université de BORDEAUX

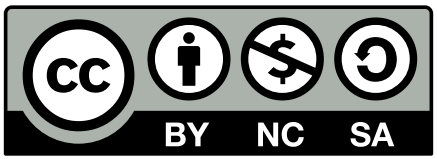
Projet de Programmation

(Comment ça marche ?)

Emmanuel Fleury
<emmanuel.fleury@u-bordeaux.fr>

LaBRI, Université de Bordeaux, France

22 janvier 2025



Plan

Plan

Objectifs du cours

université de BORDEAUX

Les objectifs de cette UE sont :

- Mise en pratique des **concepts de développement** du cours ;
- Se familiariser avec les **outils de développement** ;
- Pratiquer le **travailler en équipe**.
- S'attaquer à un **projet de taille conséquente** ;
- Apprendre à **tenir les délais et respecter les contraintes**.
- Apprendre à **faire des présentations techniques**.
- Apprendre à **rédiger un rapport technique**.

Organisation des projets

université de BORDEAUX

- Il y a **20 groupes** constitués de **4 à 6 étudiants** ;
- Il y a **10 sujets** qui sont réalisés en **C**, **Java** ou **Python** ;
- **Chaque sujet** est choisi par **2 groupes** avec des **langages de programmation différents** ;
- **Un cours** de 2h chaque semaine **pour présenter les concepts nécessaires** ;
- **Le premier mois** est consacré au **rapport préliminaire** et à l'écriture de **l'architecture du projet et de ses besoins étendues** ;
- Puis, **chaque semaine** le groupe aura **30mn pour présenter les besoins achevés** à leurs chargés de TD ;
- Chaque besoin devra être **présenté avec son code, ses tests et sa documentation** pour validation auprès des chargés de TD ;
- Les chargés de TD pourront **valider les besoins** présentés ou **demander de retravailler** certaines parties.
- À la fin du projet, pensez à réserver du temps pour écrire le **rapport final** et préparer la **présentation finale** (1h).

Agenda

Le projet se déroule sur 13 (+1) semaines :

- **Semaine 0** (aujourd'hui) :
  Constitution des **groupes** et choix des **sujets** ;
- **Semaine 1** (**Répétition Oral : 30mn**) :
  Oral : **Présentation du projet** et **répartition des besoins dans le groupe** ;
- **Semaine 2** (**Répétition Oral : 30mn, Rapport préliminaire (draft)**) :
  Oral et **Rapport préliminaire (draft)** : Corrigé Semaine 1 + **Architecture** ;
- **Semaine 3** (**Oral : 30mn, Rapport préliminaire (final)**) :
  Oral et **Rapport préliminaire** : Corrigé Semaine 2 + **Specs étendues** ;
- **Semaines 4–12** (8 semaines, Livraison besoins) :
  Livrer les **besoins** et démontrer qu'ils sont corrects ;
- **Semaine 13** (**Oral : 1h + Rapport final**) :
  Oral et **Rapport final** : Présentation du projet, architecture finale, besoins achevés, performances, critique du cahier des charges original, ...

Plan

Présentation du projet (Semaine 3)

université de BORDEAUX

La présentation du projet (30mn) doit contenir :

- Le **contexte** du projet ;
  (sujet choisi, langage choisi)
- Une **explication** détaillée du sujet ;
  (règles du jeu, structures de données employées, algorithmes spécifiques, *etc.*)
- La répartition des **besoins** dans le groupe ;
  (répartition des besoins, ébauche des dépendances entre eux, *etc.*)
- Un **agenda prévisionnel** des livraisons de besoins.
  (quels besoins ?)

