Retour d'Expérience sur une UE Projet en Licence Informatique

Nicolas Bonichon, Aurélien Esnard

SIF – Journée Enseignement 2023

Retour d'Expérience sur une UE Projet en Licence Informatique

nicolas.bonichon@u-bordeaux.fr
aurelien.esnard@u-bordeaux.fr

Exposé à la SIF le 10 mai 2023.

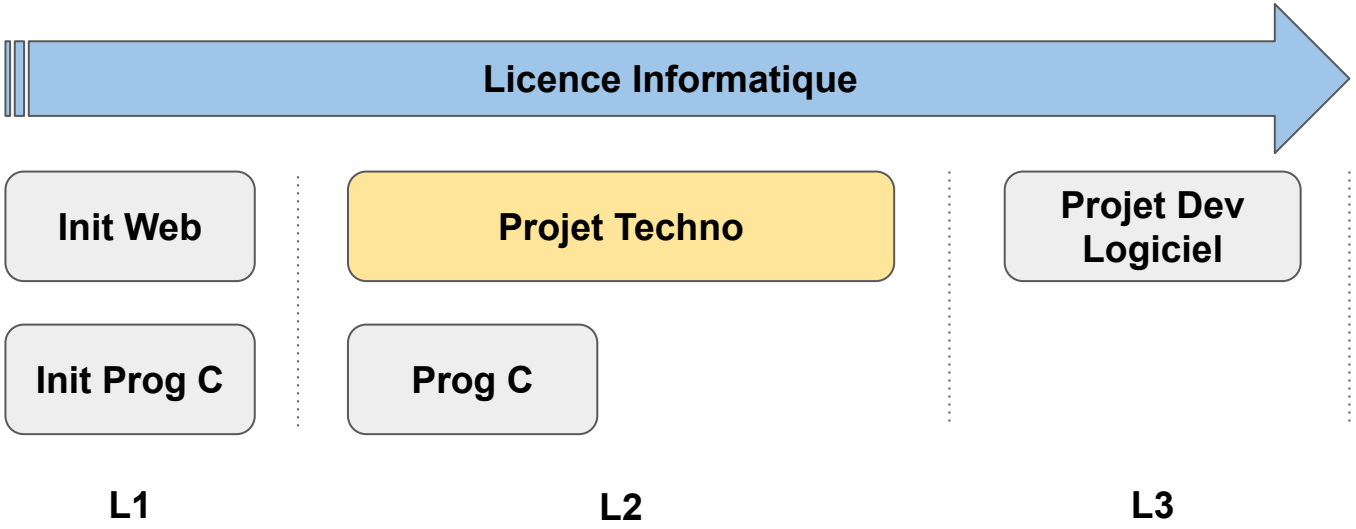
université de BORDEAUX

Contexte

Un retour d'expérience sur une UE Projet en Licence Informatique à l'Université de Bordeaux.

Fiche d'identité de l'UE Projet Techno

- UE à 6 ECTS se déroulant sur toute l'année
- 6 CM + 1h20 de TD Machine / semaine (40 heures de présentiel)
- ~250 étudiants inscrits, essentiellement en L2 Informatique
- 12 groupes de TD et 9 chargés de TD
- ~75 équipes de 3 étudiants



Objectifs Pédagogiques

Méthodes & Bonnes pratiques

“Seul, on va plus vite. Ensemble, on va plus loin.” (Proverbe Africain)

- renforcer le niveau en prog & algo
- travailler en équipe
- tests unitaires, couverture
- code propre, documenté
- déboggage, code maintenable
- découpage modulaire & interop.
- respecter un cahier des charges
- …

Socle Technique

“A poor workman always blames his tools.” (Proverbe Britannique)

- VSCode
- GCC / GDB / Valgrind
- Git & GitLab
- CMake / Make
- SDL, Android
- HTML / JS / WASM
- …

Pourquoi un Projet ?

