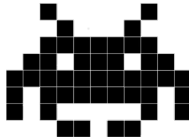


# PLM: Programmer's Learning Machine

## L'exerciseur du programmeur

Martin Quinson

3 décembre 2014



# La g n se de PLM: mon exp rience d'enseignant

- ▶ Enseignant depuis ma ma trise, depuis 2005   Telecom Nancy
- ▶ D'abord sur mes recherches, mais faible niveau des  l ves en programmation
- ▶ Depuis 2006: bases de l'algo-prog (algo, Java, scala, C, shell, syst mes)

Constat: seuls ceux qui savent programmer avant savent apr s

- ▶ Les autres ont des probl mes avec *if* au bout de 6 mois!

Comment aider ceux qui en ont besoin?

- ▶  normes diff rences de niveaux en entr e: ceux qui savent trustent la parole
- ▶ Expliquer les bases de base : aussi difficile qu'enseigner   faire du v lo
- ▶ *Si seulement ils pouvaient bosser par eux-m mes...*
- ▶ *... si seulement nos TP pouvaient  tre moins arides et plus attirants*

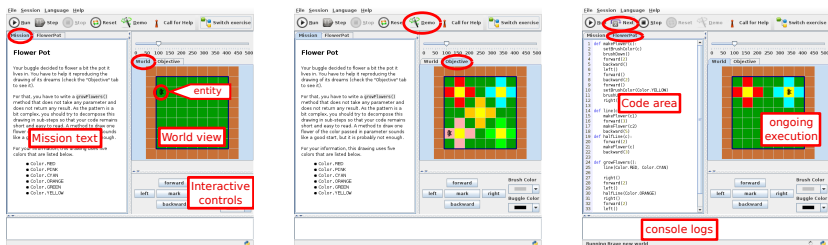
PLM: Programmer's Learning Machine (2008)

- ▶ D velopp e (avec G. Oster) pour une mise   niveau pseudo-autonome tutor e
- ▶ Inspir e des Buggles de F. Turbak – Wellesley College

# La PLM aujourd'hui

## Exerciseur interactif dédié à la programmation

- ▶ Outil interactif et graphique pour apprendre à coder
- ▶ C'est en forgeant qu'on devient forgeron (et qu'on apprend à aimer ça)



## Usage classique

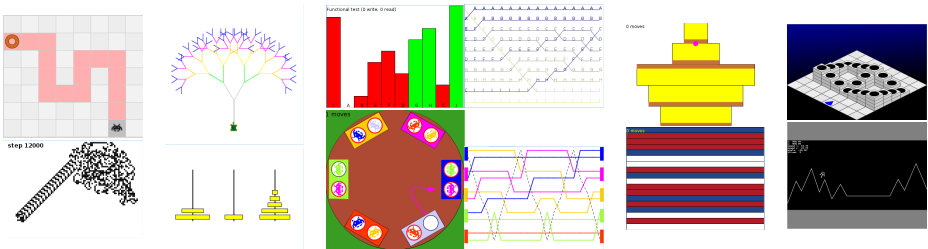
- ▶ On lit la mission à gauche, on compare à droite l'état initial et l'état désiré
- ▶ On tape le code, on clique sur un bouton, et ça s'anime à droite
- ▶ Boucle de *feedback* très courte (et motivante pour les élèves)

# Bonnes propriétés de PLM

- ▶ L'interface est multi-lingue anglais/français (+ brésilien/italien en cours)
- ▶ Les exercices sont multilingues Java/Scala/Python (+ C en cours; js/ruby?)
- ▶ Mode démo, exécution pas à pas, vitesse d'animation pas à pas, sessions
- ▶ Plusieurs mondes parallèles pour mieux tester le code élève
- ▶ Documentation embarquée dans l'outil

## Différents types d'univers pour différentes situations-problèmes

- ▶ Micro-mondes génériques; Problèmes classiques; Tris; Jeux; (JUnit-like)
- ▶ Des vues d'état et des vues temporelles; Des interactions à la souris



- ▶ Ajouter un univers: 100 à 300 lignes de code (Java ou Scala)