Rapport préliminaire (final, semaine 3)

université de BORDEAUX

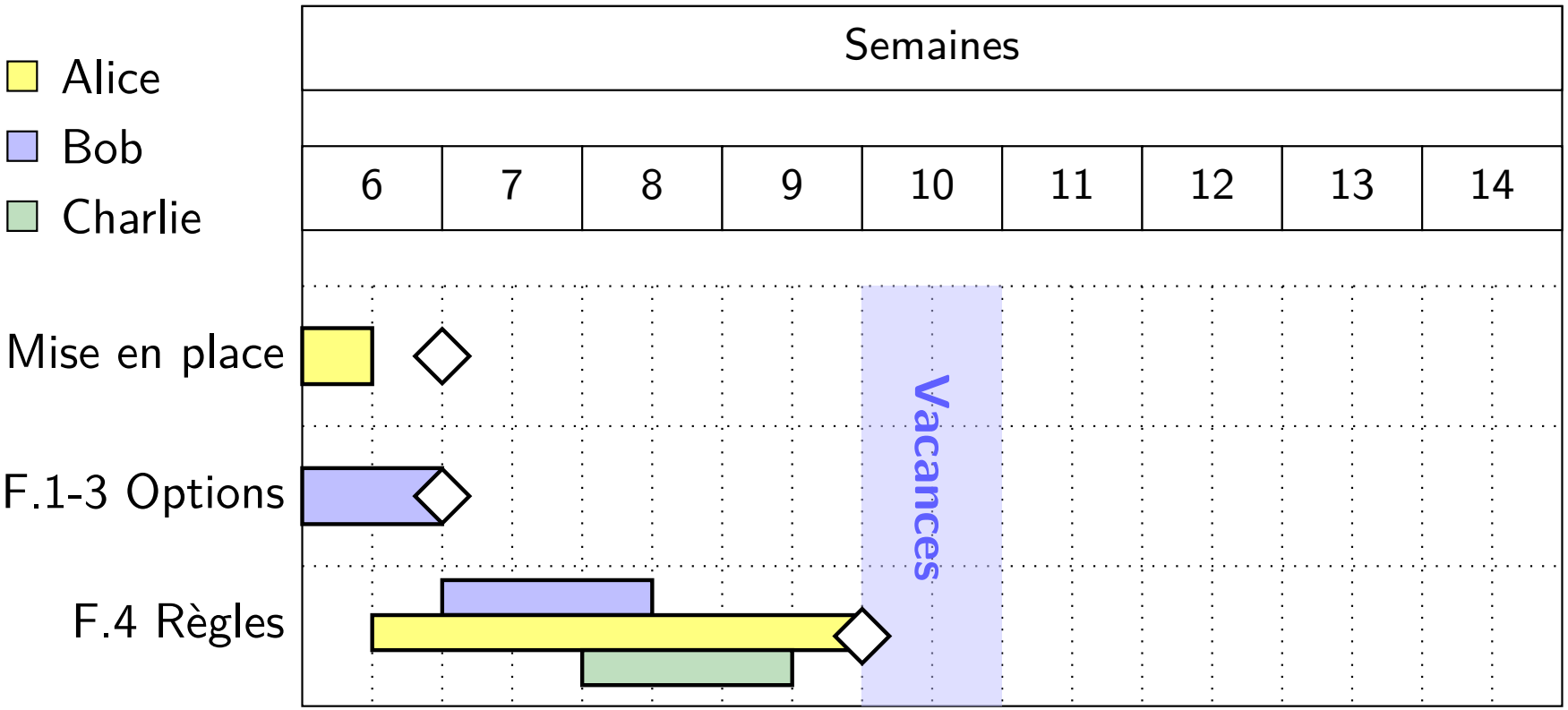
Le rapport (écrit en LaTeX) doit contenir :

