

PLaTon

Platform for **L**earning and **T**eaching **online**

— *un outil pour apprendre et pour enseigner* —

Dominique Revuz - Magdalena Kobylanski

<https://premierlangage.github.io/PLaTon-web/>



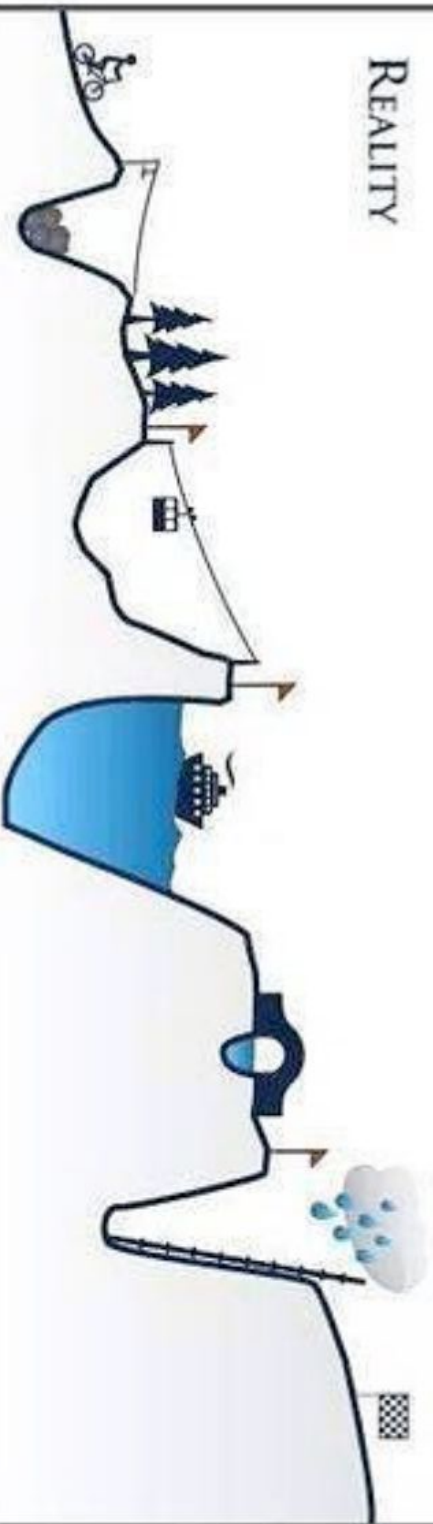
Wims Workshop 2020 - Besançon - 10 Novembre



YOUR PLAN



REALITY





Christophe, Adam, Quentin, Yohan, Kilian, Magdalena, Dominique, Nicolas, Marie, Olivier, Anna,

Des absents : Mamadou, David, Etienne, Damir, Fanny

Résumé

Présentation d'un Successeur potentiel de WIMS.

- les composants d'une plateforme d'enseignement,
- organisation du logiciel,
- langages d'écriture des exercices et des activités.

La questions de l'éditorialisation des ressources: ré-utilisabilité, partage, évolution, curation, édition partagée.

Un point sur l'état des lieux et une roadmap des prochains développement.

Héritage de WIMS



- des **ressources pédagogiques** disponibles, réutilisables, restructurables, modifiables, partagées
- une volonté de proposer des **exercices répétables**
- une partie de la **philosophie d'édition** (faciliter la **création d'exercices** variés, le **partage**, fédérer une **communauté**)
- les **notions** d'exercices, de ressources, de classe, de niveaux de programmation

des **différences** :

- organisation du **code**, **langages**, **capacités fonctionnelles** de la plateforme
(ex : co-édition, mode de partage, processus de montée en qualité)
- WIMS a montré la voie. Notre objectif : **capitalisation** sur ce qui fonctionne,
changements technologiques sur ce qui bloque

Pourquoi Platon

- Un langage de programmation partagé : python3, django, typescript, angular
- La nécessité de protéger les exécutions (informatique)
- Apporter les possibilités des Websockets et de l'interactivité
- Proposer des activités scénarisé (remédiation, révision, découverte)
- Gérer la métacognition
- Gestion de la classe
- Analytics (globales & RGPD)
- Communauté de partage de ressources “à la” github
- LTI / CAS
- Distribué / sécurisé

PLATON 2019-2020 en chiffres

150 107 corrections d'exercice en direct effectuées par les sandbox (24h/24 et 7j/7)
58 984 réponses parfaites des apprenants (note maximale 100/100 obtenue)

1 110 utilisateurs actifs

5239.0 h 35.0 m temps de travail cumulé des utilisateurs sur PLaTon
4.0 h 43.0 m temps de travail moyen pour chacun des 1110 utilisateurs

34 enseignants-éditeurs de ressources

1324 fichiers ressources pédagogiques d'extension .pl
18345 actions d'édition sur les ressources (ajout, suppr. ou modif. de fichier)

31 cours PLaTon créés

615 exercices auto corrigés déployés

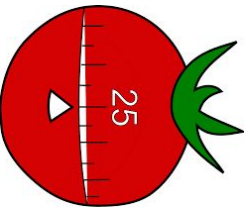
88 activités (fiches de TP) déployées

2 universités sur le serveur MLV : U. Gustave Eiffel et U. Paris Saclay

Les concepts fondateurs de PLaTon

- *Activités / EAO*
- Indicateurs
- Editorialisation / Communauté
- Tri-Force / alignement