Problématique

- Avant 2012, UE au S4 avec des modalités classiques : cours + TD + mini-projets
- Adhésion modérée des étudiants aux mini-projets
- Une incompréhension légitime : *“Quand je code ma petite bricole, je n'ai pas besoin des outils et méthodes présentés en cours !”*

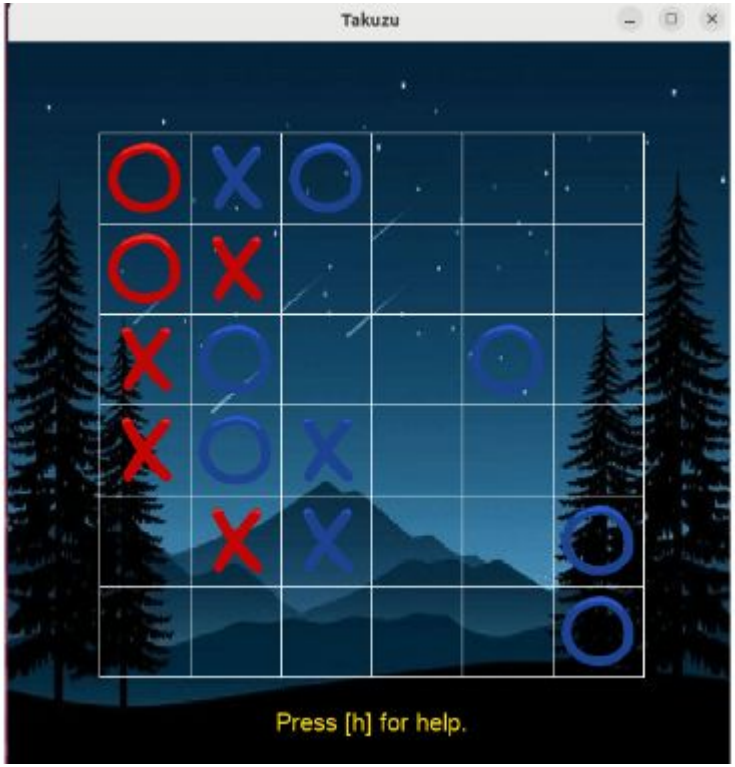
⇒ Accepter de perdre un peu de temps pour en gagner beaucoup !

Evolution vers une UE Projet

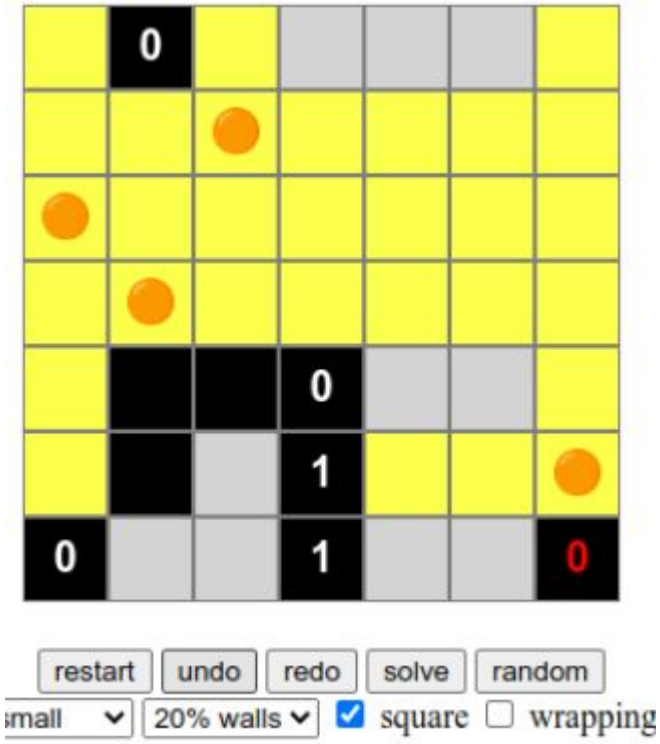
- Un projet long avec un scénario rendant l’application des méthodes et l’utilisation des outils utile voire nécessaire…
- Passage progressif sous Moodle / VPL & GitLab, … → gestion simplifiée des projets

Programmer un Jeu (Puzzle Logique)