- Le **contexte** et l'**existant** du projet ;
  (sujet choisi, langage choisi, bibliographie associée avec au moins 5 références)
- Une **explication** détaillée du sujet ;
  (règles du jeu, structures de données employées, algorithmes spécifiques, *etc.*)
- La liste des **besoins visés** ;
  (besoins visés, dépendances entre eux, *etc.*)
- Un **agenda** prévisionnel des livraison de besoins ;
  (quels besoins pour quelle semaine et par qui ?)
- L'**architecture** visée pour le projet ;
  (découpage en modules, dépendances, *etc.*)
- Les **spécifications étendues** pour chaque besoin visé.
  (raffinement sur les spécifications initiales, découpage en sous-besoins, ce qu'il faut à mettre en place pour valider chacun des besoins visés, *etc.*)

Agenda du projet

L'**agenda du projet** doit donner une idée de la répartition du temps de travail anticipé de chaque membre du groupe sur chaque besoin visé et la date de livraison prévue.Vous devez chacun travailler **deux demies-journées par semaine** sur le projet (environ 10h par semaine).

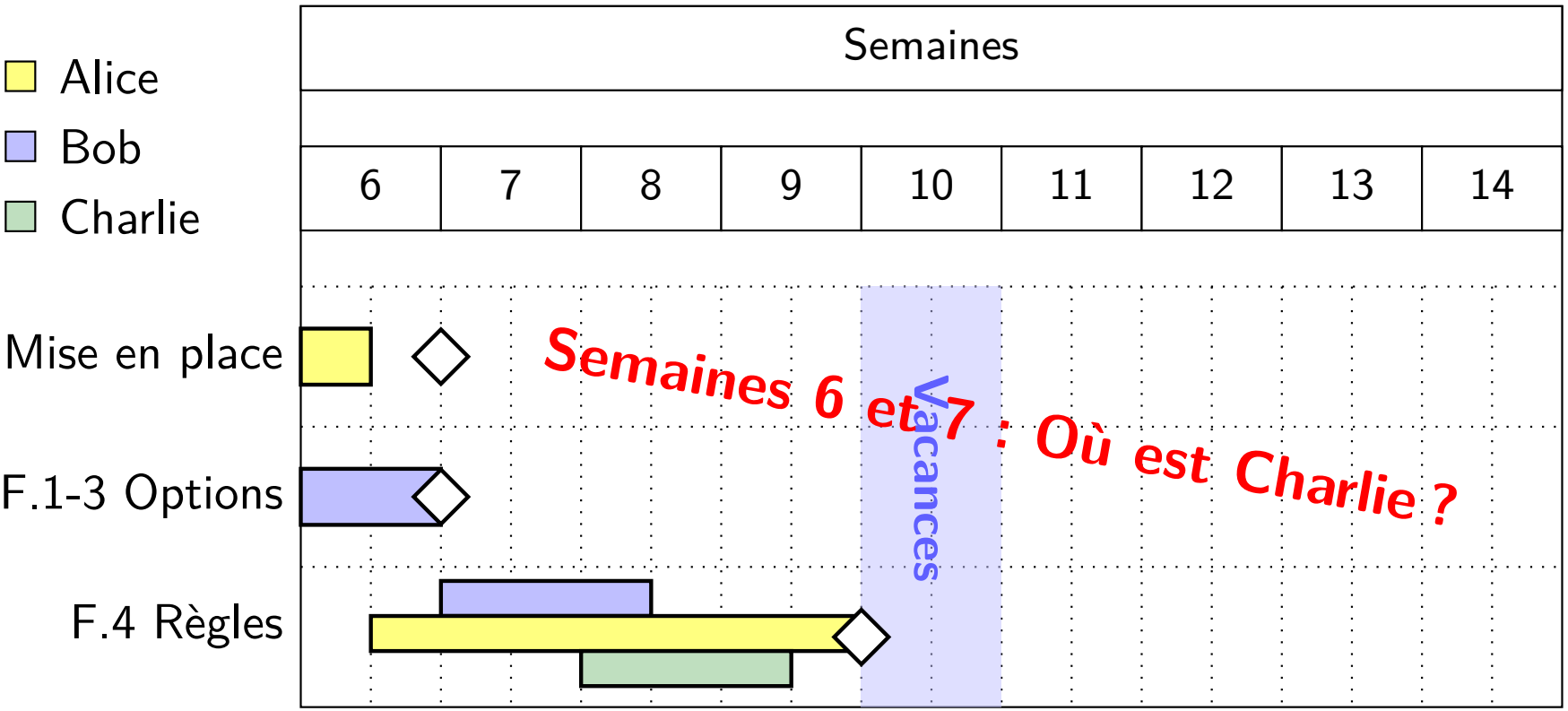
Voici un exemple de diagramme de Gantt pour l'agenda du projet :



