

**UN JEU DE RÔLES
POUR L'ENSEIGNEMENT DE LA PROGRAMMATION**

**Viviane Guéraud
Jean-Pierre Peyrin**

**Laboratoire de Génie Informatique
IMAG-Campus
BP 53 X
38041 GRENOBLE CEDEX**

RÉSUMÉ

Nous faisons d'abord un constat : l'enseignement de la programmation a considérablement évolué en quelques années ; il s'intéresse maintenant plus aux formes de raisonnement et aux méthodes de travail qu'à la codification des programmes. Mais il doit prendre en compte un paradoxe. D'une part, chez l'étudiant, le besoin de méthodes de programmation n'apparaît concrètement que lors des premiers projets consistants. D'autre part, l'enseignant présente ces méthodes sur des exemples trop simples pour justifier pleinement leur emploi, car il doit présenter en même temps que les méthodes les techniques de base de la programmation.

Nous proposons alors, pour résoudre ce problème, une activité nouvelle à insérer dans le cursus informatique ; il s'agit d'un jeu de rôle pratiqué par trois joueurs sur un réseau de micro-ordinateurs. Ce jeu met en évidence, par la pratique, les comportements nécessaires au programmeur dans le travail qui le mène de l'énoncé du problème à l'une de ses solutions.

UN JEU DE RÔLES POUR L'ENSEIGNEMENT DE LA PROGRAMMATION

**Viviane Guéraud
Jean-Pierre Peyrin**

**Laboratoire de Génie Informatique
IMAG-Campus
BP 53 X
38041 GRENOBLE CEDEX**

Grenoble 1987

Pendant très longtemps, l'enseignement de la programmation a été réduit à l'enseignement d'un langage. Les élèves apprenaient « comment dire », à eux d'imaginer « quoi dire ». Ce stade est maintenant dépassé et nous pouvons constater un consensus des enseignants ([Meyer 78], [Arsac 80], [Lucas 83], [Ducrin 84], [Laurent 85], ...) qui s'intéressent dorénavant plutôt aux formes de raisonnement et aux méthodes de travail en vue d'une bonne conception des programmes. Là est bien le coeur de la programmation, comme en témoigne Anna Gram dont le souci est de rendre la démarche informatique systématique par l'étude des principaux raisonnements qui sont à la base de la construction de logiciels [Gram 86].

Nous nous situons ici dans le contexte de la programmation impérative car c'est l'usage dans la pratique de l'enseignement de premier niveau. Ce choix est réaliste puisqu'il conduit facilement aux langages les plus répandus, donc à l'attente des étudiants ainsi que des collègues d'autres disciplines. Mais il est peut-être contestable : en effet les notations actionnelles interdisent moins une construction empirique des programmes (puisque l'on peut "raisonner" en termes de l'exécution) que les notations fonctionnelles ou logiques.

LA COMPLEXITÉ EN TOUTE SIMPLICITÉ

La forme de l'enseignement ne peut pas être essentiellement un exposé théorique, mais doit être une pratique sur des exemples significatifs, en mettant en évidence la démarche utilisée pour parvenir à une solution. Le premier souci n'est pas l'efficacité de l'algorithme obtenu mais la clarté de son analyse ainsi que sa correction et sa lisibilité. Aussi, on note une même façon de procéder pour la plupart des "écoles" : les notions et

techniques enseignées sont abordées par des petits problèmes exemplaires bien choisis. Chacun est consacré à un petit nombre de sujets, de manière à sérier les difficultés rencontrées. Ceci a une incidence sur le choix de l'énoncé, qui doit conserver une certaine simplicité éventuellement au détriment de son réalisme [Scholl 87].