# Notre progression pédagogique en CS0

## Objectif: *tactical programming*

- ▶ Le B-A-BA de la programmation: écrire des trucs de base sans difficulté
- ▶ Que ce ne soit plus un problème quand les algos seront non-triviaux

## Implémentation agnostique point de vue langage (Java, Python, Scala)

- ▶ Instructions (et commentaires)
- ▶ Conditionnelles (40 exercices Bat pour savoir écrire des tests logiques)
- ▶ Boucles *while*; *switch cases* (+ 3 exos d'applications)
- ▶ Les variables
- ▶ Boucles *for* et *do/while*
- ▶ Méthodes (+5 exos qui forcent à factoriser le code)
- ▶ Méthodes avec résultat, avec paramètres
- ▶ Tableaux (+12 exos d'applications, certains en Bat)
- ▶ Application: parcours de tableau 2D (5 exos) les turmites (turing 2D – 4 exos)

190 exos scénarisés, pour plus de 50h de bonheur <3

# C'est même documenté dans la PLM directement

	Instructions	Comments	Conditionals	While loops	Variables	For loops	Do/While loops	Methods	Switch	Arrays
Welcome in the Baggles' World		✗	✗	✗	✗	✗	✗	✗	✗	✗
Java Instructions		✗	✗	✗	✗	✗	✗	✗	✗	✗
Writing more complex programs			✗	✗	✗	✗	✗	✗	✗	✗
Conditional instructions	✓	✗		✗	✗	✗	✗	✗	✗	✗
While loops	✓	✗	✓		✗	✗	✗	✗	✗	✗
Baggle Seeking	✓	✗	✓		✗	✗	✗	✗	✗	✗
Storing and manipulating data	✓	✗	✓	✓		✗	✗	✗	✗	✗
For loops	✓	✗	✓	✓	✓		✗	✗	✗	✗
Do .. while loops	✓	✗	✓	✗	✗	✗		✗	✗	✗
Methods	✓	✗	✓	✓	✗	✗	✗		✗	✗
	Instructions	Comments	Conditionals	While loops	Variables	For loops	Do/While loops	Methods	Switch	Arrays
Building methodically	✓	✗	✗	✗	✗	✗	✗		✗	✗
Methods returning a result	✓	✗	✓	✗	✓	✓	✗		✗	✗
Methods with parameters	✓	✗	✓	✗	✗	✓	✗		✗	✗
Methodically drawing	✓	✗	✗	✗	✗	✗	✗		✗	✗
Methodically drawing (only bigger)	✓	✗	✗	✗	✗	✓	✗		✗	✗
Drawing bigger and bigger	✓	✗	✗	✗	✗	✓	✗		✗	✗
Even more pattern to draw	✓	✗	✗	✗	✗	✓	✗		✗	✗
Baggle Dance Revolution	✓	✗		✓	✗	✗	✗	✗	✗	✗
Baggle Dance Revolution 2	✓	✗		✓	✗	✗	✗	✗		✗
Slug Hunting	✓	✗	✓	✓	✗	✗	✗	✓	✗	✗
Slug Tracking	✓	✗	✓	✓	✗	✗	✗	✓	✗	✗
	Instructions	Comments	Conditionals	While loops	Variables	For loops	Do/While loops	Methods	Switch	Arrays
Snake World	✓	✗	✓	✓	✓	✗	✗	✓	✗	✗
Knitting and Arrays	✓	✗	✓	✗	✗	✓	✗	✓	✗	
Knitting, Arrays and modulus	✓	✗	✓	✗	✓	✓	✗	✓	✗	
Traversal by column	✓	✗	✓	✓	✓	✗	✗	✓	✗	
Traversal by line	✓	✗	✓	✓	✓	✗	✗	✓	✗	
Zig-zag traversal	✓	✗	✓	✓	✓	✗	✗	✓	✗	
Diagonal Traversal	✓	✗	✓	✓	✓	✗	✗	✓	✗	

Malheureusement, ceci n'est pas généré automatiquement (pas toujours à jour)