Un prétexte pour apprendre les méthodes et les outils de dev. logiciel…



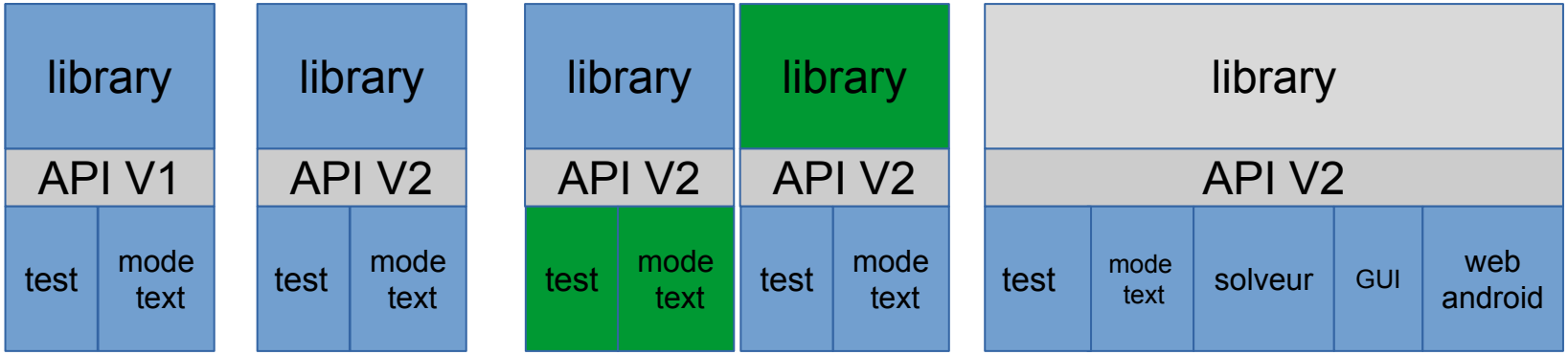
Takuzu (version SDL) – 2022-2023.



Lightup (version web) – 2021-2022.

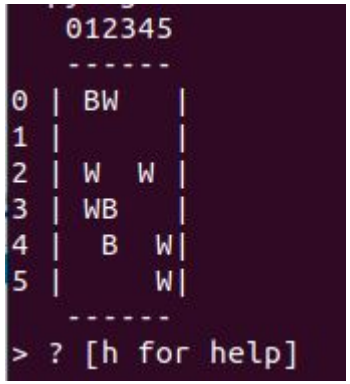
Scénario

fourni par l'enseignant | produit par les étudiants | autres étudiants



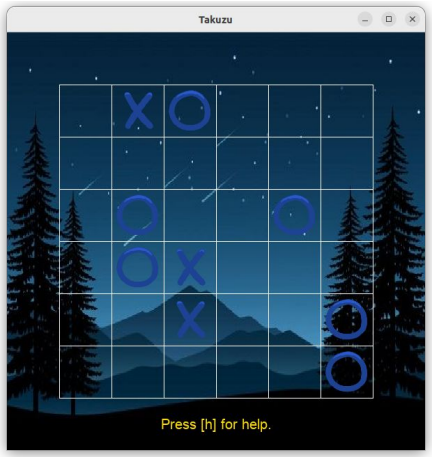
Caprice du client : V1 → V2

Relecture croisée

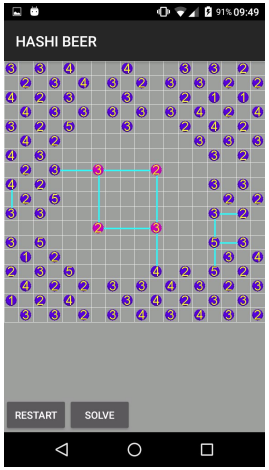


texte

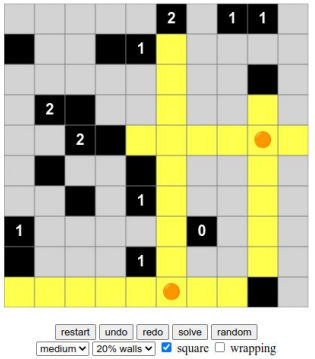
solveur



graphique



android



web

Les Principaux Jalons

Semestre 3

- TD00 - Stage Env.
- TD01 - Make [I] [A]
- TD02 - Interface Texte [E] [A]
- TD03 - Git & GitLab [I] [A]
- TD04 - CMake [E] [A]
- TD05 - Tests [E] [A]
- TD06 - Implém. du Jeu [E] [A]
- TD07 - Extension du Jeu [E] [A]
- TD08 - Livraison du Projet [E] [A]

[I] : travail & rendu Individuel
[E] : travail & rendu en Équipe

Semestre 4

- TD08 - Evaluation Croisée [I] [M]
- TD09 - Git & GitLab Avancé [I] [A]
- TD10 - Fichiers & Random [E] [A]
- TD11 - Solveur [E] [A]
- TD12 - Interface Graphique [E] [M]
- TD13 - Rapport Latex [I] [M]
- TD14 - Portage Android [E] [M]
- TD15 - Interface Web [E] [M]
- TP Noté [I] [A]

[A] : correction Automatique
[M] : correction Manuelle (grille)

⇒ Evaluation en Contrôle Continu Intégral avec plus de 30 notes !