Cette idée convient tout à fait à la présentation des techniques puisque chacune d'elles est présentée dans un contexte d'application. Elle est aussi utilisée pour la présentation des méthodes. Par exemple, l'analyse descendante est introduite dans [Arsac 80] par l'exercice "Quel jour de la semaine tombe la fête de Noël ?", dans [Ducrin 84] par l'exercice "Calcul de la paie d'un ouvrier", dans [Lucas 83] par l'exercice "Nombre d'occurrences du couple 'LE' dans un texte". Mais la difficulté est de faire pratiquer ces démarches systématiques, de manière satisfaisante, sur des exercices aussi simples. La spécification et l'analyse semblent artificielles lorsqu'un énoncé est trop simple.

Le premier vrai problème en programmation, c'est la maîtrise de la complexité : il faut faire face à trop de détails à la fois [Arsac 80]. Or l'enseignant doit convaincre de cette complexité pour que le débutant puisse et veuille pratiquer dès le départ une conduite rigoureuse.

Cette difficulté est résolue si l'on s'adresse à des programmeurs professionnels au cours d'un stage de formation continue. On peut en fait, avant de faire travailler un tel public sur les exemples scolaires, lui présenter un "gros" problème. Les stagiaires sont ainsi mis devant leurs limites, leurs incertitudes et leurs difficultés. Et lorsque l'on revient à des "petits" problèmes, le discours du formateur est écouté et bénéfique. Les méthodes prennent tout leur poids. La difficulté n'étant pas alors de faire accepter l'intérêt de ces méthodes et de les faire pratiquer pendant le stage, mais bien de parvenir à ce que le programmeur, une fois remis dans son contexte habituel, ne reprenne pas ses vieilles habitudes.

Par contre, une telle pratique est difficilement réalisable au niveau d'un public débutant. Les exemples et exercices proposés ne parviennent pas à le convaincre, puisqu'il peut parvenir à un résultat correct (qu'il estime correct !) sans avoir utilisé les méthodes enseignées. Elles lui apparaissent alors comme une rigidité de théoricien, comme des "marottes" d'enseignant. La tentation est trop grande de vite griffonner quelques instructions, de les rentrer en machine, et de voir la réponse de l'ordinateur. La référence pour le débutant, ce n'est pas ce que pense l'enseignant de son travail, c'est l'ordinateur : "ça tourne!". Son but est d'aligner très vite, ligne après ligne, les instructions qui composeront son programme, en commençant naturellement par écrire le "début" ou "begin" fondamental ! Il n'a pas envie de prendre conscience de la démarche qu'il a utilisée, ni même de la provenance de ses erreurs. L'enseignant fait remarquer une erreur ? "Ah oui, mais si je mets «+1» ici, ça marche" !

Lorsque les projets consistants arrivent plus tard dans le cursus, le débutant qui n'a pas attaché assez d'importance au discours de l'enseignant, et qui a malgré cela programmé les premiers travaux pratiques, a perdu plusieurs mois et ne parvient souvent pas à combler son retard.

L'enseignant a donc deux préoccupations contradictoires lorsqu'il commence un cours de programmation : présenter les concepts de base à l'aide d'exercices presque simplistes mais accessibles, et concrétiser ces concepts à l'aide d'un problème réel mais quasiment infaisable par les débutants.

Nous voulons imaginer une activité permettant au débutant de comprendre la nécessité profonde d'une attitude méthodique et de transformer son comportement naturellement empirique en expérimentant les méthodes proposées sur des problèmes significatifs.

UN CHANGEMENT DE DOMAINE

Il s'agit de proposer au débutant de mener à bien une sorte de projet, mais en omettant tous les détails de programmation. Nous souhaitons isoler des comportements pour les mettre en valeur . En fait nous voulons modéliser les différentes attitudes d'un programmeur lors de la résolution d'un problème et en proposer une expérimentation.

La façon la plus évidente d'offrir un travail sur la complexité d'une analyse sans aborder les techniques de la programmation est de remplacer ces techniques non encore enseignées par d'autres techniques connues, donc de changer de domaine.

Le débutant mettra en pratique, dans ce nouveau domaine, des attitudes similaires à celles que l'on demande à un programmeur. La réflexion sur ces comportements se fera au cours de cette pratique, et l'enseignant établira l'analogie avec la programmation au cours de son enseignement.

