

# Chapitre 1 : Présentation générale de la programmation par blocs

## 1 Introduction

Dans ce chapitre, nous aborderons la programmation par blocs, examinant ses fondements, ses objectifs, sa comparaison avec la programmation classique, ainsi que les plateformes courantes utilisées dans ce domaine. De plus, nous identifierons les défis et proposerons des solutions pour promouvoir une utilisation efficace de la programmation par blocs dans l'éducation et le développement des compétences en programmation.

Voici le sommaire du chapitre :

- Introduction
- Définition & Objectifs
- Comparaison avec la Programmation Classique
- Avantages et Inconvénients de la Programmation par Blocs
- Évolution Historique et Plateformes Courantes
- Tableaux de Bord Collaboratifs
- Problématique et Solution Proposée
- Conclusion

## 2 Définition & Objectifs

### 2.1 Définition

La programmation par blocs, également connue sous le nom de programmation visuelle, est une approche de développement logiciel qui repose sur l'assemblage de blocs graphiques pour créer des applications. Chaque bloc représente une fonction ou une opération spécifique, et les blocs sont interconnectés pour former un programme complet. Cette méthode offre une interface visuelle facile à utiliser, permettant aux développeurs de concevoir des logiciels sans écrire de code traditionnel. Cette approche est particulièrement bénéfique pour les débutants car elle simplifie le processus d'apprentissage en masquant la complexité de la syntaxe et de la structure du code federici2011,federici2015.

### 2.2 Objectifs

La programmation par blocs vise à simplifier le processus de développement en fournissant une alternative visuelle à la programmation textuelle. Les principaux objectifs de cette approche incluent :

#### 2.2.1 Faciliter l'apprentissage

En éliminant la nécessité de maîtriser une syntaxe complexe, la programmation par blocs rend l'apprentissage de la programmation plus accessible, surtout pour les débutants. Les environnements de programmation visuelle comme Scratch et Blockly sont largement utilisés dans les milieux éducatifs pour introduire les concepts de programmation aux jeunes apprenants costa2016.

#### 2.2.2 Accélérer le développement

La visualisation des blocs permet de comprendre rapidement la logique du programme, réduisant ainsi le temps nécessaire au développement des applications. Cette approche favorise le prototypage rapide et le développement itératif, essentiels dans les pratiques modernes d'ingénierie logicielle federici2018.

#### 2.2.3 Encourager la collaboration

La nature visuelle de la programmation par blocs facilite la collaboration entre les membres de l'équipe, même s'ils n'ont pas les mêmes compétences en programmation. Des outils comme Thunkable et MIT App Inventor offrent des fonctionnalités collaboratives permettant à plusieurs utilisateurs de travailler simultanément sur le même projet costa2016.

### 2.2.4 Favoriser la réutilisation du code

Les blocs peuvent être réutilisés dans différents projets, favorisant ainsi la création d'une bibliothèque de composants prêts à l'emploi et accélérant le développement futur. Cette modularité est un avantage clé pour la construction de systèmes complexes à partir de parties plus simples et réutilisables federici2014.

## 3 Comparaison avec la Programmation Classique

La programmation par blocs diffère de la programmation classique, qui repose souvent sur des langages textuels tels que C++, Java ou Python. Voici une comparaison entre ces deux approches :

### 3.1 Syntaxe et Apprentissage

La programmation classique nécessite la maîtrise d'une syntaxe spécifique, ce qui peut être difficile pour les débutants. En revanche, la programmation par blocs offre une interface visuelle intuitive, réduisant la courbe d'apprentissage et facilitant l'adoption par les novices costa2016.

### 3.2 Rapidité de Développement

La programmation par blocs permet une conception rapide des applications grâce à son approche visuelle. En comparaison, la programmation classique peut être plus chronophage en raison de la saisie manuelle du code federici2018.

### 3.3 Flexibilité et Contrôle

La programmation classique offre souvent une plus grande flexibilité et un contrôle précis sur le code, ce qui peut être essentiel pour des projets complexes nécessitant des optimisations spécifiques. Bien que plus conviviale, la programmation par blocs peut avoir des limitations en termes de personnalisation des détails federici2011.

### 3.4 Collaboration et Réutilisation

La programmation par blocs se prête bien à la collaboration grâce à son interface visuelle. Cependant, la programmation classique peut être plus propice à la création de solutions spécifiques et à la gestion des détails fins costa2016.

En résumé, la programmation par blocs présente une méthode différente de la programmation classique, avec des avantages particulièrement remarquables en termes d'apprentissage rapide et de développement visuel. Cependant, la programmation classique conserve son importance dans des contextes nécessitant un contrôle détaillé et une personnalisation avancée.

## 4 Avantages et Inconvénients de la Programmation par Blocs

La programmation par blocs présente des avantages et des inconvénients distincts, chacun impactant son utilisation dans le développement logiciel. Cette section explore ces aspects pour fournir une vue d'ensemble complète de cette approche.