Organisation d'un TD avec Moodle

TD05 - Les Tests

Durée : 2 semaines
Rendu Initial : coeff 1, individuel, 1 seule passe
Rendu Final : coeff 5, en équipe, 2 passes

- Introduction aux tests unitaires (vidéo de N. Bonichon)
- Sujet du TD05
- 05 - Rendu Test - Initial (TM)
- 05 - Rendu Test - Final (TEAMS S3)
- 05 - Rendu Test - Final (deuxième passe) (TEAMS S3)
- Sondage TD05

Rendu Projet (VPL)

TD5 : Tests

Ce TD va vous apprendre à programmer une batterie de tests pour tester le bon fonctionnement de la bibliothèque *game*.

Exercice 1 : activité préliminaire

Considérez l'exemple d'une structure de données "file" (ou *queue* en anglais), tel que les éléments les premiers entrés sont aussi les premiers sortis (First In, First Out) : https://github.com/orel33/queue.

- Faites un *clone* de ce projet Git.

Ce petit projet se compose de plusieurs fichiers, dont voici une brève description :

- le module *queue* (`queue.c` + `queue.h`)
- un exemple d'utilisation de la queue (`sample.c`)
- un fichier de tests du module *queue* (`test_queue.c`)
- le fichier `CMakeLists.txt` pour compiler ce projet

La compilation du projet et l'exécution des tests se fait de la manière suivante :

```
$ mkdir build ; cd build
$ cmake .. ; make          # compilation
$ make test                # lancement des tests
```

- Analysez en particulier le code des fichiers `queue.h`, `test_queue.c`.
- Dans le fichier `CMakeLists.txt`, comprenez le rôle des commandes `add_test()` ainsi que de la commande `enable_testing()` en préambule.

Sujets sur GitLab Pages

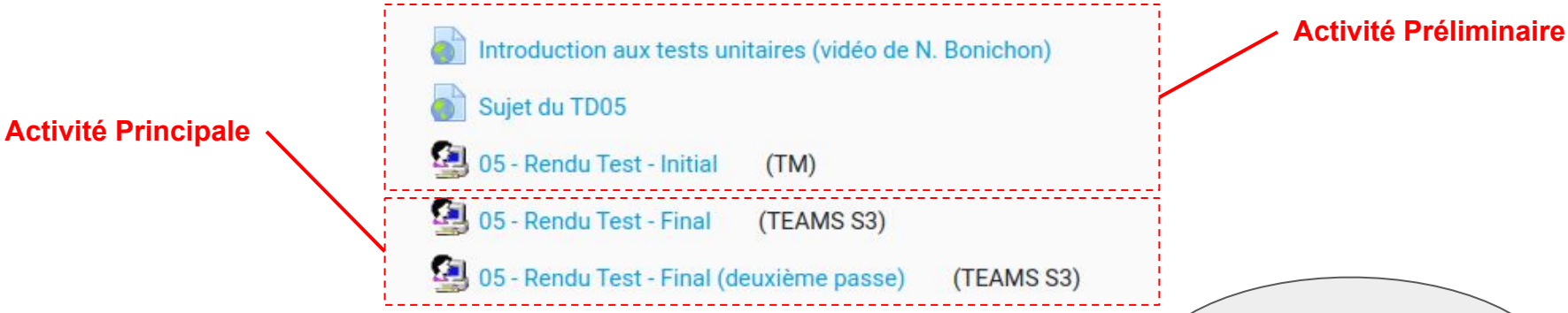
Déroulement typique de la Séance (1h20)

- debrief sur le dernier rendu
- travail / debrief sur l'activité preliminaire
- coordination de l'équipe, répartition du travail pour le prochain rendu

Stratégie d'Évaluation (Automatique)

TD découpé en deux activités

1. **Activité préliminaire** avec un rendu initial (plutôt individuel, à préparer…)
2. **Activité principale** sur le projet avec un rendu final (en équipe)