L'idée d'un changement de domaine n'est pas nouvelle : E. Dijkstra expliquait jadis les notions fondamentales de l'algorithmique à travers l'épluchage des pommes de terre [Dijkstra 71] ; le premier proverbe de H. Ledgard est illustré par la cuisson d'un poulet [Ledgard 75] ; J. Biondi et G. Clavel présentent l'analyse descendante à l'aide du problème de l'assemblage de documents [Biondi 84]...

Le domaine choisi doit être suffisamment connu du public visé, et c'est une question de culture. Choisir un domaine dont la connaissance serait insuffisante nous obligerait à enseigner les techniques nécessaires avant de pouvoir faire pratiquer les comportements sur lesquels nous voulons faire réfléchir. Alors mieux vaudrait rester dans le domaine de la programmation !

Notons que le changement de domaine introduit une fantaisie dans le travail et comporte donc un aspect ludique ; il permet au débutant de rompre avec la réalité immédiate. Nous choisissons d'exploiter cette fantaisie en présentant l'activité proposée comme un jeu [Guéraud 86]. L'intérêt de mettre le débutant en situation de jeu est de faire apparaître ou de renforcer une motivation devant un type d'activité qui risque de le dérouter à première vue.

Cela permet aussi à celui qui est devenu "joueur" d'agir spontanément, selon sa propre personnalité, comme il le ferait plus tard en situation de projet et non d'agir en fonction de l'enseignant présent. De plus, nous savons qu'un joueur est très actif et s'investit beaucoup dans le jeu, ce qui permet une relation étroite entre théorie et pratique [Beville 86]. Car, comme le dit Caillois [Caillois 67], "rien autant que le jeu n'exige d'attention, d'intelligence et de résistance nerveuse".

Nous voulons demander au débutant de vivre certains comportements, donc de tenir certains rôles. Les comportements du programmeur sont différents dans les phases d'analyse et dans les phases de programmation proprement dite. Nous souhaitons différencier nettement ces attitudes, même si elles appartiennent, à des moments différents, à une même personne. Pour cela, nous proposons plusieurs rôles. Enfin, soulignons que la programmation est un travail d'équipe qui nécessite une collaboration entre les différents intervenants.

Plusieurs personnes qui tiennent des rôles différents, et qui collaborent dans un jeu, cela conduit à la définition d'un jeu de rôles. Un jeu de rôles réunit en effet un meneur de jeu et des joueurs. Il s'agit pour chacun de se mettre dans la peau d'un personnage, et de vivre à travers lui une aventure, dans le monde fantaisiste mais cohérent qui lui est proposé. L'objectif est collectif, il s'agit de mener à bien une mission précise [Paunin 86].

UN JEU DE RÔLES DANS L'ENSEIGNEMENT DE LA PROGRAMMATION

Nous avons choisi le domaine de la bande dessinée. Le thème du jeu est la création d'une page de bande dessinée. Le jeu ne consiste pas en une simulation effective de l'activité de création d'une bande dessinée (et ne s'adresse pas aux spécialistes) mais en la simulation de l'activité de programmation schématisée dans le domaine de la bande dessinée [Guéraud 88].

Ce domaine particulier a été choisi d'une part en fonction de l'analogie qui nous est vite apparue avec l'activité de programmation (les étapes de la construction d'un programme et les étapes de la création d'une bande dessinée), d'autre part en fonction du public visé. Nous faisons l'hypothèse que le monde de la bande dessinée est suffisamment banal et connu d'un jeune public français. Il ne s'agit pas de dire que chacun pourrait être un professionnel de bande dessinée, mais que les concepts et le vocabulaire de base sont connus : chacun sait ce qu'est une bulle, une vignette, un gros plan...

Notre objectif pédagogique est de faire réfléchir l'étudiant sur la nécessité et la difficulté du travail de spécifications, ainsi que sur le processus d'analyse descendante, dont l'importance dans notre contexte n'est plus à défendre.

