Une démarche de projet en NSI

José Delamare

Lycée Blaise Pascal - Rouen

jose.delamare@ac-normandie.fr

8 mai 2023

Lycée Blaise Pascal
Rouen

Présentation

- Professeur de Sciences Physiques - IESP, MPI, SNT, ISN, NSI en binôme avec un professeur de SII
- Établissement scientifique :
  - Général : PC, SVT, NSI, Maths, SI, HGGSP
  - Technologique : STI2D (notamment SIN)
  - Professionel : Technicien d'Usinage
  - Supérieur : BTS CPI, CPGE PTSI-PT, ATS
- Établissement centre-ville, à côté de grands ensembles urbains « populaires », à forte dominante masculine (+ 80 %)
- 40 élèves en première (2 groupes de 20) issus de 5 classes.
- 32 élèves en terminale (1 groupe - 2h de dédoublement) issus de 5 classes.
- 1 salle informatique 22 postes
- 50 % des élèves de terminale ont formulé un vœu en informatique dans Parcoursup

Lycée Blaise Pascal
Rouen

Contenu du BO

- Une part de l'horaire de l'enseignement d'**au moins un quart du total en classe** doit être réservée à la conception et à l'élaboration de projets conduits par des groupes de deux à quatre élèves.
- Pour ma part en terminale, 2h / semaine - 8 groupes de 4 élèves - 1 seul projet

Objectifs

- Me faire plaisir,
- Mettre les élèves sur une seule activité sur le temps long (plusieurs semaines) - limiter le « zapping »,
- Identifier les applications potentielles de la « vraie » vie,
- Travailler la créativité (et souvent la limiter),
- Permettre la gestion différenciée des élèves,
- **Travailler sans machine et retarder le moment où on passe au codage**,
- **Montrer qu'il faut réfléchir et écrire avant de coder**.

La constitution des groupes

Notre conception

- **Groupes hétérogènes**
  - 1 fort et 1 faible dans chaque groupe de 4 élèves
  - Définir un chef de projet : le plus fort
  - Le chef de projet est chargé (en théorie) de :
    - coordonner le travail des autres
    - vérifier que le cahier des charges est respecté (typage des entrées/sorties)
    - aider les membres en difficultés
- **Groupes homogènes**
  - Prévoir une statégie différenciée pour la réalisation du projet
  - Un « pack » de base et des « extensions » pour les plus forts

La constitution des groupes

Bilan

- Les groupes fonctionnent plutôt bien (pas de mésentente) même si les élèves se connaissent peu (pb de la constitution des groupes de spécialité)
- Très très gros écart de niveau entre les forts et les faibles.
- Les plus faibles ont progressé :
  - typage des variables
  - notion de variable locale - variable globale
  - appel de fonction
  - debuggage
- Les plus forts travaillent encore plus vite - travail sur la propreté du code
- Certaines personnalités se révèlent en-dehors des apprentissages classiques (capacité à organiser, capacité à synthétiser)