Agenda du projet

L'**agenda du projet** doit donner une idée de la répartition du temps de travail anticipé de chaque membre du groupe sur chaque besoin visé et la date de livraison prévue.Vous devez chacun travailler **deux demies-journées par semaine** sur le projet (environ 10h par semaine).

Voici un exemple de diagramme de Gantt pour l'agenda du projet :



Spécifications étendues

Les **spécifications étendues** sont un raffinement des spécifications initiales, elles listent les sous-besoins et les actions à mettre en place pour réaliser le besoin visé.

Pour chaque besoin fonctionnel de la spécification :

- **Développeur(s) en charge** ;
- **Date de livraison** prévue ;
- Déterminer si c'est un **besoin fonctionnel** ou un **besoin non-fonctionnel** ;
- Listez les **sous-besoins** nécessaires pour réaliser le besoin visé ;
- Description des **actions** à mettre en place pour réaliser chaque sous-besoin ;
- Si c'est pertinent, pour chaque sous-besoin, **nommez quelques fonctions** qui permettront de les réaliser, **donnez leur signature** et **quelques tests unitaires** (quoi en entrée ? quoi en sortie ?) sur ces fonctions ;
- Enfin, expliquez comment **valider le besoin** une fois tous les sous-besoins réalisés avec des tests ou bien des scénarios précis.

Spécifications étendues (Exemples 1/2)

université de BORDEAUX

F4. Système cible

Le programme devra fonctionner sur des systèmes d'exploitation GNU/Linux.

Exemples de points à intégrer dans la réponse :

- Besoin non-fonctionnel ;
- Utiliser une Ubuntu 20.04 LTS pour le développement ;
- Tester sur une Debian 10.0 et une Fedora (compilation et tests automatiques sur une image Docker).

F17. Version

L'option '`-V`', '`--version`' affiche la version puis quitte.

Exemples de points à intégrer dans la réponse :

- Besoin fonctionnel ;
- Fonctionnalité gérée par `getopt_long()` ;
- Tester avec '`-V`' et '`--version`' et vérifier que la version est bien affichée et que le programme se termine avec un code `EXIT_SUCCESS`.

Spécifications étendues (Exemples 2/2)

université de BORDEAUX

F21. Conseil aux joueurs

Sur demande du joueur courant, le programme pourra suggérer un coup à jouer.

Exemples de points à intégrer dans la réponse :

- Besoin fonctionnel ;
- Prérequis : F9. (Interface utilisateur), F12. (Joueur artificiel) et F37. (Heuristiques) ;
- Requiert une fonction `suggest_move()` qui retourne un coup valide (basée sur la fonction `search_tree()` qui va rechercher le meilleur coup possible en fonction d'une heuristique donnée qui sera implémentée par F37.) ;
- Signature de la fonction :
  ```
  move_t suggest_move(board_t *board, player_t *player)
  ```
- Tester que la fonction retourne bien un coup valide (vérifier le coup avec un arbitre) pour un plateau de jeu et un joueur donné .

Plan

Contenu du projet Gitlab (CREMI)

Le Gitlab du CREMI servira de référence pour la livraison du code hebdomadaire et final. Les éléments suivants seront considérés comme faisant partie du rendu :

- **Le code source** du projet sur la branche `main` ;
- **Les tests** automatisés ;
- **La documentation** du code ;
- **L'historique** des commits sur `git` ;

Le projet devra aussi contenir un répertoire '`reports/`' qui contiendra :

- Le code LaTeX du **rapport préliminaire et final** ;
- Les PDF des slides des **présentations préliminaires et finale** ;

Validation des besoins

Chaque semaine, le groupe devra présenter des besoins validés.