Principe du jeu :

Il s'agit d'un jeu à trois joueurs se répartissant les rôles suivants :

- le client : il propose le synopsis, c'est-à-dire le résumé de l'histoire à raconter et présente les personnages principaux de l'histoire. Il propose donc le cahier des charges.
- le scénariste : il découpe l'histoire en différentes étapes correspondant chacune à une image de bande dessinée. Il définit ainsi, à partir du synopsis, le scénario et le découpage graphique de la page. Il fait donc l'analyse du synopsis et propose une structuration en images élémentaires.
- le dessinateur : il réalise la page de bande dessinée à partir du scénario et du découpage, en utilisant une bibliothèque de dessins. Il programme (dessine) l'algorithme (le scénario).

Les trois joueurs doivent coopérer dans le but de produire une page de bande dessinée qui réponde à la demande du client, tout en préservant leur intérêt propre (intérêt financier, intérêt de la création, rapport temps/prix).

Chaque joueur dispose d'un micro-ordinateur ; les trois ordinateurs sont connectés. La communication entre les joueurs s'effectue seulement via ce réseau avec la hiérarchie client-scénariste-dessinateur. Il y a deux types de communication. D'une part, les joueurs échangent des documents (synopsis, scénario...) et fixent des accords (délai, prix...). D'autre part, ils communiquent librement mais toujours selon la hiérarchie et via le réseau, pour demander des précisions, négocier, demander un avis, consulter un "fournisseur"... N'oublions pas que si les paroles s'envolent, les écrits restent! Personne n'a le droit de revenir sur un accord déjà conclu. Si l'un des acteurs change d'avis, il doit renégocier et de nouveaux accords doivent être formulés. A tout moment chacun peut poser une question, ou négocier une modification.

Déroulement du jeu :

Les joueurs assistent à une présentation animée du jeu et des fonctions de chaque personnage. Après cette présentation (facultative), les joueurs se mettent d'accord pour la répartition des rôles et le temps consacré au jeu. Chaque joueur définit les différents éléments qui caractérisent son rôle (niveau dans sa profession, spécialisation éventuelle dans un genre, ancienneté...). Il se retrouve alors dans le lieu d'exercice de sa fonction.

Chacun dispose sur son écran d'un environnement de jeu dont une grande partie est commune aux 3 joueurs : téléphone (pour contacter les autres joueurs), horloge, divers indicateurs, le petit manuel des règles d'or de la bande dessinée, un catalogue de produits réalisés. L'environnement de création est spécifique à chaque rôle ; il est constitué d'outils permettant la création des documents dont le joueur a la responsabilité, et d'outils de consultation pour les documents fournis par les autres joueurs.

Et la partie peut commencer ! Le bilan en sera fait à la fin du jeu, en proposant un score global prenant en compte l'état d'achèvement de la page dessinée, le temps nécessaire pour parvenir à cet état, les bilans financiers des joueurs, la satisfaction de chacun par rapport au déroulement de la partie et par rapport au comportement des autres.

Au coeur du jeu :

Qui dit jeu de rôle, dit meneur de jeu. Sa présence est nécessaire pour donner les informations, arbitrer, animer la partie. Il doit intervenir, à bon escient, pour que le jeu se déroule bien ; il a la charge de relancer la partie si elle manque d'intérêt, de débloquer la situation si un blocage se produit, d'imposer un juste milieu entre l'attitude la plus collective qui risque d'engendrer des parties un peu trop sages et le chacun pour soi absolu qui tournera au règlement de compte permanent [Paunin 86]...

Où se cache donc notre meneur de jeu ? Derrière l'ordinateur ? Dans la peau de l'un des joueurs ?