Les outils de développement

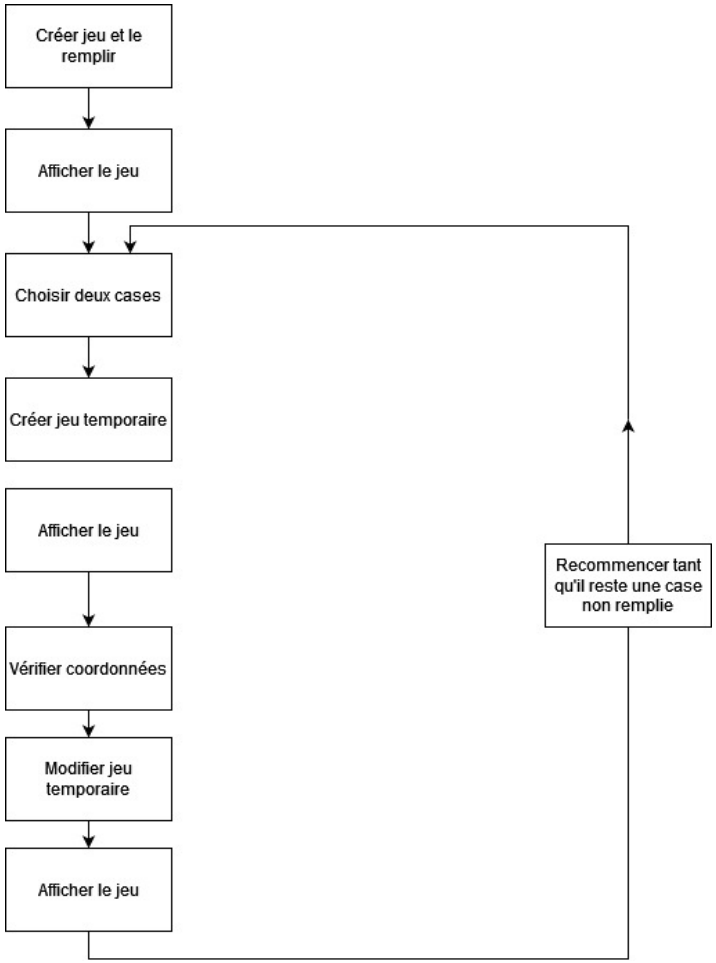
À mettre à la disposition des élèves

- Un cahier des charges **précis** Exemple
  - Règle du jeu **très précises**.
  - Ce qu'ils doivent produire / ce qu'ils ne doivent pas produire
  - Les modules à utiliser / ne pas utiliser
  - Les structures de données à utiliser / ne pas utiliser
- Un exemple de ce qu'il faut obtenir
- Un échéancier
- Un IDE

Le déroulé

Réalisation d'une « Time-line »

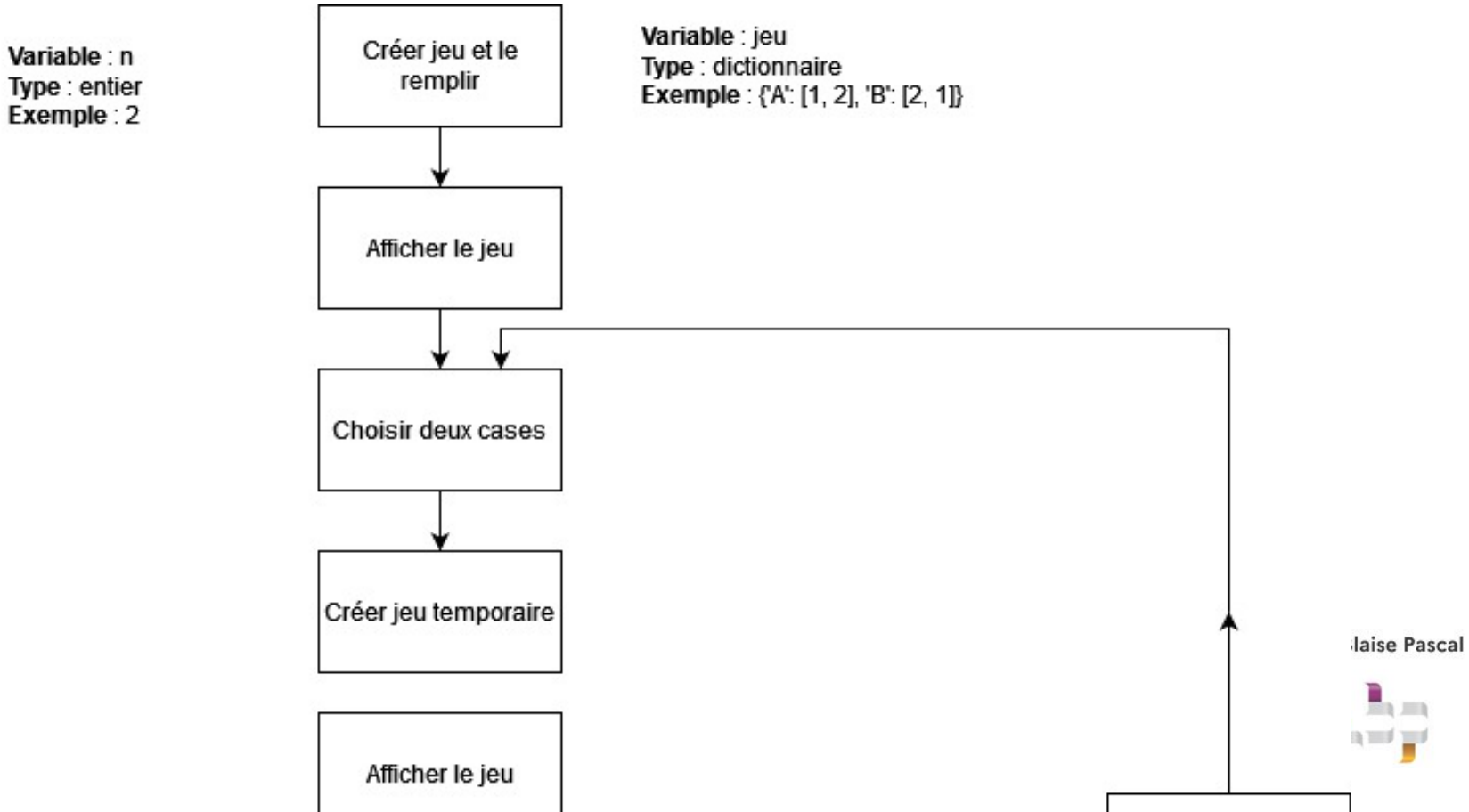
1. Identifier les différentes actions qui permettront de répondre au cahier des charges.
2. Ordonner ces actions - définir une boucle de jeu