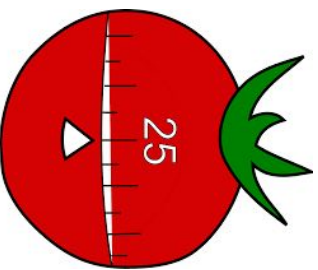
Activités

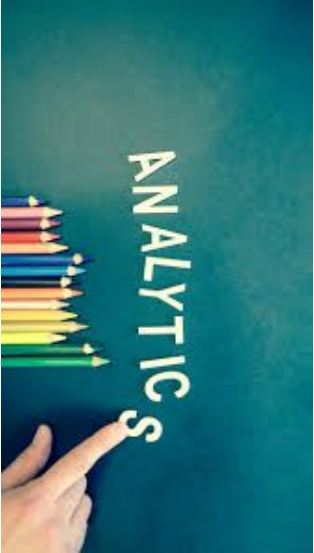


$$\begin{aligned} & \text{Express } \sqrt[n]{2005 + \sqrt{2005^2 - 1}} \\ & \quad \text{as } \sqrt[n]{a} - \sqrt[n]{b} \text{ where } a \text{ and } b \text{ are integers.} \\ & \frac{1}{\sqrt[n]{2005 + \sqrt{2005^2 - 1}}} \quad (\text{let } n = 2005) \leftarrow a \text{ and } b \in \mathbb{Z} \text{ if } n \text{ is odd} \\ & \quad \frac{1}{\sqrt[n]{n + \sqrt{n^2 - 1}}} \\ & \quad \frac{\sqrt[n]{n - \sqrt{n^2 - 1}}}{\sqrt[n]{(n + \sqrt{n^2 - 1})(n - \sqrt{n^2 - 1})}} \quad \left(\text{multiply by } \frac{\sqrt[n]{n - \sqrt{n^2 - 1}}}{\sqrt[n]{n - \sqrt{n^2 - 1}}} \right) \\ & \quad \frac{\sqrt[n]{n - \sqrt{n^2 - 1}}}{\sqrt[n]{n^2 - (n^2 - 1)}} \quad \left(\text{let } (x + y)(x - y) = x^2 - y^2 \right) \\ & \quad \frac{\sqrt[n]{n - \sqrt{n^2 - 1}}}{\sqrt[n]{1}} = \sqrt[n]{n - \sqrt{n^2 - 1}} \\ & \text{Compare } \sqrt[n]{a} - \sqrt[n]{b} = \sqrt[n]{n - \sqrt{n^2 - 1}} : \\ & \quad \left(\sqrt[n]{a} - \sqrt[n]{b} \right)^n = n - \sqrt{n^2 - 1} \quad (\text{square both sides}) \\ & \quad a - \sqrt[n]{a} \sqrt[n]{b} + b = \left(\frac{n}{2} + \frac{1}{2} \right) - \sqrt[n]{\left(\frac{n+1}{2} \right) \left(\frac{n-1}{2} \right)} \\ & \quad (a+b) - 2\sqrt[n]{ab} = \left(\frac{n+1}{2} + \frac{n-1}{2} \right) - 2\sqrt[n]{\frac{n+1}{2} \cdot \frac{n-1}{2}} \\ & \quad a - \sqrt[n]{a} = \frac{n-1}{2} = 1003 \text{ and } b = \frac{n-1}{2} = 1002 \end{aligned}$$



- Pomodoro, Diagnostic, Remédiation,
- Aide à la mémorisation (multi-sensorielle)
- Révision (avec des ontologies)
- Examens, Devoirs “maison”, Ateliers, Peer2peer,
- Attribution - Recommandation Manuelle et automatique
- Travaux de groupes, Binômes
- Animation d’amphi
- Testing rapide en cours / TD / amphis
- Questions ouvertes
- Aide à la correction de copies
- Création de **grilles critériées**
- Atelier- Correction par les pairs
- Activités composés

[illegible]



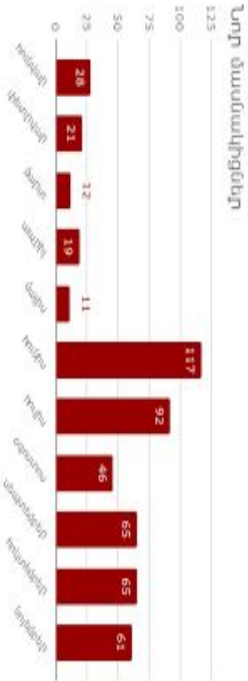
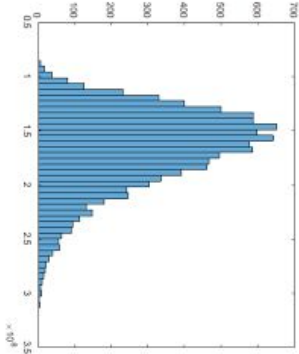
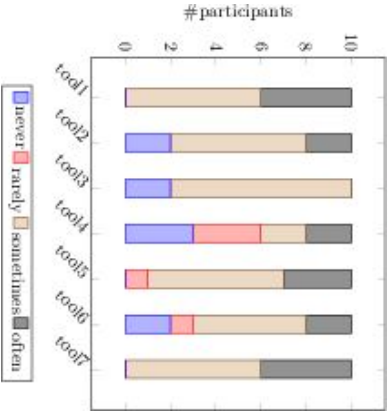
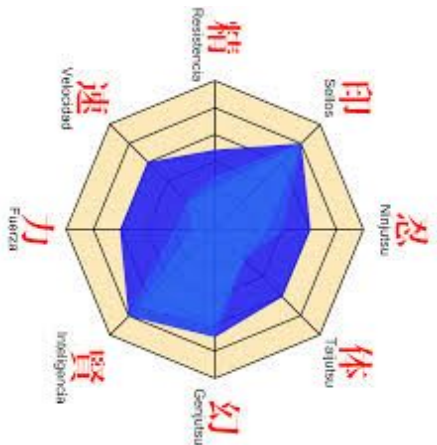
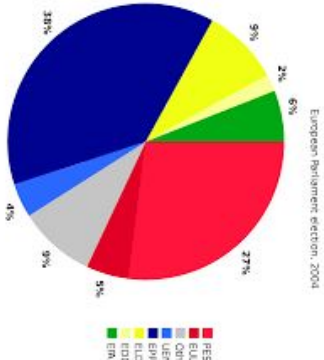
Indicateurs

Temps de travail, duré de l'exercice,
Taux de réussite brut, nombre d'essais,
Positionnement dans le groupe, la classe, la cohorte

Indicateur métacognitifs

(efficacité, procrastination, méthodologie, learning how to learn)