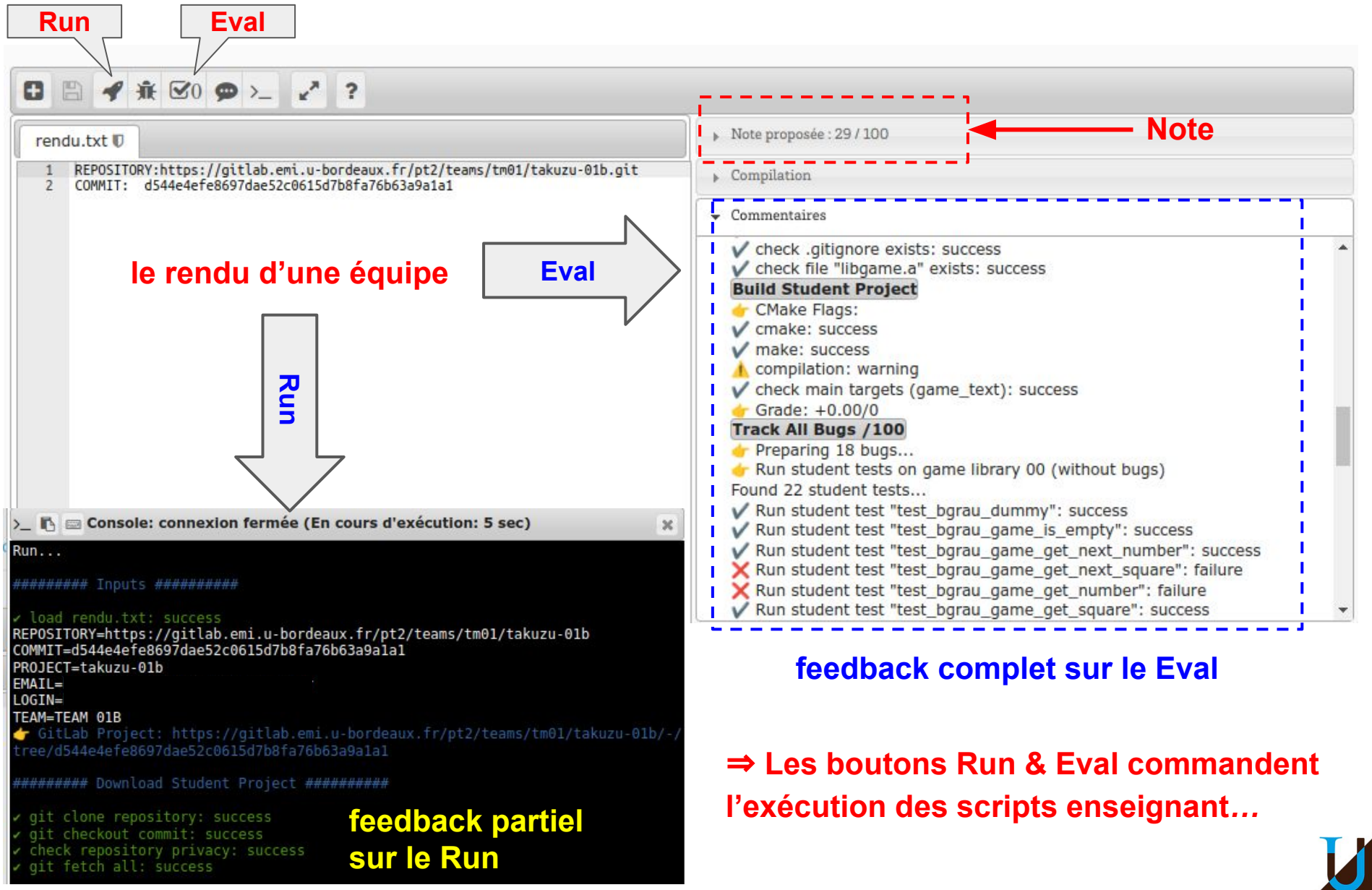


Principe de l'évaluation (automatique) en deux passes

On évite le : *"Je ne comprends pas, j'ai eu 0 alors que j'avais bien travaillé !"*

- Une première passe à l'aveugle (ou presque)
  - avant la deadline : **feedback partiel**, à volonté
  - à la deadline : feedback complet avec une note (*one shot*)
- Une deuxième passe pour se rattrapper ou se perfectionner
  - feedback complet, à volonté…
- Note finale = max (première passe, moyenne des 2 passes)

