



ÇA FERAIT PRESQUE LE CAFÉ : UNE ÉVALUATION CONTINUE, À DISTANCE, FOURNISSANT DU FEEDBACK ET AUTOMATISÉE

Retour sur 4 ans d'évolution du cours
d'Introduction à la Programmation en Sciences
Informatiques à l'ULiège.

Simon Liénardy – Uliège
simon.lienardy@uliege.be



PLAN

- Contexte
- Genèse de CAFÉ
- Challenges ?
- CAFÉ : Correction Automatique et Feedback des Étudiant
 - Fonctionnement
 - Feedback et feedforward
 - Assessment for Learning
- Retours d'expérience
- Bilan

CONTEXTE

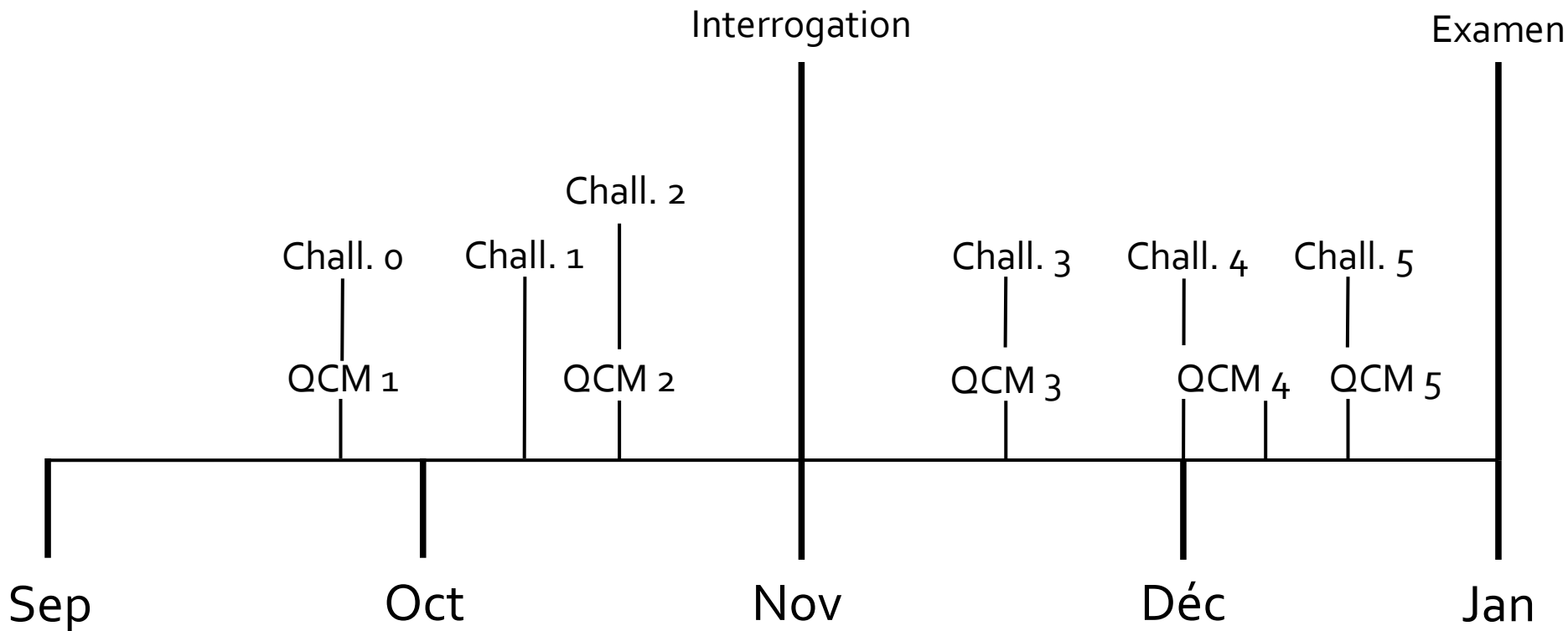
- Cours d'introduction à la programmation
 - Titulaire : Pr Benoit Donnet
- Bloc 1, Sciences Informatiques, 1^{er} quadrimestre
 - ~ 100 étudiants
- Programmation ?
 - Rédiger un **programme** dans un **langage de programmation** qui sera ensuite **traduit** en instructions exécutables par un ordinateur.
 - Emphase sur la conception **rigoureuse** de programmes corrects et efficaces.
- Constat il y a 4 ans : Besoin de **travail régulier**
 - Rien à voir avec le SARS-CoV-2

AU COMMENCEMENT...