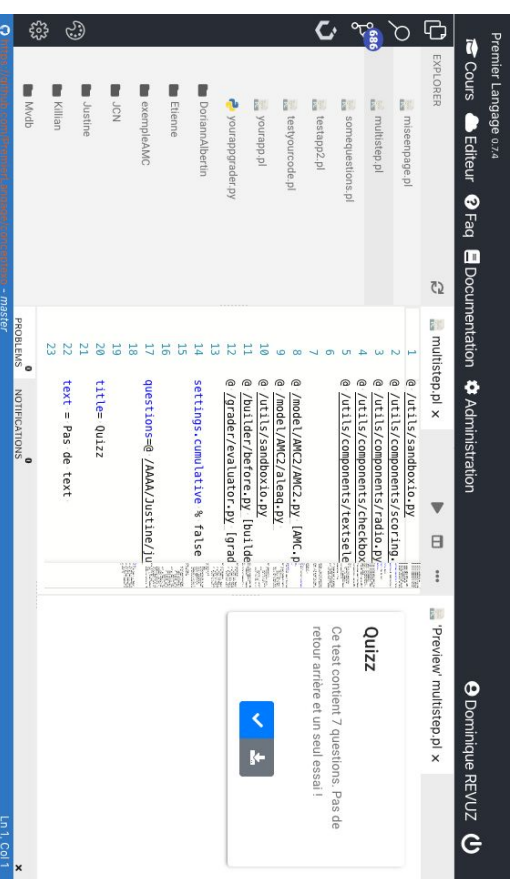
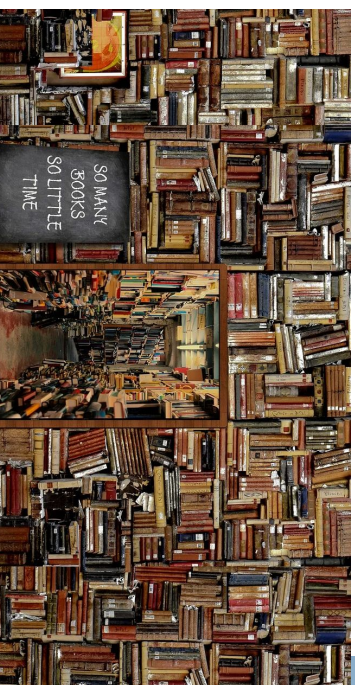
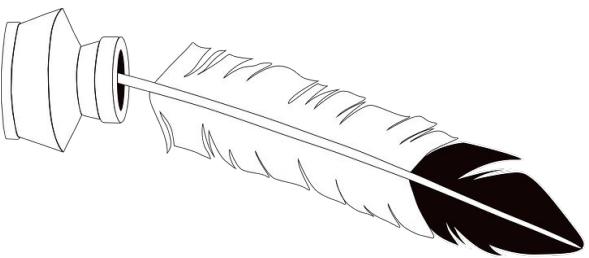
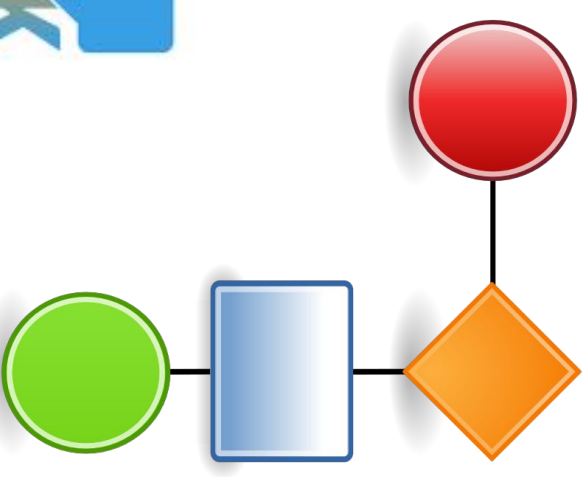
Identification des concepts incompris
Qualification des exercices (taux de réussite, liens avec la compréhension, etc)
Ordonnancement des exercices/ressources (maximisez le taux d'apprentissage)



Editorialisation

- Workflow
- Editeur
- Partage
- Indexation / compétences et AAV
- Qualification
- Curation

Construction de la communauté / Cercles



Alignement Pédagogique

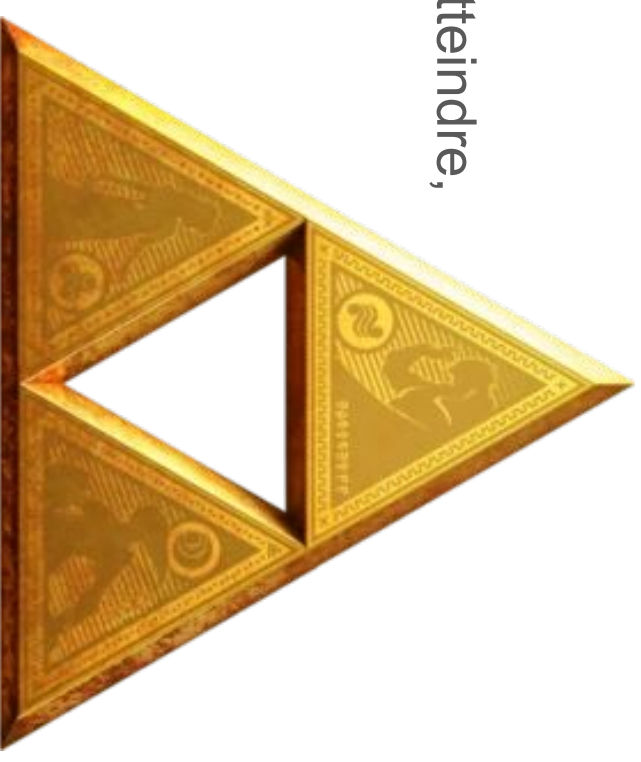
Notre souhait en créant platon est de proposer une plateforme efficace.

Que les étudiants apprennent rapidement des connaissances* qu'ils conservent.

Pour cela il faut 3 conditions (la triforme) :

- que les **objectifs** de l'enseignement soit clairs,
- que les **activités** permettent effectivement de les atteindre,
- que l'on puisse **vérifier** qu'ils sont atteint.

*Connaissance, savoir faire, compétences, what ever, ...



Activités

Solutions ?!

Les exercices

PL-Playexo langage permettant

la réutilisation et l'extension d'exercices.

la composition,

la scénarisation.

Activités Solutions ?!

Des activités variées Composables et réutilisables.

PLA-Playactivity un langage de description d'activités interactives

un serveur interactif playactivity permettant une communication instantané entre les acteurs de l'acte pédagogique.

Solutions ?!

Indicateurs

Pour les indicateurs nous sommes toujours en phase exploratoire, en effet le nombre et la qualité des outils de statistiques et outils graphiques nous mettent dans l'**embarras du choix**.

Notre approche actuelle est de construire des indicateurs élémentaires les plus complets possible et une API d'accès assurant différents niveaux d'accès et d'anonymisation (RGPD).

Editorialisation Solutions ?!

Un gros travail de défrichage de cette problématique à été faite par l'Équipe **WIMS-Ed** (D. Buskulic, E. Sandier, M. Kobylansky).

Les conclusions:

- Creative Commons
- un Workflot basé sur le principe de “diviser pour régner”
- serveur national partagé et ouvert
- édition partagée (ou une édition partagée à la google drive), ou un gestionnaire de version itératif.
- outil de gestion de communautés (chat, groupes, discord, etc)
- outil d'édition et de collecte automatique des métadonnées

TriForce

Solutions ?!

Pour apporter une aide à l'alignement pédagogique, toutes les ressources doivent être liées a des Objectifs de formation.

=> Nous avons choisi comme définition formelle celle des AAV

Acquis d'apprentissages Visés.