Mise en Oeuvre dans Moodle avec VPL

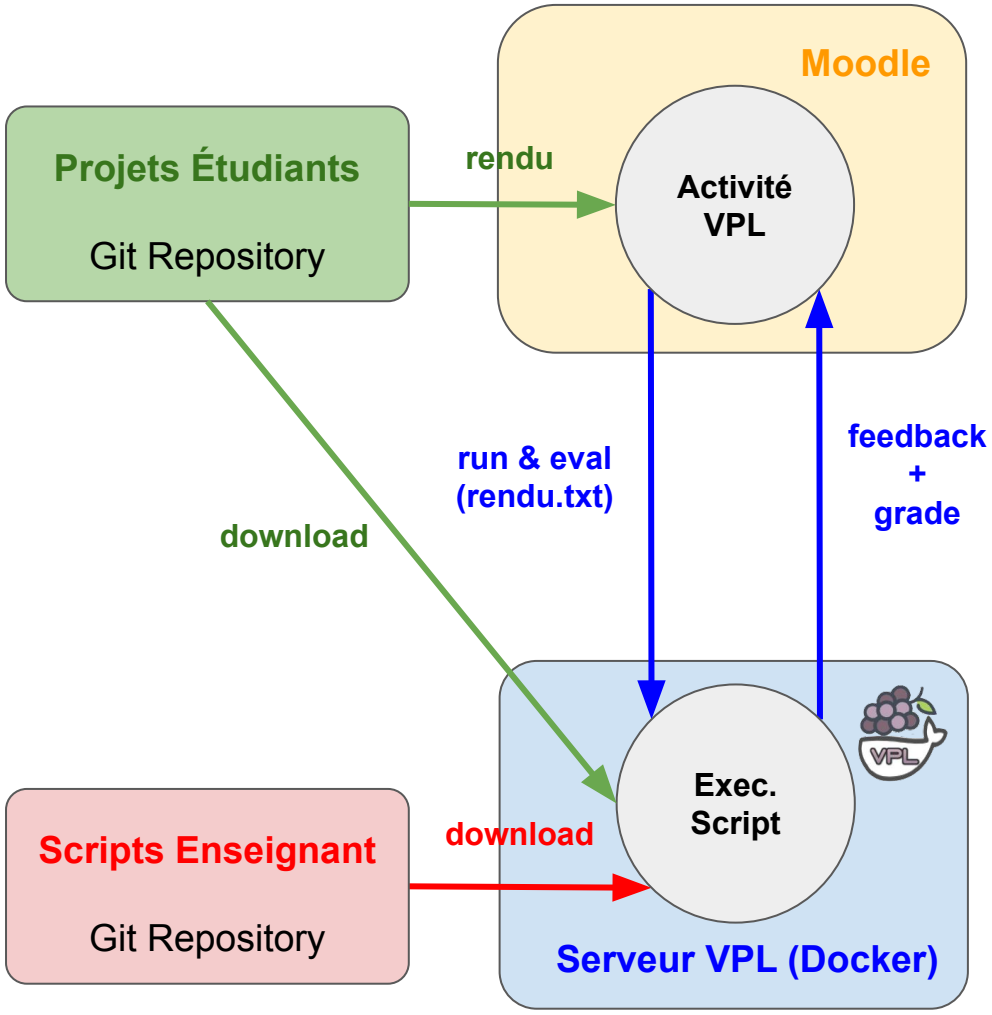


⇒ Les boutons Run & Eval commandent l'exécution des scripts enseignant…

Backend Technique

Correction Automatique avec Moodle / VPL & Git

- Rendu d'un *commit* Git pour ne pas coder dans VPL
- Contrôler l'environnement d'exécution avec Docker
- Possibilité de valider les scripts avec la solution "enseignant"



script shell

```
function eval_v2()
{
    ### download
    inputs || EXIT_GRADE 0
    download_student 0 0 0 0 0 0 || return 1

    ### check
    git_fame
    gitlab_issues "v2"
    git_extra_check -5 -5 -5 -5 -5 -5 -5 -5
    check_code_format 0 -5

    ### build
    build_student 0 0 0 0 -5 || return 1
    check_student_targets "$STUDENTDIR/build"  0 0 "libgame.a game_text"
    check_libgame_symbols_v2 0 0

    ### test
    check_libgame_test "$STUDENTDIR/build/libgame.a" "$RUNDIR/test" "testv1" "V1" 20 0 "-DTESTV1=ON"
    check_libgame_test "$STUDENTDIR/build/libgame.a" "$RUNDIR/test" "testv2" "V2" 50 0 "-DTESTV2=ON"

    ### run all tests
    # local BUGV1="01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18"
    local BUGV2="20 21 22 23 24 25 26 27 28 29 30 31 32 33"
    track_bugs "$RUNDIR/bugs" 30 0 "00" "$BUGV1 $BUGV2"

    return 0
}
```