GENÈSE DE CAFÉ

- Dispositif longitudinal d'évaluation formative en ligne
 - Exercitation à intervalle régulier, **alignée avec le *pacing* du cours**
 - Correction systématique des tentatives et **feedback individuel**
 - RH limitées -> système automatique
 - **Intégré** au dispositif global d'évaluation certificative dans une logique d'évaluation continue

DISPOSITIF GLOBAL D'ÉVALUATION



Pondération :

Interro

Examen

QCM

Chall

Individuel, séance de consultation copie

Collectif, à chaud, en début de TP

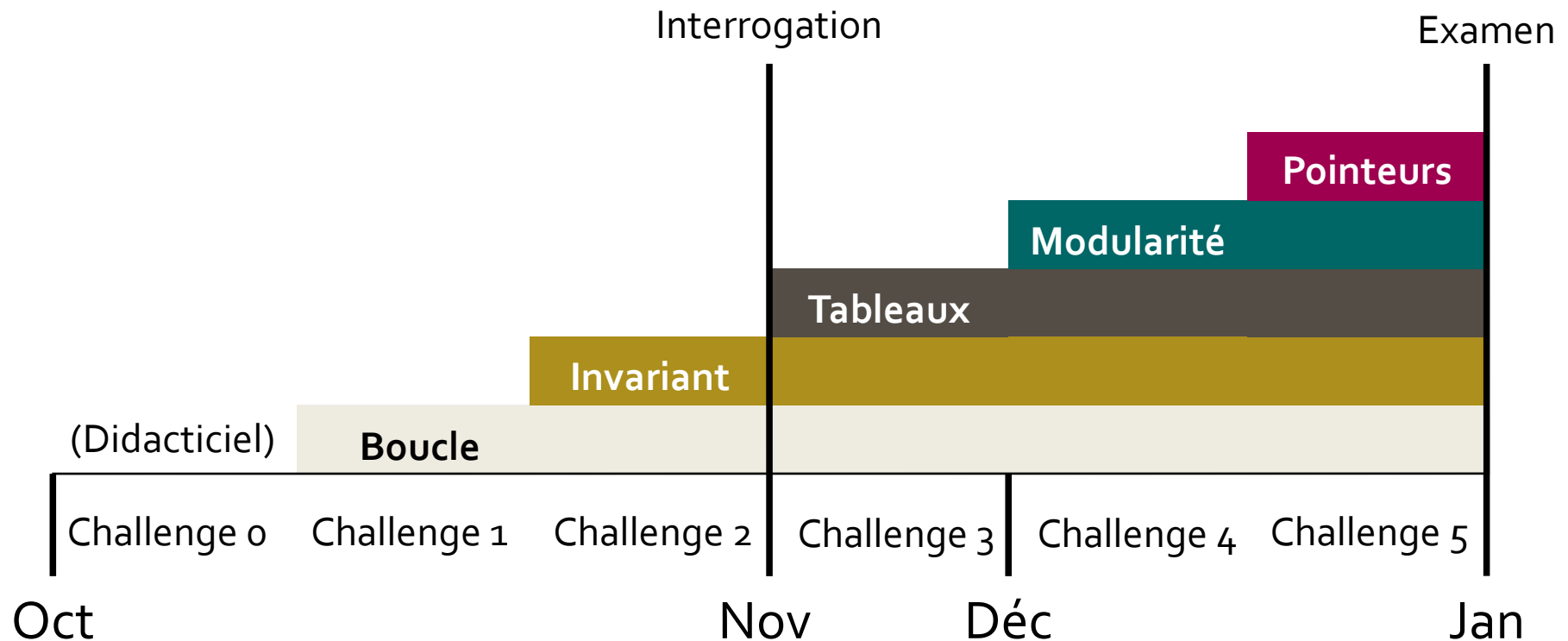
Feedback ?

Collectif en séance de remédiation

Individuel, instantané et automatique

CHALLENGES ?