Proposé par <http://www.fa2l.be/>



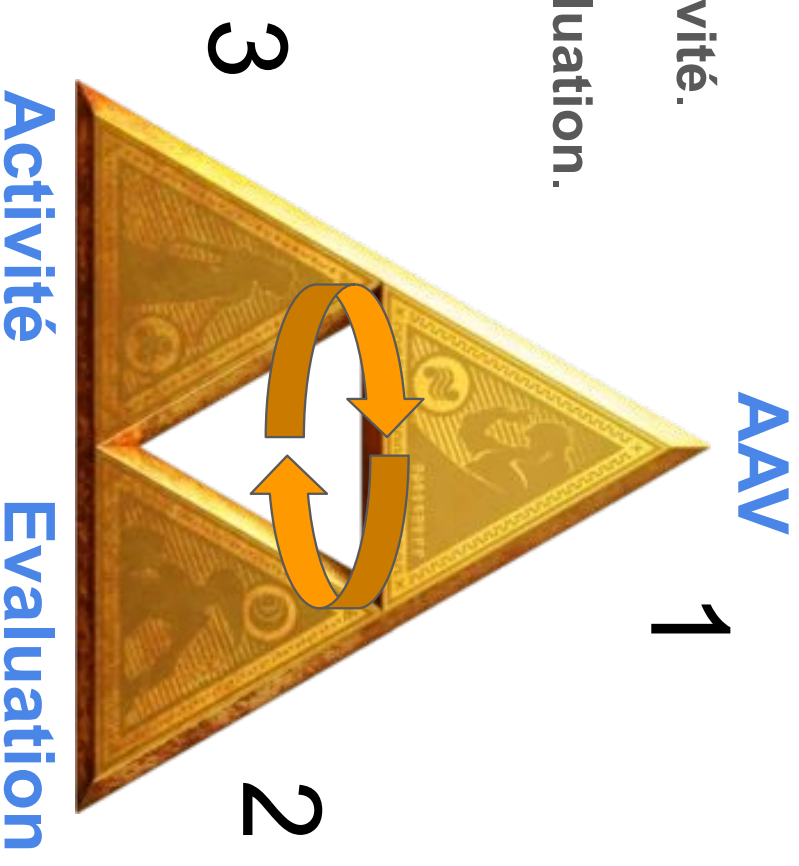
TriForce

Solutions ?!

Toute **ressource** est lié à au moins un **AAV**.

Tout **AAV** est lié à au moins une ressource **activité**.

Tout **AAV** est lié à au moins une ressource **évaluation**.



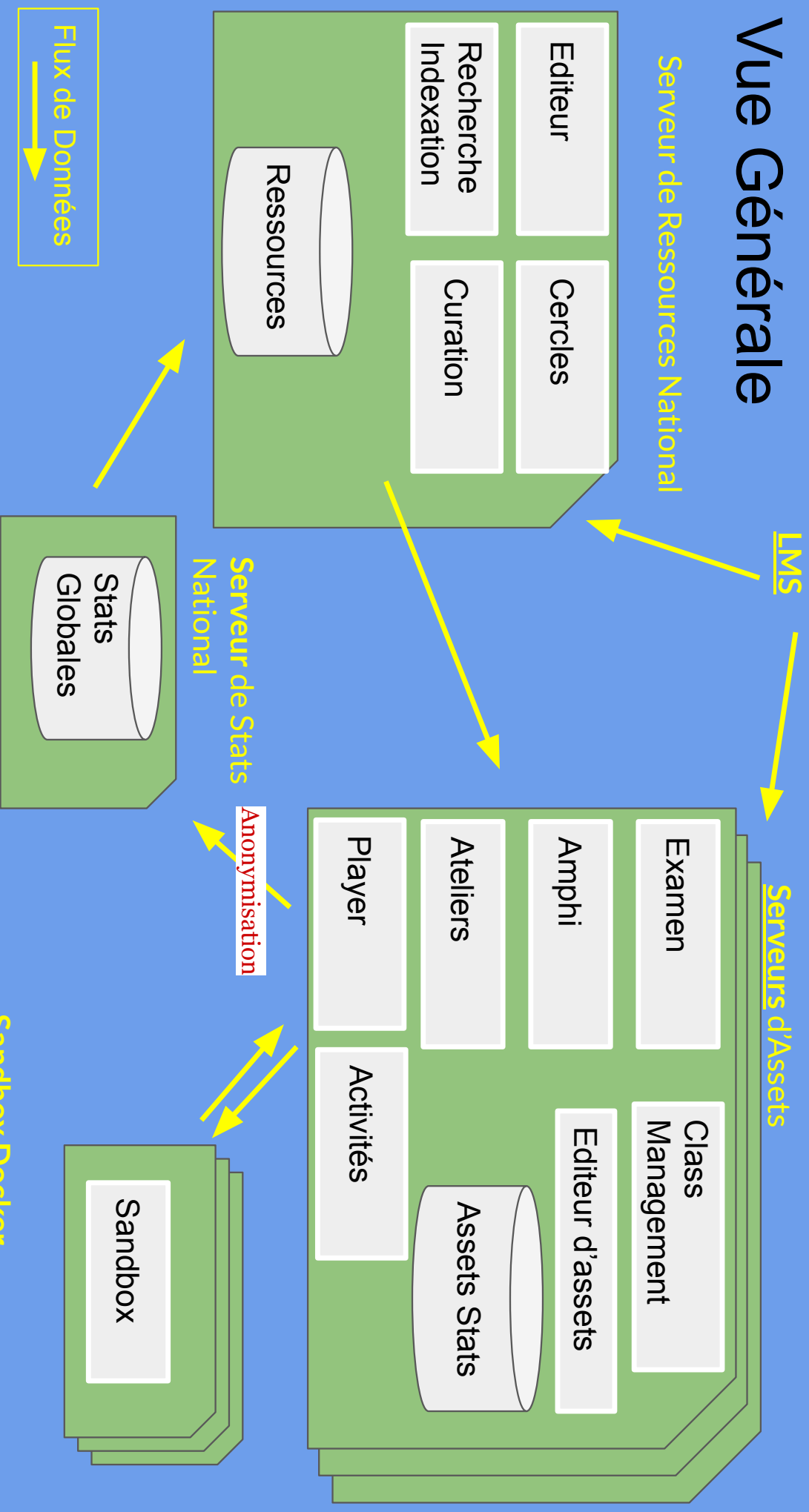
Organisation du logiciel

Pour répondre aux multiples contraintes et besoins exprimées .

Nous sommes partis d'une architecture ouverte avec comme objectif d'avoir une plateforme très accueillante pour de nouveaux usages et permettre un maximum d'utilisation différentes.