# Point de vue d'enseignants

## Usages en production

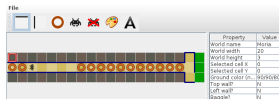
- ▶ Telecom Nancy: utilisé dans plusieurs modules (même pendant ma délég')
- ▶ Beaucoup d'intérêt, quelques usages disparates chez les collègues

## Avant de l'utiliser chez vous

- ▶ Utiliser un tel outil demande un **{gros travail de préparation}**  
Tester en salle de TP, faire les exercices, anticiper les questions
- ▶ Un plan B en cellulose pour la première fois, mais c'est plaisant après

## Créer des ressources (exercices, leçons et univers)

- ▶ Très simple d'ajouter des ressources, mais très peu l'ont fait à part nous :-/
- ▶ Un mode *édition* est en travaux (papier à musique des exercices de prog)



# PLM et le métier d'enseignant

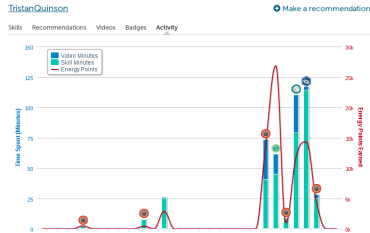
- ▶ En médecine, le monitoring des patients **aide au diagnostic** du problème
- ▶ Les enseignants **travaillent en aveugle** et demandent aux apprenants
- ▶ PLM libère le temps d'aider ceux en difficulté, mais on ne sait plus qui c'est



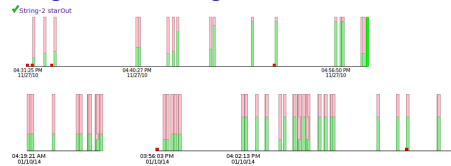
# PLM et le métier d'enseignant

- ▶ En médecine, le monitoring des patients **aide au diagnostic** du problème
- ▶ Les enseignants **travaillent en aveugle** et demandent aux apprenants
- ▶ PLM libère le temps d'aider ceux en difficulté, mais on ne sait plus qui c'est

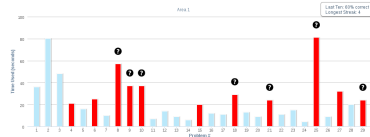
## Activité sur Khan Academy



## Progrès sur CodingBat



## Problèmes sur un exo (Khan)



## Détail du problème (Khan)

Area 1

Make a recommendation

Previous problem: Previous step: 1/10 - 81% - 100% - 100% - 100% - 100% - 100% - 100% - 100% - 100%

Next step: Next problem

What is the area of the rectangle?

Area 1

Area of rectangles and the distributive property

Comparing area of parts of a set

Calculating the square footage of a house

Next section: Previous

Answer:

Check answer

Show me how to solve it

Watch a video

8/15

# Projet de recherche autour de PLM

## Objectif: PLM comme terrain d'expérimentations

- ▶ **Pédagogie:** quelle est la bonne façon d'enseigner la programmation?
- ▶ **Modélisation:** gamification, tuteur intelligent
- ▶ **Data-mining:** erreurs corrélées, corrélation au *background* des élèves, erreurs typiques (identification d'obstacles didactiques)
- ▶ **Collaboratif:** susciter la collaboration sans triche

## Déjà fait

- ▶ 2 ans ingénieur financés par le MOOC lab Inria
- ▶ Des idées pour 5 ans au moins
- ▶ Sauvegarde anonyme de toutes les version intermédiaires à tous les exos
- ▶ Reste à moissonner <https://github.com/mquinson/PLM-data>

# Premières mesures de moisson (1/2)

(Ces données dates de cette nuit, désolé pour leur format très brut)

## Volumes totaux

- ▶ 188185 evts dans 742 traces (potentiellement multisession), 15190 exos passés

## Tailles des traces disponibles

- ▶ En nombre d'exercices différents réussis ou tentés

Seuil nombre exos	190+	150+	100+	50+	25+	10+	5+	1+
Nb traces ( <i>exos passés</i> )	1	6	24	66	98	209	365	661
Nb traces ( <i>exos tentés</i> )	1	8	28	73	112	258	420	742