Terme préféré au mot « Devoir », à la lisière de l'évaluation et de l'enseignement

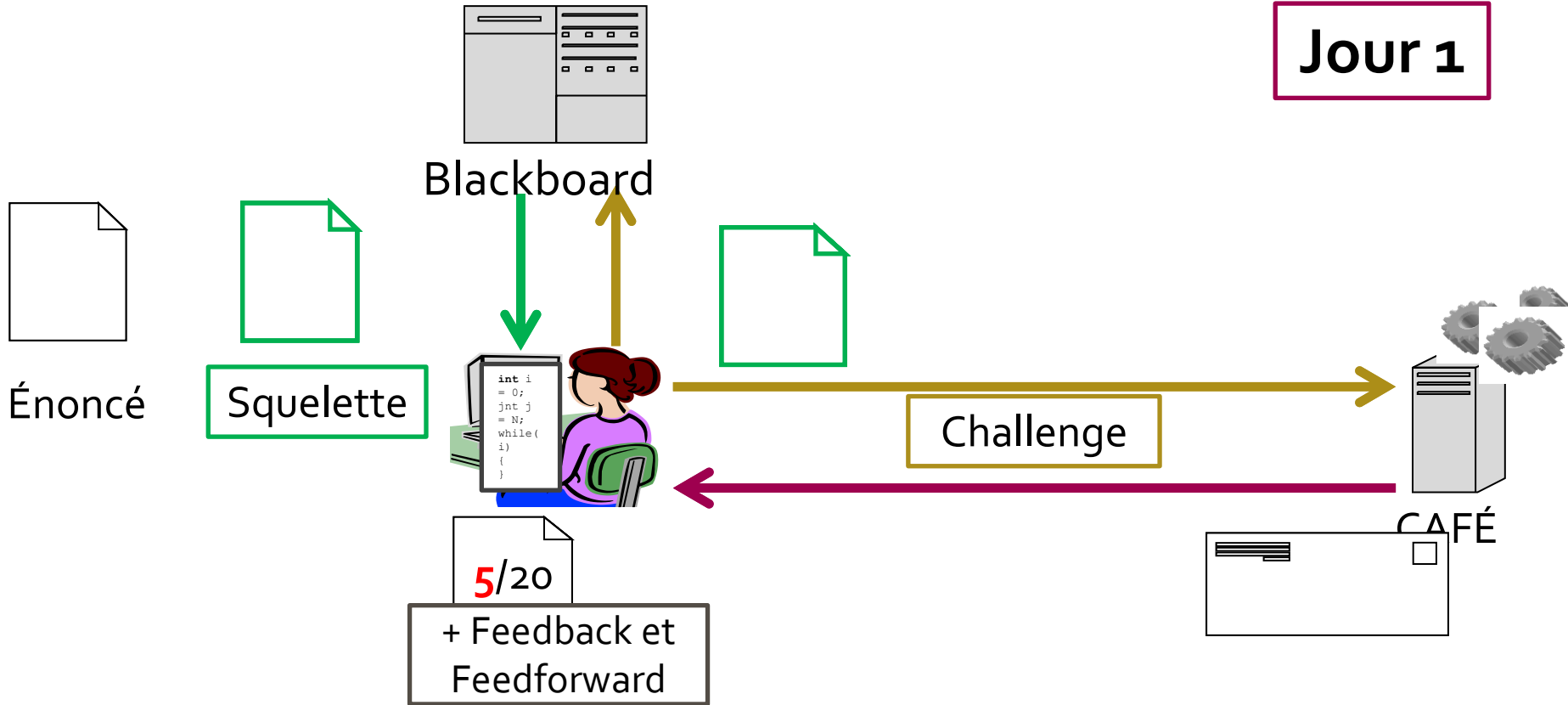


- Difficulté croissante et cumulative

CAFÉ, WHAT ELSE?