### 4.1 Avantages

#### 4.1.1 Accessibilité

La programmation par blocs offre une interface visuelle conviviale, abaissant les barrières pour les débutants. Cela permet à un plus grand nombre de personnes d'explorer et de participer au développement logiciel sans avoir besoin de connaissances approfondies en langages de programmation traditionnels grover2013,mohamad2011a.

#### 4.1.2 Rapidité de Développement

L'assemblage visuel des blocs simplifie la création d'applications, accélérant ainsi le développement. Cette approche favorise l'itération rapide, ce qui peut être particulièrement bénéfique dans les projets nécessitant des cycles de développement courts mohamad2011a.

#### 4.1.3 Compréhension Visuelle

La représentation graphique des blocs permet une compréhension visuelle immédiate de la logique du programme. Cela facilite la collaboration en équipe en rendant le code plus accessible et en encourageant une communication claire weintrop2015a.

#### 4.1.4 Réutilisation du Code

La modularité inhérente à la programmation par blocs facilite la réutilisation des composants. Les blocs peuvent être enregistrés sous forme de modules réutilisables, accélérant le développement futur et favorisant la création d'une bibliothèque de composants prêts à l'emploi mohamad2011b.

### 4.2 Inconvénients

#### 4.2.1 Limitations de Personnalisation

Bien que la programmation par blocs offre une approche conviviale, elle peut présenter des limitations en termes de personnalisation. Les projets complexes nécessitant un contrôle précis du code peuvent trouver cette approche moins adaptée weintrop2015b.

#### 4.2.2 Complexité des Projets Importants

Dans les projets logiciels de grande envergure, la programmation par blocs peut devenir complexe et moins efficace. La gestion des interactions entre de nombreux blocs peut devenir laborieuse, entraînant des difficultés de maintenance moors2018,matsuzawa2016.

## 5 Évolution Historique et Plateformes Courantes

### 5.1 Évolution Historique

La programmation par blocs a commencé il y a longtemps et s'est améliorée au fil des ans :

- Années 1960-1970 : Des gens ont eu des idées pour rendre la programmation plus facile en utilisant des images, mais ça n'a pas très bien fonctionné à cause de la technologie de l'époque papert1980.
- Années 1980-1990 : Les choses ont commencé à s'améliorer avec des ordinateurs qui pouvaient mieux gérer les images. Les gens ont commencé à utiliser des programmes qui rendaient la programmation plus simple et ces programmes étaient très populaires dans les écoles et les entreprises harvey1986.
- Années 2000-2010 : Internet est devenu très important et les gens ont commencé à créer des programmes de programmation par blocs que l'on pouvait utiliser en ligne. Des plateformes comme Scratch et Blockly sont devenues très populaires pour apprendre à programmer resnick2009.

### 5.2 Plateformes Courantes

Plusieurs plateformes populaires utilisent la programmation par blocs, notamment :

### 5.2.1 Scratch

Développée par le MIT, Scratch est une plateforme de programmation visuelle destinée aux jeunes. Elle permet aux utilisateurs de créer des histoires interactives, des jeux et des animations en utilisant des blocs de code empilables resnick2009.

### 5.2.2 Blockly

Créé par Google, Blockly est une bibliothèque JavaScript qui permet de créer des environnements de programmation par blocs. Il est souvent utilisé dans des projets éducatifs et des applications Web pour enseigner les concepts de programmation fraser2013.

### 5.2.3 MIT App Inventor

MIT App Inventor est une plateforme en ligne qui permet aux utilisateurs de créer des applications mobiles Android à l'aide d'une interface de programmation par blocs. Elle est particulièrement populaire dans le domaine de l'éducation pour enseigner le développement d'applications wolber2014.

### 5.2.4 Thunkable

Thunkable est une plateforme similaire à MIT App Inventor, permettant la création d'applications mobiles pour Android et iOS. Elle propose également des fonctionnalités avancées pour les développeurs expérimentés becker2017.

## 5.3 Comparaison des Plateformes de Programmation par Blocs

	Microsoft Make-Code	mBlock	Scratch	Code.org
Développeur	Microsoft	Makeblock	MIT Media Lab	Code.org
Type de Programmation	Programmation matérielle et visuelle	Programmation matérielle et visuelle	Programmation visuelle	Programmation visuelle
Langage de Programmation	JavaScript, Python, Make-Code Blocs	mBlock Blocks, Python	Scratch	Code.org Blocks, JavaScript
Accessibilité	Gratuit, web et application de bureau	Gratuit, web et application de bureau	Gratuit, web, application de bureau	Gratuit, web
Cible d'âge	10 ans et plus	8-18 ans	8-16 ans	5-18 ans
Utilisation de Blockly	Oui	Non	Non	Oui
Support Matériel	micro:bit, Circuit Playground, LEGO Mindstorms	mBot, Codey Rocky, divers dispositifs matériels	Aucun (peut être étendu avec extensions tierces)	Aucun spécifique (peut être étendu)
Exemple d'utilisation avec robots	Utilisé pour programmer le micro:bit et les robots LEGO Mindstorms	Utilisé pour programmer des robots Makeblock comme mBot et Codey Rocky	Peut être utilisé pour programmer le robot LEGO Mindstorms avec extensions	Utilisé dans des cours pour contrôler des robots simples comme les BeeBots
Communauté	Grande, support officiel et forums	Grande, support officiel et forums	Très large, support officiel et forums	Très large, utilisé dans de nombreuses écoles