L'ordinateur a un rôle d'arbitre. Il contrôle certaines cohérences, que ce soit à l'intérieur de la création de chacun des joueurs (par exemple, que le scénariste décrive autant de scènes dans le scénario qu'il découpe d'images sur la page graphique) ou entre les travaux des divers joueurs. Il enregistre les documents et accords intermédiaires des partenaires. Il contrôle les différents compteurs relatifs au jeu...

L'ordinateur a un rôle d'animateur. Il génère divers événements (par exemple, au dessinateur : "Vous venez de renverser de l'encre sur votre dessin", "Vous êtes obligé de vous absenter pour cause de maladie"...), de façon aléatoire, ou en fonction de certains aspects du déroulement du jeu.

L'ordinateur a aussi un rôle d'informateur, tout d'abord par la mise à jour des différents indicateurs qu'il gère. Il est capable de donner des informations de base sur les différents métiers de la bande dessinée et de fournir des exemples de produit déjà réalisés (du synopsis à la page dessinée).

En cela, l'ordinateur est meneur de jeu. Cependant il n'a pas tous les moyens d'un meneur de jeu humain. Par exemple, il ne contrôle pas totalement le respect des règles de communication (un client peut toujours aller discuter de vive voix avec un dessinateur), ni la cohérence totale entre l'histoire demandée au départ et la page finale de dessins (cohérences partielles seulement). Il n'a pas la clairvoyance nécessaire pour pouvoir

toujours relancer par ses actions l'intérêt du jeu notamment d'un point de vue pédagogique. L'ordinateur ne remplace pas le meneur de jeu humain, il l'aide et le relaie seulement.

Nous avons vu que le meneur de jeu n'a pas un rôle naïf. Il doit avoir la maîtrise du jeu. Il ne doit pas perdre de vue l'objectif du jeu, qui est la pratique de certaines méthodes et de certains comportements nécessaires à la programmation. Il va devoir agir en fonction de ces comportements attendus. Tel joueur n'utilise pas tel comportement, le meneur de jeu doit intervenir pour que le joueur soit mis devant ce manque, cette erreur, non pas de façon théorique, mais bien par la pratique ; il sera pénalisé dans le jeu par une perte de temps ou d'argent, non pas sur décision du meneur de jeu, mais bien par la conséquence de son erreur... Tel autre joueur utilise telle attitude utile en programmation, le meneur de jeu va agir de telle façon qu'il puisse en tirer tous les bénéfices.

Mais alors qui est mieux placé que le client pour intervenir de cette manière ? Il peut définir un synopsis suffisamment "riche" pour pouvoir donner lieu à des pratiques intéressantes. Il peut, en fonction de l'attitude de chacun, demander une modification du produit afin de faire apparaître le bien-fondé de tel comportement ou au contraire la nécessité de changer de façon de procéder, que ce soit pour le scénariste, ou pour le dessinateur, occupés par le souci de mieux jouer.

En conséquence, nous donnons le rôle du client à l'enseignant, car il saura tirer pleinement profit de la situation qui est en train de se jouer. En outre, l'enseignant disposera de tous les éléments pour discuter ensuite, avec ses élèves, de ce qui s'est passé au cours de telle ou telle partie, et des leçons que l'on peut en tirer pour la programmation.

"TOI, MOI ET LUI", ou l'analogie avec le travail de programmation.

Toi

Scénariste, tu viens de recevoir un synopsis des mains du client. Garde-toi bien de foncer tête baissée ! Regarde bien ce document ! Sache y débusquer les imprécisions, les ambiguïtés. N'hésite pas à contacter ce client, à le cuisiner un peu. Il faut que tu en saches le plus possible avant de mettre les mains à la pâte. Après il risquerait d'être trop tard.

Ne te contente pas d'accords susurrés dans l'écouteur, ton client a peut-être la mémoire courte ! Ne t'engage pas dans n'importe quelle aventure, assure-toi d'avoir les ressources (personnelles, financières, humaines) et le temps nécessaire pour en venir à bout !