CORRECTION AUTOMATIQUE ET FEEDBACK DES ÉTUDIANTS

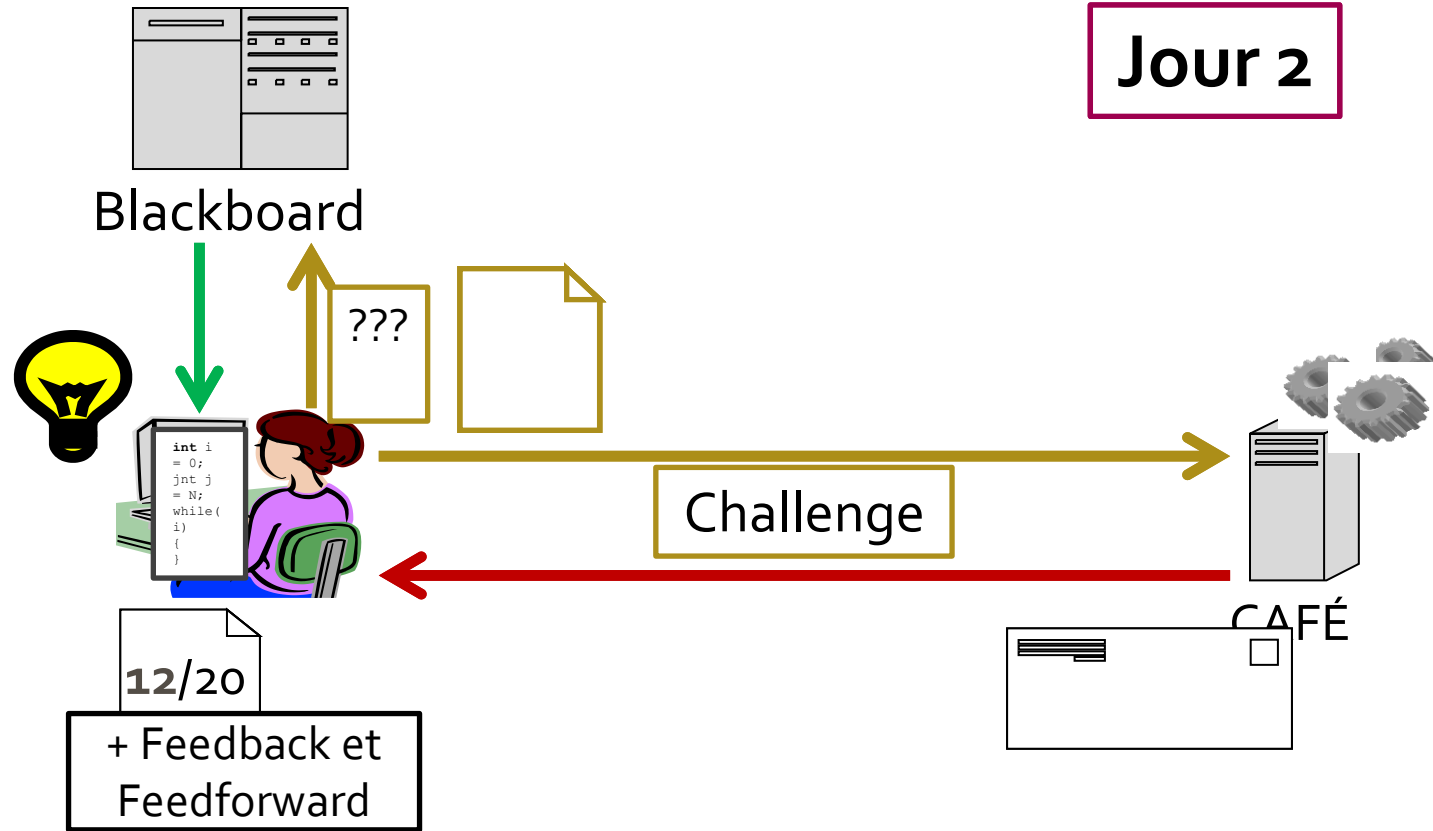
Jour 1



CAFÉ, WHAT ELSE?

CORRECTION AUTOMATIQUE ET FEEDBACK DES ÉTUDIANTS

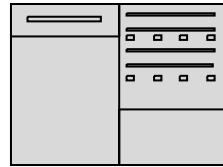
Jour 2



CAFÉ, WHAT ELSE?

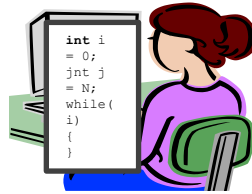
CORRECTION AUTOMATIQUE ET FEEDBACK DES ÉTUDIANTS

Jour 3



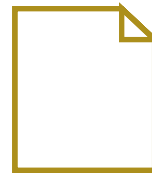
Blackboard

→ Closing the feedback loop [Boud, 2000]

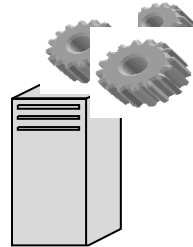


19/20

+ Feedback et Feedforward



Challenge



CAFÉ



FEEDBACK ET FEEDFORWARD

Bonjour !

