Satratzemi, M., Stelios, X., Tsompanoudi, D. : Distributed pair programming in higher education : A systematic literature review. *Journal of Educational Computing Research* **61**(3), 546–577 (2023)
23. Satratzemi, M., Xinogalos, S., Tsompanoudi, D., Karamitopoulos, L. : Examining Student Performance and Attitudes on Distributed Pair Programming. *Scientific Programming* **2018** (Oct 2018)
24. Schenk, J., Prechelt, L., Salinger, S. : Distributed-pair programming can work well and is not just distributed pair-programming. In : *Companion Proceedings of the 36th International Conference on Software Engineering*. p. 74–83. ICSE Companion 2014 (2014)
25. Schümmer, T., Lukosch, S.G. : Understanding tools and practices for distributed pair programming. *Journal of Universal Computer Science*, 15 (16), 2009 **15**(16), 3101–3125 (2009)
26. Song, D., Hong, H., Oh, E.Y. : Applying computational analysis of novice learners' computer programming patterns to reveal self-regulated learning, computational thinking, and learning performance. *Computers in Human Behavior* **120**, 106746 (2021)
27. Tsompanoudi, D., Satratzemi, M. : A web-based authoring tool for scripting distributed pair programming. In : *2014 IEEE 14th International Conference on Advanced Learning Technologies*. pp. 259–263. IEEE, Athens, Greece (2014)
28. Tsompanoudi, D., Satratzemi, M., Xinogalos, S. : Exploring the effects of collaboration scripts embedded in a distributed pair programming system. In : *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*. pp. 225–230. ITiCSE '13 (2013)

29. Tullis, T., Stetson, J. : A comparison of questionnaires for assessing website usability. UPA Presentation p. 12 p. (2004)
30. Winkler, D., Biff, S., Kaltenbach, A. : Evaluating tools that support pair programming in a distributed engineering environment. In : Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering. p. 54–63. EASE'10, BCS Learning & Development Ltd., Swindon, GBR (2010)
31. Wood, K., Parsons, D., Gasson, J., Haden, P. : It's never too early : pair programming in cs1. In : Proceedings of the Fifteenth Australasian Computing Education Conference - Volume 136. p. 13–21. ACE '13 (2013)