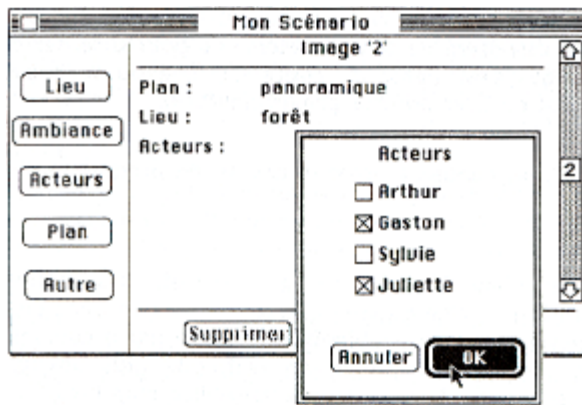
Lorsque tu commenceras à prendre l'histoire à bras le corps pour lui donner vie, ne va pas t'isoler avec le dessinateur dans la grande tour d'ivoire ! Votre création y serait très libre mais vous risqueriez de tomber de haut en retournant proposer votre oeuvre au client qui n'y reconnaîtra sûrement pas ses petits !

Tu peux improviser une scène et demander au dessinateur de la mettre en image. Puis une autre, et encore une autre... tacher de recoller les morceaux tant bien que mal et vite refile le monstre au client. Tu peux aussi te faire un plan d'action, c'est à dire un scénario complet, demander l'accord du client puis passer le relai au dessinateur. A toi de choisir tes risques !

Tu as du mal à trouver du premier coup une décomposition de l'histoire qui soit juste, équilibrée, rythmée ? Mais c'est le contraire qui serait bien étonnant ! Pas de fée secourable ? Alors cent fois sur le métier, remets ton ouvrage !

A l'heure de décrire chaque scène pour le dessinateur, souviens-toi des ennemis qu'il a fallu débusquer dans le synopsis qui paraissait pourtant bien innocent. Essaie de ne pas laisser s'installer leurs complices dans ton propre scénario.

A cette étape, tu trouveras des outils. Ils t'aideront à ne pas vouloir exprimer tous les niveaux de détails à la fois. Ils te suggèrent de planter d'abord le décor général de la scène : où, qui...



Ensuite, ils te suggèrent de donner son rôle précis à chacun des acteurs de la scène.

	Image 2, Gaston
Position	
Vue de...	Vue de : Face
Expression	Expression : joie
Action	Action : se promène
Dialogue	
Autre	Autre : ne porte pas ses lunettes
	Annuler OK

Ces outils sont en général spécialisés parce qu'ils répondent à un besoin habituel du scénariste. Résiste à l'envie de vouloir te servir de tous, ne choisis que ceux qui répondent à ton besoin particulier. Enfin, tout au fond de l'atelier, tu trouveras un dernier outil, très rustique ; ne le méprise pas, il te permettra d'exprimer certaines choses non prévues mais essentielles pour ta scène !

Et maintenant tu peux ôter ton masque, ami scénariste, nous t'avons vu à l'oeuvre, nous avons reconnu tes soucis d'analyste !

Moi

Moi, je suis un dessinateur un peu particulier. Mes outils ne sont pas le crayon et la gomme, mais les ciseaux et la colle. Mes dessins sont tous constitués à partir de dessins de base que je trouve dans ma banque de dessins ! Cela perd en originalité, mais est assez efficace. Pourquoi toujours tout réinventer ?

Je suis un dessinateur malin ! Je ne l'ai pas toujours été mais depuis que je joue à ce jeu-là, j'ai beaucoup progressé ! J'ai appris à ne pas m'occuper de tous les niveaux de détails à la fois ! Par exemple, j'ai pris l'habitude de préparer à part certaines portions de dessins. C'est plus facile, plus clair et je gagne en souplesse car je sais bien qu'on finit toujours par me demander des modifications ! De plus, j'économise ainsi mes efforts, dans le cas de certains décors qui figurent plusieurs fois dans la page, même si je dois changer les dimensions de ces décors, car mes outils me le permettent !