Voici plus de détails sur votre soumission :

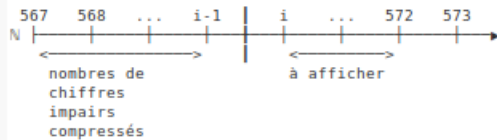
Tentative de compilation de votre code

La compilation a réussi.

Attention : cela ne signifie pas que votre code est correct ou qu'il respecte quelque contrainte que ce soit (voir section 1.2 de l'énoncé).

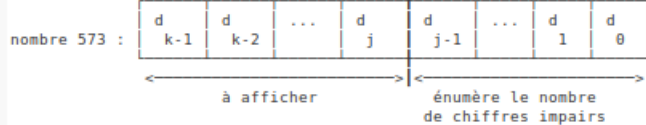
Invariant du SP1 : Énumération dans $[a, b]$ et affichage de la quantité de chiffres impairs pour chaque valeur de l'intervalle

Voici comment le système comprendra votre invariant lors de la correction :



Invariant du SP2 : Calcul du nombre de chiffres impairs pour un nombre donné

Voici comment le système comprendra votre invariant lors de la correction :



Recommandations

N'hésitez pas à soumettre une nouvelle fois si vous en avez besoin et l'opportunité.

Bon travail !

Titres

Information sur le fichier déposé

Invitation à soumettre à nouveau

FEEDBACK ET FEEDFORWARD

Bonjour !

Votre résultat est de 10.2/20

Note /20

Voici plus de détails sur ce résultat :

Conseil de régulation prioritaire

Pour corriger votre code, testez chez vous avec différentes valeurs, comparez au résultat attendu (décrit dans l'énoncé). Utilisez ensuite votre Invariant pour corriger votre code.

Pour corriger votre code, testez chez vous avec différentes valeurs, Comparez avec le résultat attendu (décrit dans l'énoncé). [...]

Respect de l'énoncé lors de l'écriture du code

Correct !

Correction de l'Invariant

Voici comment le système comprend votre invariant :

0 | i | N
A: A Additionner | Déjà Additionné

Information sur la performance de l'étudiant

Concernant le tableau A :

Boite 5 : Contenu incorrect ! Les indices commencent tous à ...
(Vadémécum de l'Invariant, règle 2)
Boite 6 : Variable détectée : i
Boite 4 : Contenu incorrect ! On s'attend à l'indice avant la première case
(Vadémécum de l'Invariant, règle 2)

[...]

Le reste des boites :

Boite 22 : Variable détectée : reste

Passons à l'analyse du code :

=> Le code et l'invariant sont compatibles en ce qui concerne l'initialisation de i
Pas d'inspection pour les tableaux B et C puisque les variables de parcours sont identiques.

=> Le code et l'invariant sont compatibles en ce qui concerne l'initialisation de reste

Autre conseil de régulation

Pour corriger votre/vos Invariant/s, relisez l'énoncé, faites un dessin (en utilisant par ex. l'outil en ligne GLI), comparez ce dessin au schéma muet de l'énoncé, complétez ensuite le squelette selon les recommandations de l'énoncé.

Vérification du résultat

Lorsque A et B sont tirés au sort :

Votre code ne fournit pas le résultat attendu,
avec N = 29, A = 10341994, B = 135516839 :

```
0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 1 1 0 0 1 1 1 1 0 0 1 1 1 0 1 0 1 0  
+ 0 1 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 0 1 0 0 1 0 1 0 1 0 0 1 1 1  
0 1 0 0 0 1 0 1 1 1 0 0 0 1 1 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1
```

Votre solution :

48 48

Exemple de comportement erroné du programme

ASSESSMENT FOR LEARNING


1. Authentic assessment
2. Balancing summative and formative assessment
3. Creating opportunities for practice and rehearsal
4. Designing formal feedback to improve learning
5. Designing opportunities for informal feedback
6. Developing students as self-assessors and effective lifelong learners