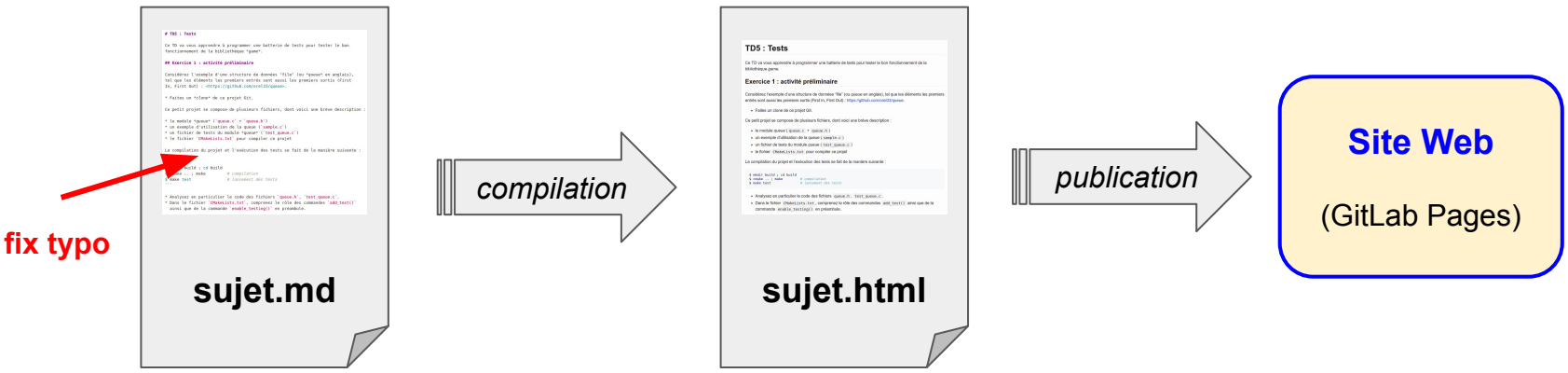
> 5000 lignes de scripts shell

https://github.com/orel33/vpltoolkit
https://github.com/GuillaumeBlin/vplbdx

Backend Technique

Écrire et publier son cours avec GitLab CI

- Gestion des ressources de cours avec Git & GitLab
- Travail collaboratif de l'équipe pédagogique (Web IDE)
- Amélioration continue des supports avec GitLab CI
  - compilation des supports *Markdown* avec *mkdocs* (sobriété)
  - publication automatique sur le web à chaque *commit push*