L'expérience prouve que les utilisateurs invente des scénarios d'usage originaux.

Vue Générale



Serveur de Ressources National

Editeur / IDE

preview

Highlight

tests

Prepared

Curation

Recherche
Indexation

Tags

Validité

Edition

Cercles

facebook

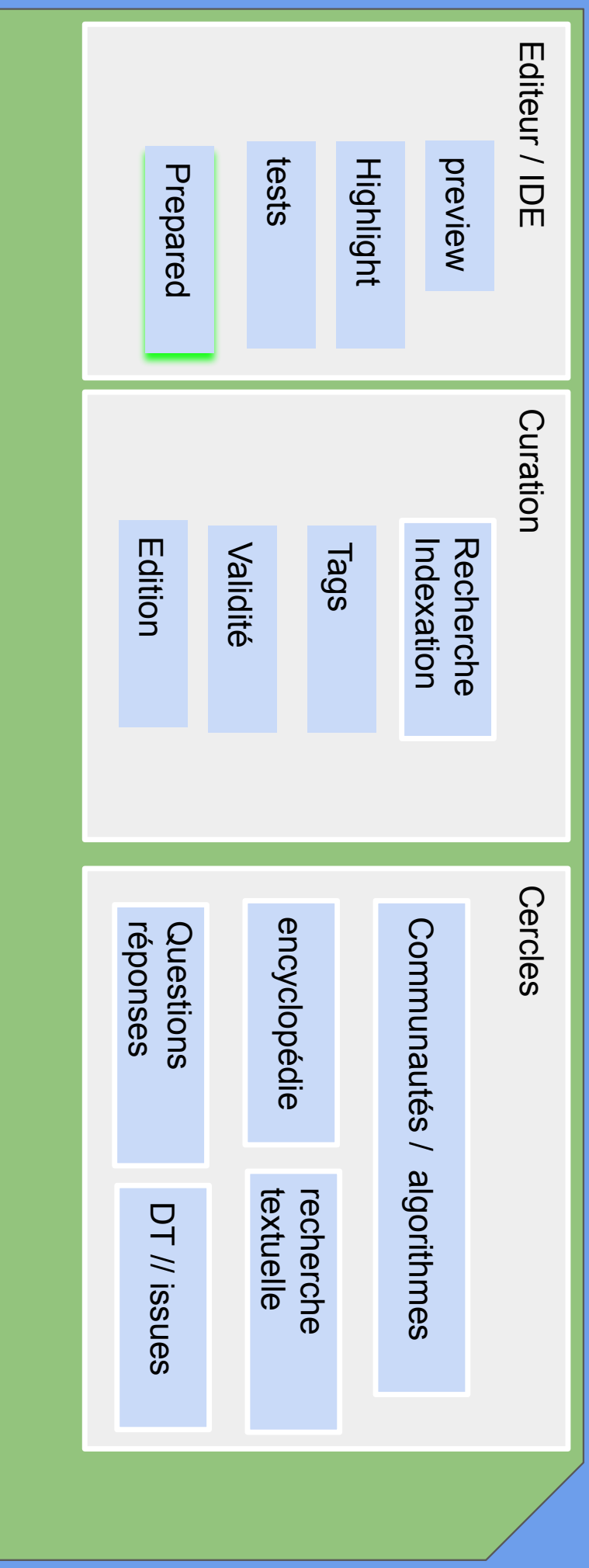
Q & A // Issues

Wikipedia

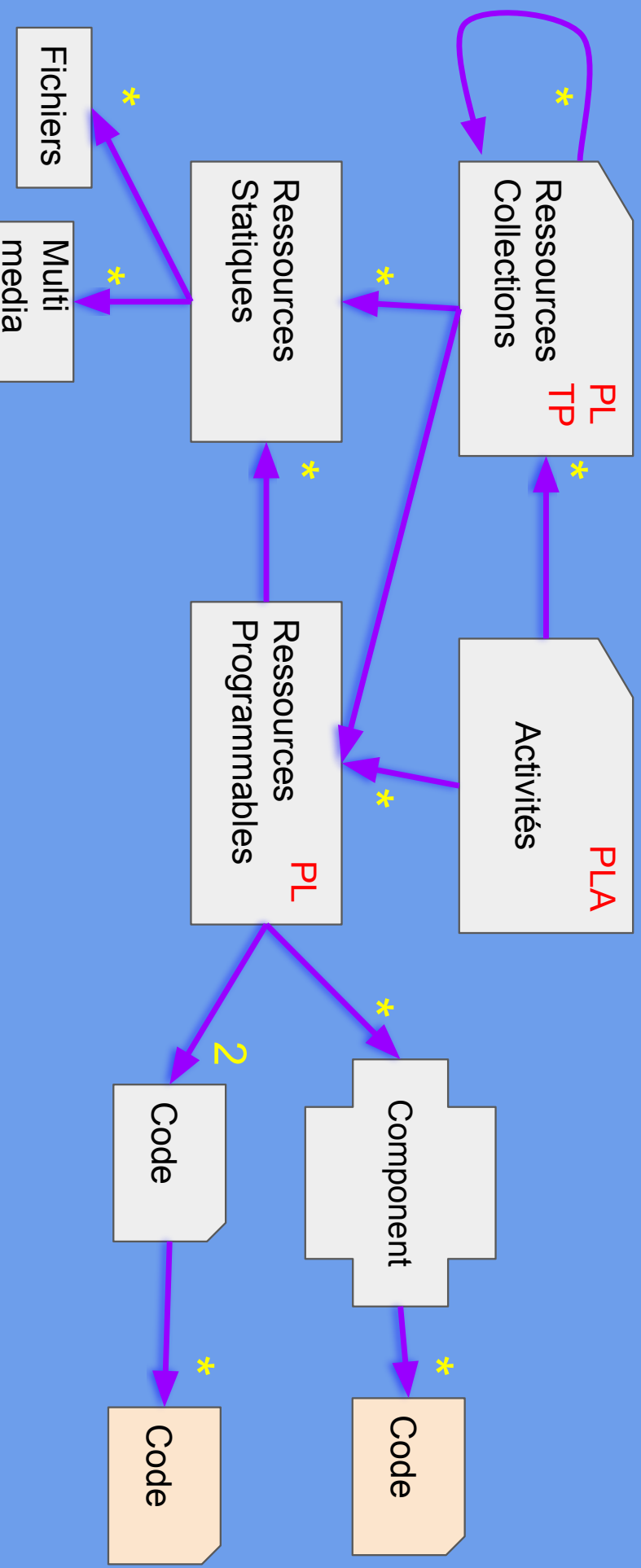
google

Stackoverflow

Serveur de Ressources National



Organisation des ressources



Lien de composition
* Arité multiple

le langage de description d'exercices



Un exercice

```
lasaisie := Input
before=
title= Votre exerciceur !
text==
Veuillez choisir votre exerciceur.
==
form==
{{ lasaisie|component}}
==
evaluator==
if "wims" == lasaisie.value:
    grade = (100, 'Good 🍌🍌🍌')
else:
    grade = (0, 'Bad answer 🍌🍌🍌')
==
```

<https://pl.u-pem.fr/filebrowser/demo/31089/>

Premier Langage 0.7.4

 Dominique REVUZ 

Votre exerciceur !