CAFÉ
participe à
cette
orientation

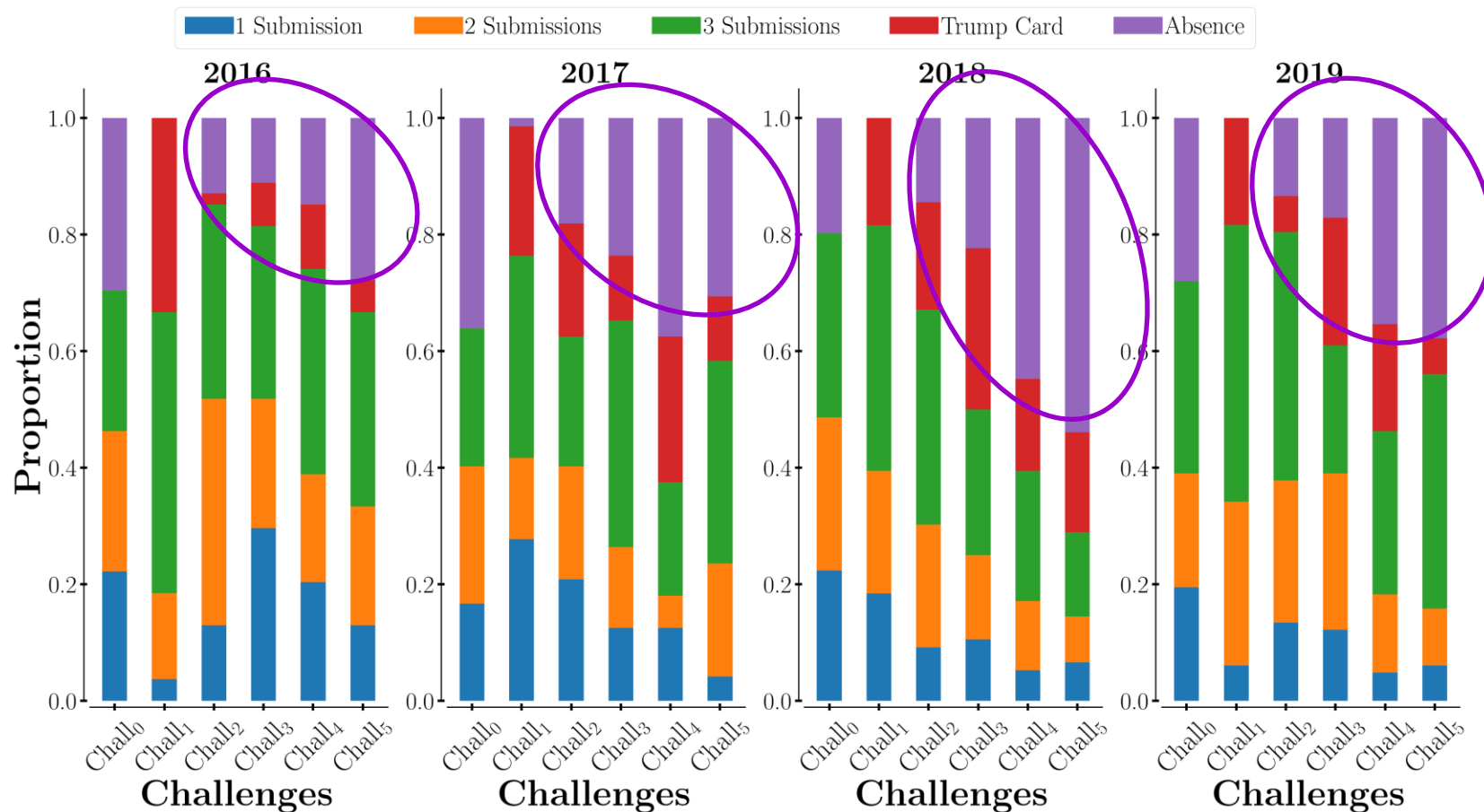
Kay Sambell, Liz McDowell, and Catherine Montgomery. *Assessment for learning in higher education*. Routledge, 2013.

– M'SIEUR, J'AI AQUA-PONEY !

GESTION DES EXCUSES

- Une fois pendant le quadrimestre, chaque étudiant est autorisé à ne pas participer à un Challenge
- Automatique : il suffit de ne rien soumettre
- Aucune pénalité
- Concept de **Joker** (: **Trump Card**)
- Gagnant-gagnant
 - On se débarrasse des excuses bidons
 - On peut espérer augmenter le sentiment de contrôlabilité