⇒ https://pt2.gitlabpages.inria.fr/support/site/

Externaliser les supports & ressources en dehors de Moodle

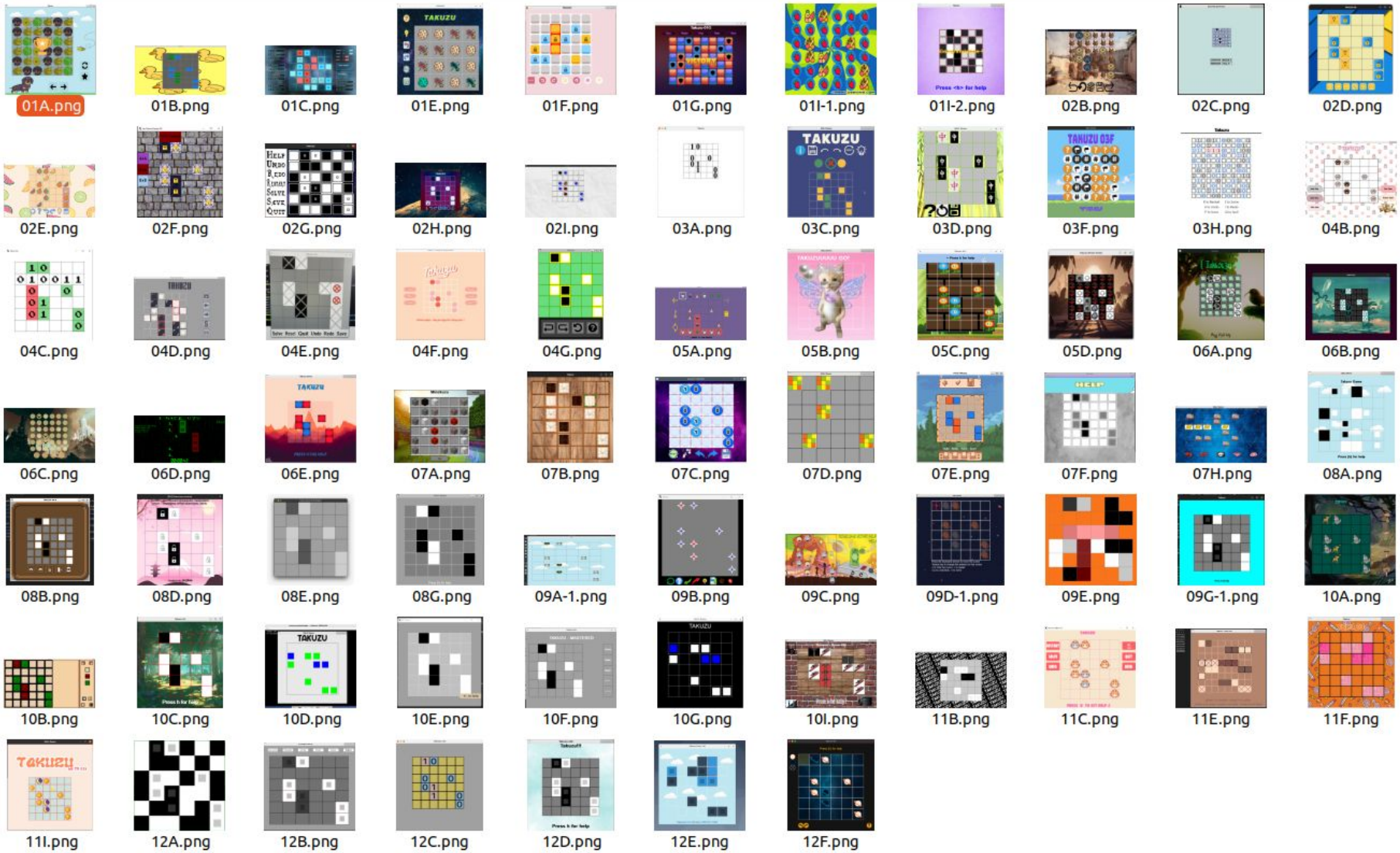
- Utiliser des liens externes pour faciliter la mise à jour automatique des supports et leur partage avec la communauté…

Conclusion

- Une approche projet très motivante pour les étudiants :-)
- Une UE qui demande beaucoup de travail
  - pour les étudiants, mais aussi pour les enseignants !
- Une UE qui apporte beaucoup de satisfaction
  - étudiants : fierté du travail accompli en fin d'année
  - enseignants : retours très positifs des étudiants (modalités plébiscité par les étudiants)
- Bénéfices de l'évaluation automatique
  - gestion d'une cohorte importante (~250 étudiants) à coût quasi-constant
  - maintenir un rythme de travail soutenu tout au long de l'année !
- Difficultés
  - scripts shell difficiles à maintenir (potentiellement chronophage)
  - retours des scripts pas toujours explicites (pour les étudiants et les enseignants)
  - difficile d'avoir des scripts qui acceptent tous les codes étudiants → amender à la main
- Bénéfices indirects / cachés
  - satisfaisant de coder une script d'évaluation plutôt que de corriger toujours les mêmes codes
  - changement de posture en TD : *"l'enseignant est le gentil qui aide l'étudiant à satisfaire le méchant script"*

Page Moodle : https://moodle1.u-bordeaux.fr/course/view.php?id=10858 (accès authentifié)
Ressources Web : https://pt2.gitlabpages.inria.fr/support/site/ (accès libre)

Questions ?



SDL Wall du jeu Takuzu, 2022-2023.

Annexes

Postionnement sur la Pédagogie de Projet

En quelques mots-clés…

- **une réalisation concrète ?** oui, production d'un petit logiciel avec plusieurs interfaces (texte, graphique, android, web)
- **intégrer de nouvelles connaissances & compétences ?** oui, notamment avec les activités préléminaires sur les technologies de base du développement logiciel…
- **choix libre du projet ?** non, un projet imposé…
- **créativité, résolution de problèmes ?** partiellement, car on est dans un cadre imposé, mais les choix d'implémentation sont libres…
- **collectif & coopératif ?** oui, un travail en équipe tout au long de l'année…
- **apprentissage de l'autonomie ?** progressivement, …
- **motivation ?** oui, forte motivation des étudiants pour un projet de type "jeu" renforcé par les feedback rapides dans VPL et les challenges !
- **bilan & restitution finale ?** partiellement, un rapport final et une relecture croisée du code (évaluations par les pairs)

Retour des Etudiants

Des retours très positifs. Enquête de satisfaction basée sur un sondage anonyme, avec plus de 50% de participation.