- ▶ En nombre de lignes dans des exercices réussis

Seuil nb lignes	4000+	3000+	2000+	1000+	500+	250+	50+	10+
Nombre traces	0	2	13	53	88	146	384	560

# Dynamique au cours du temps

**Légende:** Nombre de (exercices réussis / traces actives) sur la période

<i>Mois</i>	<i>Total mensuel</i>	<i>Limité au Java</i>	<i>Python</i>	<i>Scala</i>
Septembre	10314/ 337	280/ 32	303/ 17	9731/307
Octobre	2960/ 333	669/ 72	1136/145	1155/149
Novembre	1333/ 106	314/ 19	944/ 75	75/ 14
Décembre	49/ 14	19/ 3	23/ 8	7/ 3

<i>Sem</i>	<i>Total</i>	<i>Lundi</i>	<i>Mardi</i>	<i>Mer</i>	<i>Jeu</i>	<i>Ven</i>	<i>Sam</i>	<i>Dim</i>
1 sept	96/ 5	4/ 2	6/ 1	8/ 2	51/ 4	27/ 1	0/ 0	0/ 0
8 sept	5816/164	1311/ 90	1639/ 89	187/ 20	1039/ 68	1281/ 74	168/ 20	191/ 14
15 sept	2832/102	1293/ 77	851/ 63	326/ 44	80/ 6	61/ 6	81/ 8	140/ 9
22 sept	1055/124	435/ 55	219/ 32	51/ 6	221/ 31	50/ 8	27/ 5	52/ 9
29 sept	1529/220	545/ 84	283/ 41	136/ 33	333/ 40	103/ 34	65/ 12	64/ 12
6 oct	742/131	236/ 61	189/ 37	131/ 18	92/ 17	46/ 6	12/ 4	36/ 7
13 oct	572/ 38	50/ 6	93/ 4	88/ 14	52/ 7	71/ 8	141/ 10	77/ 12
20 oct	630/ 72	116/ 18	113/ 13	101/ 18	45/ 8	95/ 16	72/ 17	88/ 15
27 oct	492/ 59	110/ 18	51/ 10	41/ 11	63/ 10	88/ 20	78/ 11	61/ 7
3 nov	516/ 54	34/ 7	71/ 12	53/ 10	228/ 22	42/ 4	24/ 5	64/ 12
10 nov	356/ 46	33/ 4	71/ 17	6/ 4	170/ 21	19/ 3	13/ 3	44/ 6
17 nov	217/ 34	7/ 3	13/ 5	21/ 3	90/ 12	39/ 5	18/ 5	29/ 7
24 nov	128/ 21	5/ 2	11/ 3	16/ 2	7/ 2	3/ 2	40/ 6	46/ 9
1/12	49/ 14	24/ 7	16/ 7	9/ 3	0/ 0	0/ 0	0/ 0	0/ 0

*Ok, ok, c'est super moche, désolé.*

# Quelques pistes de recherche envisagées (1/2)

## Financement Inria (type ADT)

- ▶ 2 ans ingénieur, un stagiaire SHS pour une étude, des missions

### (1) Infrastructure logicielle et génie logiciel: code maintenable

- ▶ Intégration continue, tests unitaires et d'intégration multi-plateformes (fait)
- ▶ Réarchitecturer pour web et tablettes, [Interfaçage](#) avec moodle, edX, etc.

### (2) Modélisation de l'apprenant: Personnalisation, ITS et aide au diagnostic

- ▶ Collecte traces anonymes, données publiées pour la recherche (fait)
- ▶ Infra de modélisation d'apprenants, pour comparer les approches
- ▶ Console enseignant, événements + filtres pour diagnostic
- ▶ Gamification, en donnant des points et des badges quand le modèle avance

## Anonymat des apprenants

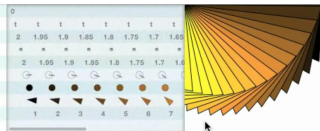
- ▶ Traces collectées anonymes *by design*, librement accessibles pour aider la science
- ▶ Identité d'utilisateur optionnelle, dans une base séparée gérée par l'enseignant