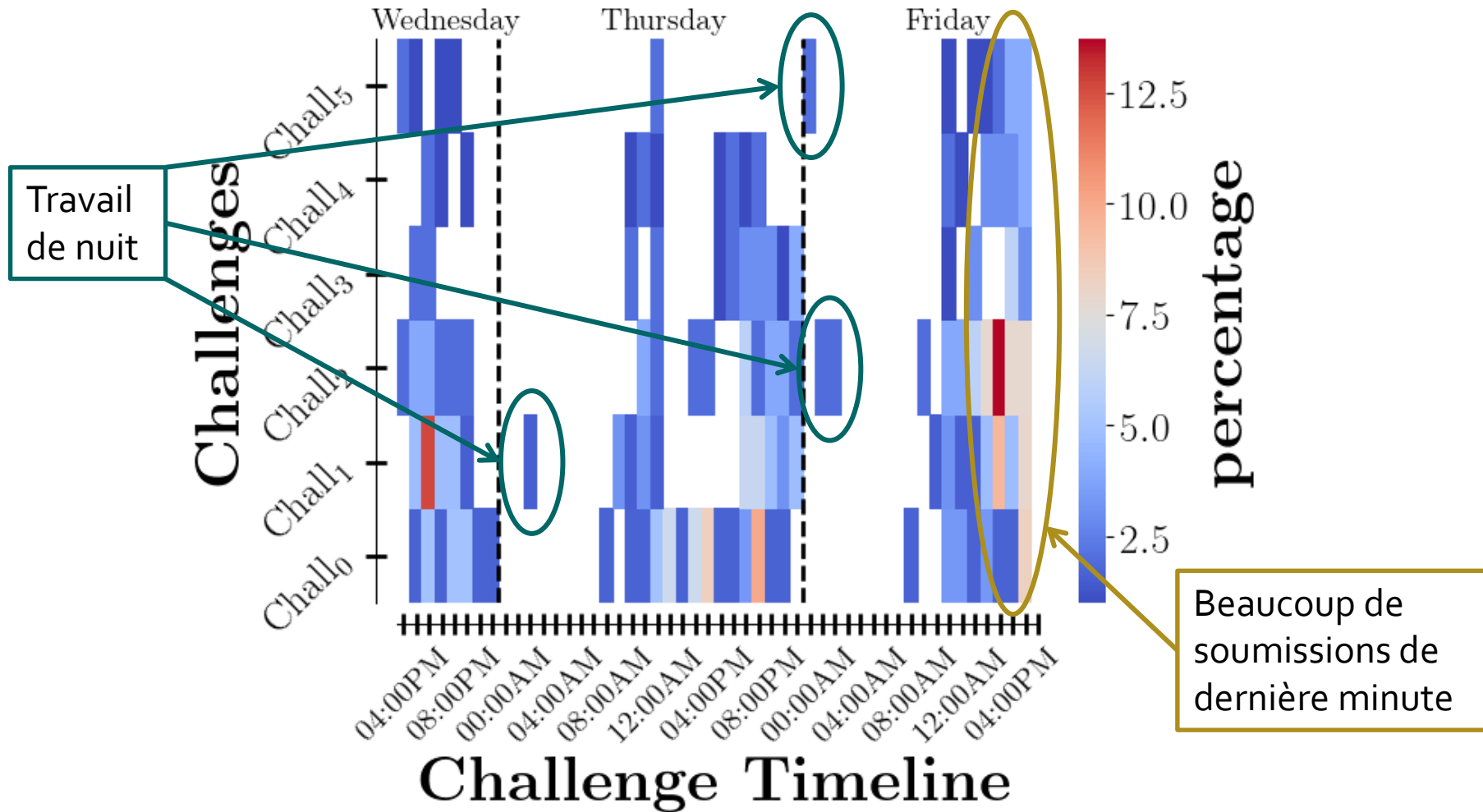
PARTICIPATION AUX CHALLENGES



- Si participation, 3 soumissions en général
- Absences s'accumulent (décrochage, après résultats des interros)

LAST CALL & PROGRAMMES DE NUIT

PARTICIPATION 2: HEURE DE PREMIÈRE SOUMISSION



FRAUDARE HUMANUM EST

GESTION DU PLAGIAT

```
while(((nb_tetes >= ((nb_pattes)/ 4)) && (nb_tetes
  <= ((nb_pattes)/2))) && ((nb_moutons *4
  + ((nb_tetes - nb_moutons)*2) != nb_pattes)){

    nb_dindons = (nb_tetes - (nb_moutons +1));
    nb_moutons++;

}
```

Ce sont 2
étudiants
différents !
[Challenge 1,
2017]

```
while(((nb_tetes >= ((nb_pattes)/ 4)) && (nb_tetes
  <= ((nb_pattes)/2))) && ((nb_moutons *4
  + ((nb_tetes - nb_moutons)*2) != nb_pattes)){

    nb_dindons = (nb_tetes - (nb_moutons +1));
    nb_moutons++;

}
```