Veuillez choisir votre exerciceur.

Good 🍌🍌🍌

Answer

wims



Un exercice => Un Modèle

```
# Fichier Saisie Simple
# ssimple.pl

@ /utils/sandboxio.py

grader =@ /grader/evaluator.py
builder =@ /builder/before.py

lasaisie =: Input
lasaisie.type = text
lasaisie.placeholder = Answer
lasaisie.appearance = outline
form==
{{ lasaisie|component }}
```

```
# suite

evaluator==
if good == lasaisie.value:
    grade = (100, 'Good 🌟🌟🌟')
else:
    grade = (0, 'Bad 🙄🙄🙄')
==
```

```
extends= @ ssimple.pl

before=
title= Votre exerciceur !
text==
Veuillez choisir votre exerciceur.
==
good=wims
```

Une fois que l'on a un modèle il est facile de le réutiliser.

extends= @ ssimple.pl

before=

title= Fleuve

text==

Plus long fleuve de france

==

good=Rhin

extends= @ ssimple.pl

before=

title= Chimie

text==

Nom usuel du **monoxyde de**

dihydrogène .

==

good=eau

extends= @ ssimple.pl

before=

title= Nobel !

text==

Premier prix nobel de
littérature.

==

good=Prudhomme

Faire mieux ?

```
# suite
evaluator==
if good == lasaisie.value:
    grade = (100, 'Good 🌟🌟🌟')
else:
    grade = (0, 'Bad 🙄🙄🙄')
==
```

Je souhaite avoir le même exercice:

- * Insensible à la case.
- * Pouvoir fournir plusieurs réponses valables.
- * Accepter des erreurs d'orthographe

```
extends= @ ssimple.pl
# Modifications l'évaluateur
evaluator==
import ortho

for good in goods:
    if good == lasaisie.value.lower():
        grade = (100, ' Good 🌟🌟🌟 ')
    elif ortho(good, lasaisie.value)
        grade = (-1, 'corrigez l'orthographe.
SVP. ')
    else:
        grade = (0, 'Bad answer 🙄🙄🙄')
==
```

Mieux

J'en ai marre d'écrire mes exercices à la main !

J'ai un fichier CSV (excel, openoffice, une base de donnée, un json, etc) qui contient les informations que doivent connaître mes élèves.

Un exercice “csv”.

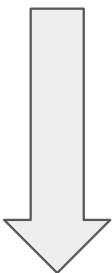
```
extends = /template/word_definition.pl
title = Git commands
@git.csv [definitions.csv]
text_word==
Que fait la commande *git {{word}}* ?
==
text_definition==
Quelle commande git *{{definition}}* ?
==
level = 1
```

fichier git.csv

```
niveau;nom;definition;fauxnom;faussedefinition
1;clone;Permet de copier en local un repository existant;copy;Permet de faire un double d'un pull du repository
1;status;Permet d'afficher l'état du repository courant;state;Permet de détecter l'état d'un repository
1;pull;Permet de récupérer les modifications distantes d'un repository;changes;Affiche les modification d'un repository
```

Oui mais je ne sais pas programmer !

```
inputbox =: Input
inputbox.type = number
inputbox.placeholder = Nombre de bonnes réponses (1)
inputbox.maxLength = 2
inputbox.appearance = outline
```



```
inputbox2 =: Input
inputbox2.type = number
inputbox2.placeholder = Nombre de mauvaises réponses (3)
inputbox2.maxLength = 2
inputbox2.appearance = outline
```

...

Exercice préparé : CheckBox

Nombre de bonnes réponses (1)

0

Nombre de mauvaise réponses (3)

0

Les bonnes réponses

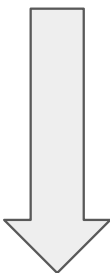
les mauvaises réponses

Les composants sont utilisables aussi pour les enseignants pour produire des exercices.

Un langage qui permet à tous de créer des exercices préparés:

Exercice préparé : CheckBox

```
# numeric:2:"Nombre de bonne réponses":1
nbg= 1
# numeric:2:"Nombre de bonne réponses":3
nbg= 3
# textmulti:2:"Les bonnes réponses"
```



Nombre de bonnes réponses (1)	
0	
Nombre de mauvaise réponses (3)	
0	
Les bonnes réponses	
les mauvaises réponses	

Les composants sont utilisables aussi pour les enseignants pour produire des exercices.

Un exercice avec un média

<https://pl.u-pem.fr/filebrowser/demo/31126/>

Différentes façons de produire cet exercice en utilisant une librairie de génération de parole, utiliser une archive contenant les différents média, création par drag and drop.

<https://pl.u-pem.fr/activity/play/741/>

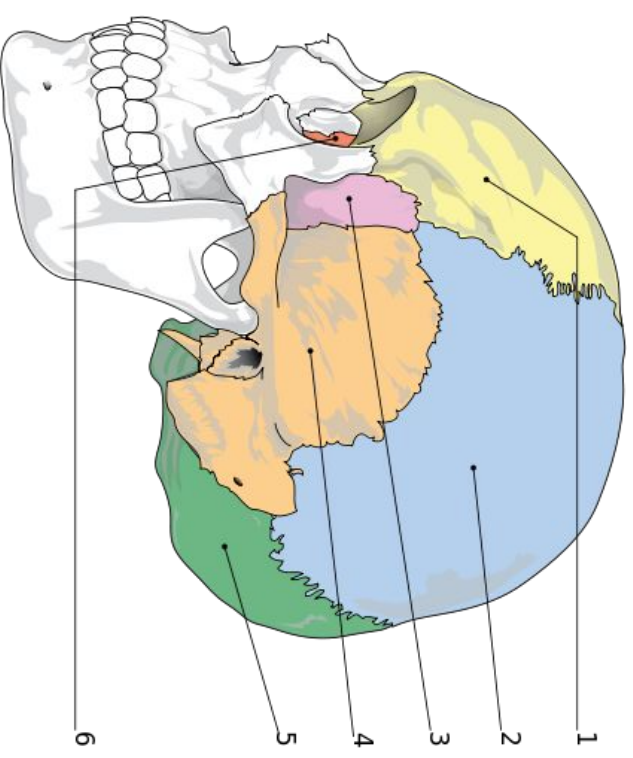
<https://pl.u-pem.fr/filebrowser/demo/31290/>