## Quelques pistes de recherche envisagées (2/2)

- (3) Vers une pédagogie expérimentale avec mesures objectives d'efficacité
- ▶ Évaluation des supports grâce aux traces capturées (+ tests A/B)
  - ▶ Remédiation personnalisée pour les masses: détection d'erreurs classiques
  - ▶ Éditeur intégré d'exercices (en cours); **Contenus pédagogiques** algo dist et HPC
- (4) Améliorer l'expérience utilisateur Diffusion accrue pour plus de traces
- ▶ Interactions enseignant/apprenants: « lever le doigt », vidéo-conférence
  - ▶ Interfaces niveau source: Scratch, métriques de qualité; **Debugueur innovant**



```
var i = 0;
while (i < 20) {
  var scaleFactor = 1 + (20 - i)/20;
  resetMatrix();
  scale(scaleFactor);
  rotate(i * 6);
  fill(i * 30, i * 18, 0);
  triangle(0,0, 100,-20, 95,40);
  i += 1;
}
```



- (5) Vers des apprenants tuteurs et acteurs. Co-apprentissage
- ▶ Forum collaboratif gamifié (sans triche); **Commenter et proposer des exos**
  - ▶ Collaborations Peer-review et Pair Programming comme au XXI<sup>e</sup> siècle

# Conclusions

## La PLM veut aider plusieurs publics

- ▶ **Élèves:** apprennent à leur rythme dans un «jeu» sérieux
- ▶ **Enseignants:** simplifie la menée d'exercices, monitoring d'élèves à aider
- ▶ **Auteurs:** réutilisation du code non-fonctionnel, retour par instrumentation
- ▶ (plus les chercheurs sur ces thèmes)

## Pleins de petites idées pour la suite

- ▶ **De nouveaux univers:** backtracking, POO, système, PatchWorld, Robbozzle
- ▶ **De nouveaux langages:** Tous les JSR231 et même le langage C ou scratch
- ▶ **Finitions** de l'interface et de l'ergonomie

## Mais aller vraiment plus loin est un **projet de recherche**

- ▶ **Multidisciplinaire** (tradition de pédagogie de l'informatique à Nancy)
- ▶ **Balbutiant** voire tâtonnant un peu
- ▶ **Mais passionnant !** yapuka

# Apprendre l'informatique par la programmation



Quoi?

Bon sens, idées fortes

36 minutes

Activités débranchées

36 heures

Coding goûters avec *Snap!*

36 jours

Exerciseur interactif


36 semaines

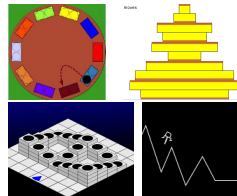
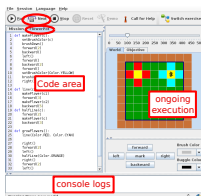
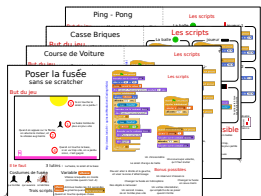
Tous mes cours  sur 

36 ans

Communautés de logiciels libres

Avec qui?

Mise en réseau avec  `je.code()`,





# Dispositifs d'apprentissages

## Mes objectifs en la matière

- ▶ Permettre aux apprenants d'expérimenter, de tester des choses par eux-mêmes
- ▶ Il faut enraciner l'apprentissage dans le réel, dépasser le virtuel
- ▶ Des supports quasi autonomes pour essayer largement

## Pleins de projets

- ▶ SMN: Informatique débranchée, sans ordinateur
- ▶ Coding Goûter: Initiation pour les enfants et leurs parents
- ▶ CodCodCoding: Initiation périscolaire en MJC
- ▶ PLM: Exerciseur du programmeur
- ▶ leaSH: Learning Shell
- ▶ Mes cours (algotprog, prog système, CSH) sur github, avec version enseignant
- ▶ (SimGrid: Simulateur complet d'infrastructures distribuées)
- ▶ jecode.org: forum d'acteurs de terrains pour la mise en réseau
- ▶ Journées ISN-EPI, Formations LAMAP, Groupe IREM Algorithmes, MoocLab