FRAUDARE HUMANUM EST

GESTION DU PLAGIAT

```
while(((nb_tetes >= ((nb_pattes)/ 4)) && (nb_tetes  
  <= ((nb_pattes)/2))) && ((nb_moutons *4)  
  + ((nb_tetes - nb_moutons)*2) != nb_pattes)){
```

Inutile !

```
  nb_dindons = (nb_tetes - (nb_moutons +1));  
  nb_moutons++;
```

```
}
```

Un peu trop
complexe !

```
while(((nb_tetes >= ((nb_pattes)/ 4)) && (nb_tetes  
  <= ((nb_pattes)/2))) && ((nb_moutons *4)  
  + ((nb_tetes - nb_moutons)*2) != nb_pattes)){
```

```
  nb_dindons = (nb_tetes - (nb_moutons +1));  
  nb_moutons++;
```

```
}
```





Importance de la prévention dès le début du quadrimestre

FRAUDARE HUMANUM EST

GESTION DU PLAGIAT

Plus clair avec un détecteur automatique, comme MOSS

(<https://theory.stanford.edu/~aiken/moss/>)

41-94		41-94	
105-134		105-134	

```
unsigned int patte = 1;
unsigned int nb_moutons = 0 ;
unsigned int nb_dindons = 0 ;
unsigned int pt_mouton = 4 * patte ;
unsigned int pt_dindon = 2 * patte;

nb_moutons = 0;
while(((nb_tetes >= ((nb_pattes)/ 4)) && (nb_tetes <= ((nb_pattes)

    nb_dindons = (nb_tetes - (nb_moutons +1));
    nb_moutons++;
}
if(nb_pattes == nb_tetes * 2){
    nb_moutons = 0;
    nb_dindons = nb_tetes;
    printf("%d %d", nb_moutons, nb_dindons);
}else
printf("%u %u", nb_moutons, nb_dindons);
```

```
unsigned int patte = 1;
unsigned int nb_moutons = 0 ;
unsigned int nb_dindons = 0 ;
unsigned int pt_mouton = 4 * patte ;
unsigned int pt_dindon = 2 * patte;

nb_moutons = 0;
while(((nb_tetes >= ((nb_pattes)/ 4)) && (nb_tetes <= ((nb_pattes)

    nb_dindons = (nb_tetes - (nb_moutons +1));
    nb_moutons++;
}
if(nb_pattes == nb_tetes * 2){
    nb_moutons = 0;
    nb_dindons = nb_tetes;
    printf("%d %d", nb_moutons, nb_dindons);
}else
printf("%u %u", nb_moutons, nb_dindons);
```

FERMETURE DES CAFÉS

UTILISATION PAR TEMPS DE PANDÉMIE

- Cette année, utilisation pour l'interro et l'examen
 - Étudiants familiers avec le dispositif
- Pas de correction automatique
- Vérification :
 - Du bon format des fichiers soumis
 - Du bon fonctionnement de la traduction du programme
- Détection de plagiat et correction facilitées

MALGRÉ LES PRÉCAUTIONS, TOUT PEUT ARRIVER...

- Erreurs de **format** lors des soumissions
- **Interprétation** par les étudiants :
 - de l'énoncé
 - Certains se l'expliqueront entre eux comme ils l'ont compris sans poser la question à l'équipe pédagogique
 - du feedback/feedforward qu'ils reçoivent
 - Expliquer comment le lire et le comprendre
- Assez difficile de communiquer sur les **erreurs dans l'énoncé**

BILAN

- **Charge de travail** induite pour l'équipe encadrante par les Challenges
 - Design des Challenges et scripts de correction : ~ 2 sem.
- Travail fourni **impossible sans CAFÉ**
 - $(5 + 1) \times 100$ étudiants \times 3 soumissions = 1800 corrections
 - En fait, indépendant du nombre d'étudiants
- Impact positif sur la **réussite**

MERCI DE VOTRE ATTENTION !

- Sources :

- Article

- Simon Liénardy, Laurent Leduc, Dominique Verpoorten and Benoit Donnet. 2020. Café: Automatic Correction and Feedback of Programming Challenges for a CS1 Course. In *Proc. ACE '20: Twenty-Second Australasian Computing Education Conference*

- Code de CAFÉ :

- <https://github.com/slienardy/CAFE>