Une petite chose encore ! Je n'oublie pas de prévoir la place des bulles au fur et à mesure de la construction de l'image, sinon je n'arrive plus à les insérer ! Il paraît que je ferais un bon programmeur ! C'est « lui » qui me l'a dit.

Lui

Lui a déjà révélé son identité, ou plutôt sa triple identité : client, meneur de jeu, enseignant.

Comme nous l'avons vu précédemment, son souci est de faire pratiquer des méthodes ou comportements utiles en programmation. Il va agir en tant que meneur de jeu - client dans ce but. Il doit maintenir à la fois l'intérêt ludique et l'intérêt pédagogique de la partie. Il lui faudra maîtriser la partie, faire preuve de fermeté et de diplomatie, être capable d'improviser... Il ne devra sombrer ni dans un non-interventionisme total, ni dans une omniprésence étouffante.

Mais ne devrait-on pas pouvoir en dire autant de tout enseignant en situation de cours ?

POUR CONCLURE

Une première version du jeu est réalisée sur des Apple Macintosh. Son intérêt principal est de nous permettre d'expérimenter son insertion dans un cursus informatique et d'en tirer à notre tour des enseignements.

Le jeu tel qu'il est conçu ne constitue pas, de toutes manières, une fin en soi. Le débutant n'apprendra pas à programmer en 10 parties ! Nous proposons ce jeu comme un exemple d'outil que les enseignants de programmation peuvent utiliser, au sein de leur enseignement. Nous permettons ainsi aux élèves de pratiquer les « comportements » de façon significative en lui faisant jouer les différents rôles au cours de plusieurs parties. Enfin, nous espérons que tous, enseignants et étudiants, se sentent suffisamment à l'aise dans le domaine de la bande dessinée pour en tirer plaisir et profit.

BIBLIOGRAPHIE :

- [Arsac 80] J. ARSAC : *Premières leçons de programmation*, Cedic/Fernand Nathan, 1980
- [Beville 86] G. BEVILLE : *Jeux de formation*, Les Editions d'Organisation, 1986
- [Biondi 84] J. BIONDI, G. CLAVEL : *Introduction à la programmation*, Masson 1984
- [Caillois 67] R. CAILLOIS : *Les jeux et les hommes*, Gallimard, 1967
- [Dijkstra 71] E.W. DIJKSTRA : *A short introduction to the art of programming*, EWD 316, Department of Mathematics, Eindhoven (Pays-Bas), 1971
- [Ducrin 84] A. DUCRIN : *Programmation*, Dunod, 1984
- [Gram 86] A. GRAM : *Raisonner pour programmer*, Dunod, 1986
- [Guéraud 86] V. GUÉRAUD, S. PAINVIN : *Voulez-vous jouer avec moi ?*, LGI, Projet Arcade, Rapport N°5, Université J. Fourier - Grenoble 1, 1986
- [Guéraud 88] V. GUÉRAUD : Thèse en cours, Université J. Fourier - Grenoble 1, 1988
- [Laurent 85] J.P. Laurent : *Initiation à l'analyse et à la programmation*, Dunod, 1985
- [Lucas 83] M. LUCAS, J.P. PEYRIN & P.C. SCHOLL : *Algorithmique et représentation des données - Tome 1 : Files et automates d'états finis*, Masson, 1983
- [Ledgard 75] H.F. LEDGARD : *Programming proverbs*, Hayden Book Company, 1975, traduit en français et annoté par J. ARSAC, Dunod 1978
- [Meyer 78] B. MEYER, C. BAUDOIN : *Méthodes de programmation*, Eyrolles, 1978
- [Paunin 86] T. PAUNIN, P. ROSENTHAL ET G. SAGOT : *N'ayez plus peur des jeux de rôle*, Jeux et stratégies N°38, Avril-Mai 1986
- [Scholl 87] P.C. SCHOLL, J.P. PEYRIN : *schémas algorithmiques fondamentaux*, IGEI, Université J. Fournier - Grenoble 1, 1987