- Les besoins devront être **présentés sous forme de commits ou de code**, avec leur **tests** et leurs **documentation** ;
- Les chargés de TD pourront **valider les besoins** présentés ou **demander de retravailler** certaines parties ;
- Les besoins validés seront **comptabilisés dans la note finale** du projet (1/3 de point par besoin) après avoir été revérifiés à la fin du projet.
- Les **besoins refusés devront être retravaillés** et présentés à nouveau la semaine suivante.
- Les **besoins non validés** à la fin du projet ne seront **pas comptabilisés dans la note finale**.

Plan

Rapport final (Semaine 12)

université de BORDEAUX

Le rapport (écrit en LaTeX) doit contenir (au moins) :

- Le **contexte** et l'**existant** du projet ;
  (sujet choisi, langage choisi, bibliographie associée avec au moins 10 références)
- La liste des **besoins réalisés** ;
  (besoins effectivement réalisés pendant le projet)
- Une **explication** détaillée du projet final ;
  (règles du jeu, structures de données employées, algorithmes spécifiques, *etc.*)
- L'**architecture** finale du projet ;
  (architecture finale, en faire une critique, donner des voies d'améliorations)
- Les **tests** réalisés dans le projet.
  (les tests réalisés pour valider chacune des besoins visés)
- Une **analyse critique du projet** ;
  (performances, cas limites, limitations, bugs, améliorations possibles, *etc.*)

Présentation finale (Semaine 13)

université de BORDEAUX

La présentation finale doit contenir :

- Le **contexte** et l'**existant** du projet ;
  (sujet choisi, langage choisi)
- Une **explication** détaillée du projet final ;
  (règles du jeu, structures de données employées, algorithmes spécifiques, *etc.*)
- L'**architecture** finale du projet ;
  (architecture finale, en faire une critique, donner des voies d'améliorations)
- Les **algorithmes** et **structures de données** utilisées ;
  (détailler les algorithmes et structures de données utilisées)
- Une **analyse critique du projet** ;
  (performances, cas limites, limitations, bugs, améliorations possibles, *etc.*)
- Tout autre sujet que vous jugerez pertinent de présenter...

Notation

La note finale sera répartie de la manière suivante :

- **Présentation préliminaire : 10%**
- **Rapport préliminaire : 10%**
- **Quiz du Cours : 15%**
- **Rapport final : 15%**
- **Présentation finale : 15%**
- **Besoins validés : 35%**

Page de l'UE

www.labri.fr/perso/fleury/pages/projet-de-programmation.html

On y trouve :

- Ces slides ;
- Le planning du projet ;
- Quelques documentations utiles.

Questions ?

Conseils pour utiliser Gitlab en PdP

université de BORDEAUX

- **Créez un 'Issue' par besoin (ou sous-besoin) que vous aurez à réaliser** en lui attribuant un responsable (ou plusieurs), une date de rendu et une petite description ;
- **Créez un 'Milestone' pour chaque TD de rendu de code** pour suivre les livraisons hebdomadaires ;
- **Pensez à utiliser les 'Labels'** pour marquer les besoins fonctionnels, non-fonctionnels, les bugs, les améliorations, *etc.* ;
- **Assignez une tâche 'Coordination'** à un membre du groupe pour suivre les livraisons et les validations ;
- **Pensez à utiliser les 'Discussions'** pour discuter des besoins, des problèmes rencontrés, des solutions possibles, *etc.* ;
- **Évitez de travailler directement sur la branche 'main'** ;
- **Ne procrastinez pas** et **ne laissez pas traîner les problèmes** (plus vous attendez, plus les problèmes se multiplient).