<https://pl.u-pem.fr/filebrowser/demo/31291/>

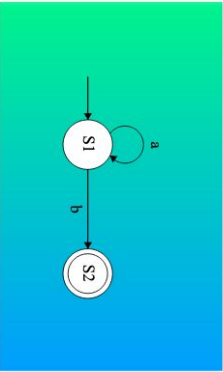
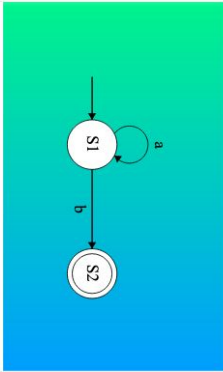
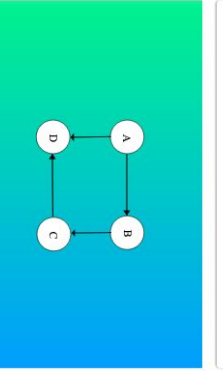
<https://pl.u-pem.fr/filebrowser/demo/31292/>

Biologie et Drag & Drop

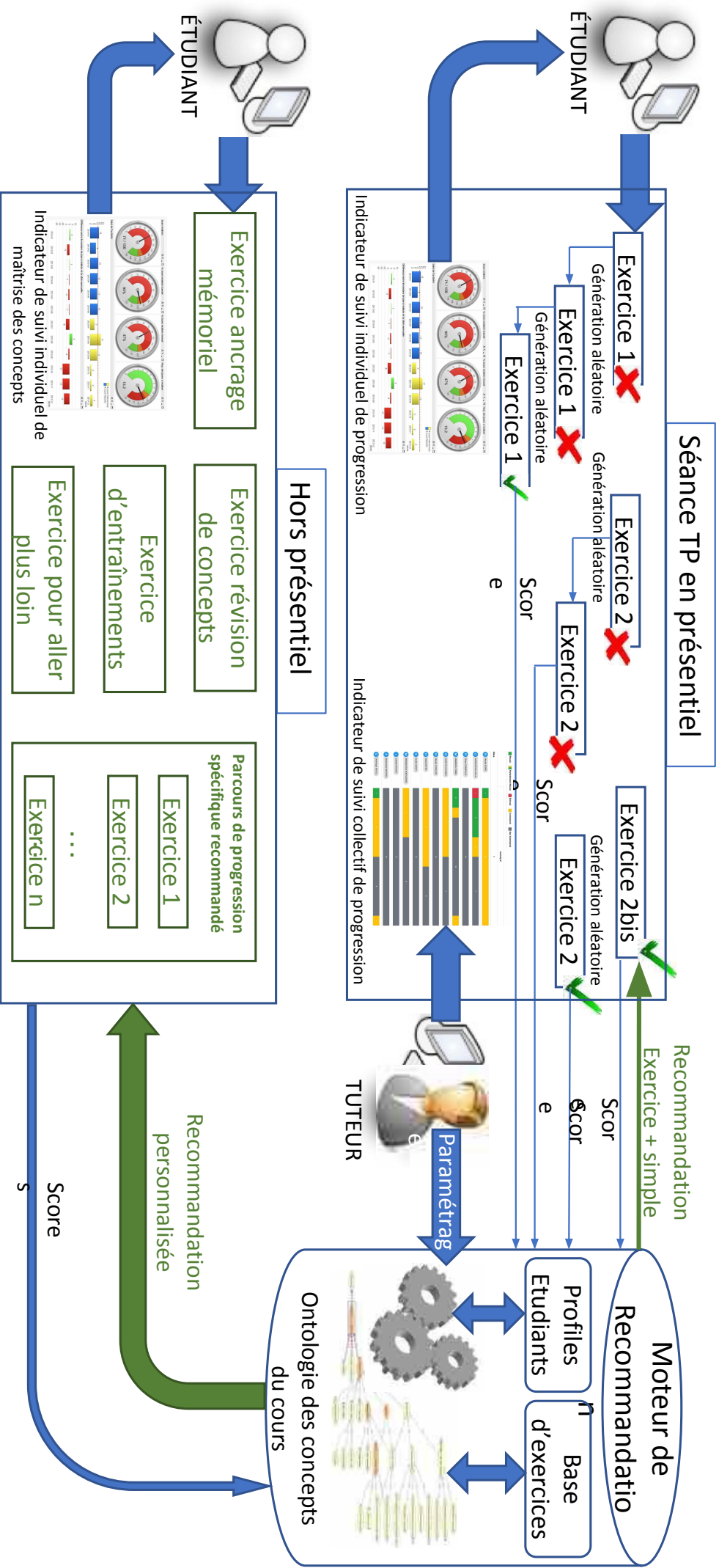
<https://pl.u-pem.fr/filebrowser/demo/31135/>



les components

Introduction Simple Usage Advanced Usage CSS API Automation Editor Automation Viewer Checkbox Group Code Editor Countdown Drag & Drop Graph Viewer Hint Input JSX Graph Match List Math Input Math Matrix Radio Group Sort List	Automation Editor A visual automation editor with drag & drop capabilities. 	Drag & Drop Drag & Drop lets users to drag and drop items between multiple sources and targets 	Math Input Math inputs provides a way for users to enter mathematical expressions. 
Automation Viewer Print a non editable automation. 	Graph Viewer Graph drawer prints a non editable graph. 	Math Matrix Matrix matrix provides a way for users to enter a matrix. 	
Checkbox Group <i>Placeholder icon has been found, but this icon differs from round</i> 	Hint 	Radio Group 	

Exemple de scénario d'organisation du module AP1 avec PL



Activités Interactive : l'Atelier

Une activité de TD avec plusieurs élèves.

Des exercices **ouverts** et fermés.

La plateforme choisi l'exercice que doit réaliser chaque élève:

- un exercice ouvert si il en reste à faire pour l'élève,
- une évaluation si il y en a à faire,
- un exercice fermé pour attendre les autres élèves.

Activités Interactive : l'amphi

Une activité d'amphi interactif.

L'enseignant passe des slides sur la plateforme (possibilité de d'utilisation hybride).

A tout moment l'enseignant peut décider de proposer un exercice de la plateforme. Les étudiants peuvent alors répondre.

L'enseignant voit les statistiques de réussite en temps réel.