	<b>Pocket Code</b>	<b>Thunkable</b>	<b>Catrobat</b>	<b>App Inventor</b>
<b>Développeur</b>	Catrobat	Thunkable Inc.	Catrobat	MIT
<b>Type de Programmation</b>	Programmation visuelle	Développement d'applications mobiles	Programmation visuelle	Développement d'applications mobiles
<b>Langage de Programmation</b>	Catrobat Language	Blocs Thunkable	Catrobat Language	Blocs App Inventor
<b>Accessibilité</b>	Gratuit, application mobile	Gratuit avec options payantes, web	Gratuit, application mobile	Gratuit, web
<b>Cible d'âge</b>	10 ans et plus	13 ans et plus	10 ans et plus	13 ans et plus
<b>Utilisation de Blockly</b>	Oui	Non	Non	Oui
<b>Support Matériel</b>	Smartphones Android	Smartphones Android et iOS	Aucun spécifique (peut être étendu)	Smartphones Android
<b>Exemple d'utilisation avec robots</b>	Non spécifiquement conçu pour les robots mais peut contrôler via Bluetooth	Non spécifiquement conçu pour les robots mais peut contrôler via Bluetooth	Non spécifiquement conçu pour les robots mais peut contrôler via Bluetooth	Non spécifiquement conçu pour les robots mais peut contrôler via Bluetooth
<b>Communauté</b>	Modérée, communauté en ligne active	Modérée, support officiel et forums	Modérée, communauté en ligne active	Large, support officiel et forums

## 6 Tableaux de Bord Collaboratifs

Les tableaux de bord collaboratifs sont des outils essentiels dans la programmation par blocs, facilitant la gestion et la collaboration sur les projets. Ils offrent une interface visuelle pour suivre l'avancement des tâches, partager des ressources et coordonner les efforts des équipes.

### 6.1 Fonctionnalités Clés

- Suivi de Projet : Les tableaux de bord permettent de suivre l'état d'avancement des projets en temps réel, offrant une vue d'ensemble claire des tâches accomplies et des étapes restantes agilemanifesto2001.
- Collaboration en Temps Réel : Les membres de l'équipe peuvent collaborer simultanément sur le même projet, partageant des idées et des ressources via une interface commune scrumguide2020.
- Gestion des Ressources : Les tableaux de bord offrent des outils pour gérer les ressources, y compris le code, les documents et les médias, facilitant ainsi l'organisation et l'accès aux informations nécessaires kanban2007.

### 6.2 Exemples de Tableaux de Bord

- Trello : Utilisé pour la gestion de projet, Trello propose une interface visuelle basée sur des cartes et des listes, idéale pour suivre les tâches et les progrès.
- Jira : Une plateforme de gestion de projet plus avancée, Jira offre des fonctionnalités de suivi des bogues, de gestion des tâches et de collaboration en équipe.
- Asana : Asana est une autre plateforme populaire pour la gestion de projet, offrant des outils pour suivre les tâches, gérer les échéances et collaborer efficacement.

## 7 Problématique et Solution Proposée

La programmation par blocs présente des avantages significatifs pour l'apprentissage et le développement rapide d'applications. Cependant, elle peut également rencontrer des défis, notamment en termes de personnalisation et de gestion de projets complexes. Pour adresser ces défis, il est crucial de proposer des solutions adaptées.

### 7.1 Problématique

Les principaux défis de la programmation par blocs incluent :

### **7.1.1 Limitations en Personnalisation**

La flexibilité réduite pour les projets complexes nécessitant des personnalisations spécifiques peut constituer une barrière importante grover2013.

### **7.1.2 Complexité dans les Grands Projets**

La gestion des interactions entre de nombreux blocs peut devenir laborieuse et difficile à maintenir moors2018.

## **7.2 Solution Proposée**

Pour surmonter ces défis, nous proposons les solutions suivantes :

### **7.2.1 Intégration de Langages Textuels**

Combiner la programmation par blocs avec des langages textuels pour offrir une plus grande flexibilité et permettre des personnalisations plus fines. Des plateformes comme Snap! et MakeCode offrent déjà cette intégration, permettant aux utilisateurs d'écrire du code textuel pour des parties spécifiques de leur programme snap2011.

### **7.2.2 Outils de Gestion de Projets Améliorés**

Développer des outils de gestion de projets spécialement conçus pour la programmation par blocs, offrant des fonctionnalités avancées pour la gestion des interactions et la maintenance des blocs. Ces outils pourraient inclure des fonctionnalités de documentation automatique, de suivi des dépendances et de vérification de l'intégrité des blocs kanban2007.

## **8 Conclusion**

La programmation par blocs représente une avancée significative dans le domaine du développement logiciel, offrant une approche accessible et intuitive pour les débutants. Bien qu'elle présente certains défis, des solutions innovantes peuvent permettre de surmonter ces obstacles et d'exploiter pleinement le potentiel de cette méthode.