Quand l'enseignant le décide l'exercice se termine pour tous.

Démonstration

ACTEURS

Les acteurs de la plateforme : **Élèves**

- Participant à un cours / une formation
- Utilisateurs principaux. Sans eux pas de logiciel.
- Objectifs:
apprendre,
valider des objectifs de formation (autonomie)
faire certifier des acquis d'apprentissage de la formation
- Besoin:
 - a. accès aux ressources (cours, vidéo, etc) fournis par les LMS.
 - b. Possibilité de s'entraîner (seul ou en groupe).
 - c. Possibilité de connaître les éléments nécessitant plus de travail (aide au parcours)
 - d. Être aidés pour la motivation, l'engagement, l'efficacité, l'organisation.

Scénarios d'usage pour les : **Élèves**

1. Participation à un cours

- a. Objectifs
- b. Suivi // Calendrier // notifications
- c. Devoirs // retour // rendus

2. Travaux de groupes

- a. atelier (corrections par les pairs)
- b. Travaux dirigés // Travaux pratiques
- c. Projets (petit groupes)

3. Activité autonome (choix des concepts et AAV)

- a. Découverte
- b. applications // feuille d'exercice
- c. anki // mémorisation
- d. révision // enchaînement // répétitions

Scénarios pour les : **Élèves**

Être aidé pour la motivation, l'engagement, l'efficacité, l'organisation.

-> points, badges, récompenses, (ludification), flow,

-> Niveaux de difficulté, Compétitions,

-> Détection de **mauvaises** stratégies d'apprentissage :
répétitions excessives, adaptation au test pas au concept

-> Organisation dans le temps (heure, jour, semaine, année) des objectifs de formation

Un concept : **Ressources**

Les **ressources** sont des documents, programmes, fichiers, composants logiciels accessibles sur le serveur central du projet.

Les **ressources** sont **ouvertes** ce qui a pour sens :

- accessible en Creative Common Shared Alike
- extensibles (modifiable avec conservation de version)



Les ressources subissent une **curation** générale :

- évaluées, reviewed (en production, à vérifier, nouveau, défectueux,...),
- taggés sur différentes ontologies (graphes épistémiques)
- Classées (par niveau scolaires, disciplines, langue, etc)

Un concept : **Ressources**

- exercices
- documents (video, text, sons, html, markdown etc
tout ce qui peut être proposé par un navigateur)
- activités (applications django, applications pl, applications typescript)
- cours
- Snipets (portions d'activités)

Un concept : **assets**

Quand une ressource est placée dans un cours elle devient accessible au élèves et devient un asset du cours.

Quand la ressource est transformé en asset le choix de la version de la ressource est fixé.

Quand la ressource est un cours, l'enseignant conserve le choix de la version qu'il souhaite intégrer comme asset dans son cours.

Assets

L'asset gagne un **tableau de bord** propre à la gestion du cours:

- Accessibilité (datetime ouverture/fermeture/validation)
- Visibilité (oui/non)
- Statistiques ET conservation de tous les usages/accès
 - (nombre d'accès, temps passé, moyennes sur le groupes etc)
 - Tableau de bord graphique
- Note et modalité de calcul de la note (formules par défaut)

Concept : **component**

Les **components** sont des interfaces utilisateurs que l'on peut intégrer dans un exercice ou une activité.

<https://pl.u-pem.fr/components/intro>

PLaToN, au service des Enseignants

- les profs construisent leurs **triforces** (objectif / activité / évaluation)
- définir les **objectifs**, construire une **stratégie** de cours
- les profs **choisissent** / **modifient** / **fabriquent** les exercices (surtout pour les étudiants les moins experts dans le domaine ou en apprentissage)
- qui déterminent les **feedbacks** (déclenchement et contenu).
- qui **organisent les interactions** entre élèves, tuteurs, profs (websockets)
- **l'évaluation** est automatique et paramétrée par l'enseignant ou faite par l'enseignant et ou les pairs ou encore en auto-évaluation.

The screenshot displays the PLaToN web application interface. The top navigation bar includes links for 'Cours', 'Faq', 'Documentation', and 'Administration'. The main area is divided into two panels. The left panel, titled 'EXPLORER', shows a file tree with folders for 'builder', 'Chemistry', 'ComputerScience', 'AnthonyAlbare', 'Automates-Charlevoix', and 'automaton'. The right panel shows a code editor with a file named 'factor.pl'. The code defines a finite automaton for factoring, including a 'maxattempt' parameter and a 'generate' function. Below the code editor, a 'Preview factor.pl x' window shows a state transition diagram with four states: S0 (initial), S1, S2, and S3 (final). The diagram shows transitions labeled 'e' between the states. A text box above the diagram explains the task: 'Pour cette question, on considère l'alphabet $\Sigma = \{a, b, c\}$. Donnez un automate reconnaissant le langage $L1 = \{\text{mots ayant } abc \text{ en facteur}\}$. Il vous reste 5 tentativ(e) avant de voir une solution !'. At the bottom right, there are buttons for 'Initial', 'Non final', 'Renommer', and 'Supprimer'.

Les utilisateurs de la plateforme : Enseignant

Responsable d'un cours. Utilisateurs principaux. Sans eux pas de logiciel.

- **Objectifs:**
 - a. Définir dans le système les éléments suivants, les Pré-requis, les objectifs de formation (AAV), les assets (exercices, supports, activités), les modalités d'évaluation.
 - b. Construire le cours
 - c. Faire cours.
 - d. Suivre les élèves.
 - e. Organiser les évaluations.
- **Besoins**
 - a. Trouvez des ressources existantes.
 - b. Organiser le cours et l'équipe pédagogique
 - c. Proposer un alignement solide entre contenus et évaluation

Les utilisateurs de la plateforme : Enseignant

- Accéder à des ressources existantes (bibliothèque de ressources, paquets de ressources)
- Possibilité d'adapter//modifier//créer//évaluer des ressources
- Possibilité d'organiser les ressources dans le cadre d'un cours
- Être mis en relations avec des collègues avec les mêmes besoins.

Les utilisateurs de la plateforme : Créateur

Fournisseurs de Ressources. Sans eux pas de logiciel.

- Objectifs:

Créer // modifier // mettre à jour // détruire des ressources

Curation des ressources (édition des métadonnées)

Création de nouveau types de ressources (template, composants, etc)

Partager

L'importance du partage !

=> suivre ce que font les informaticiens pour les logiciels un outil de gestion de version.

=> plusieurs niveaux :

- une couche “programmeurs” (components, libs) -> git
- une couche “modelleurs” (types d'exercices) -> git
- une couche “créateurs” (exercices) ->

Les Cercles