Le déroulé

Réalisation d'une « Time-line »

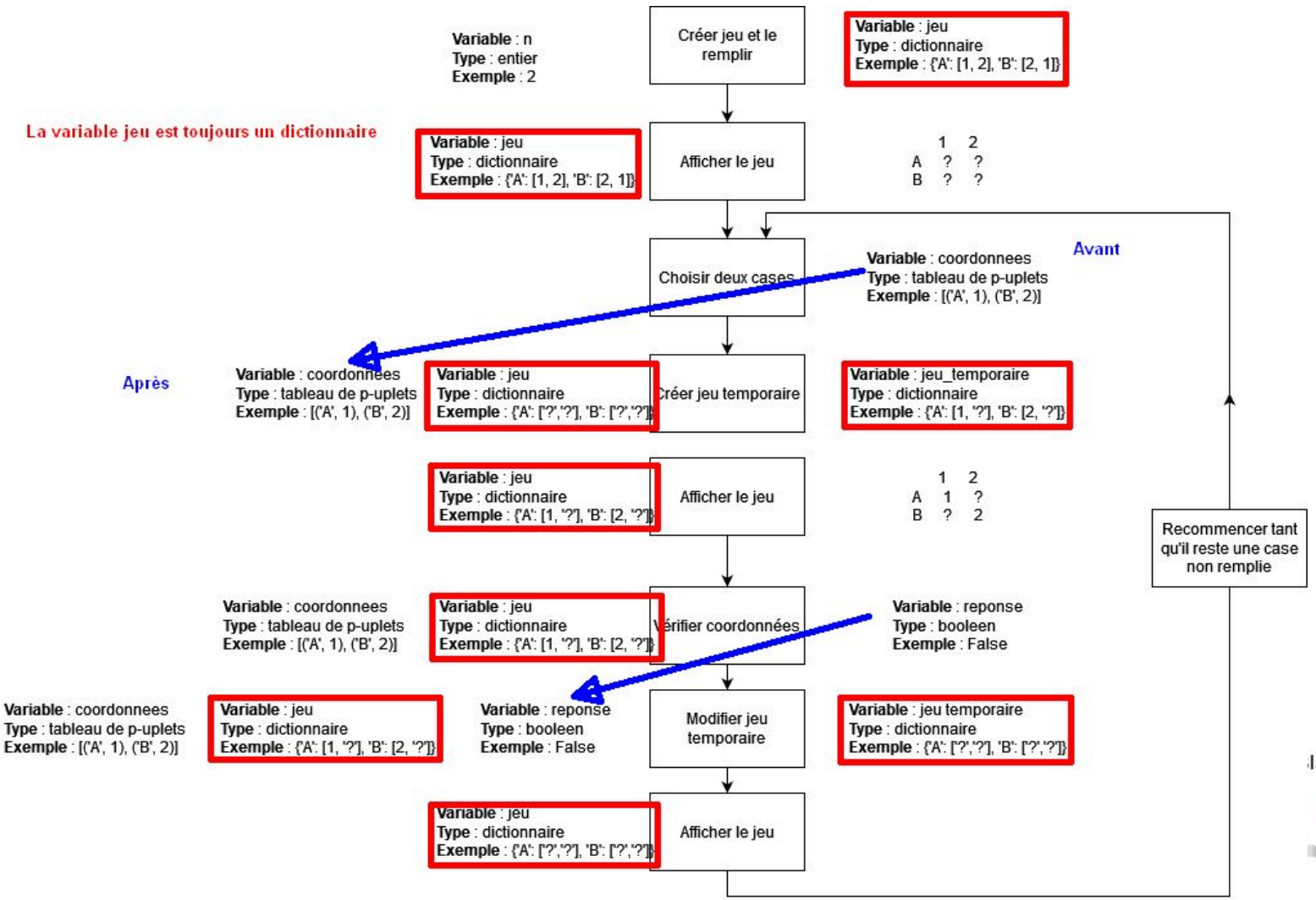
3. Identifier les entrées et les sorties de chaque action
   - nom de la variable
   - type
   - un exemple



Le déroulé

Réalisation d'une « Time-line »

4. Vérifier la cohérence sur l'intégralité de la Time-line



laise Pascal

Le déroulé

Réalisation d'une « Time-line » - Bilan

Bilan

- Identification des actions très facile.
- Permet de discuter des différentes options possibles **et d'en choisir une qui s'impose à tout le monde**.
- Permet d'évacuer tout ce qui est test de saisies valides (on considère que l'utilisateur répond correctement) - Cela peut servir pour les plus rapides.
- Difficultés pour beaucoup à définir une structure de données et d'en donner un exemple
  - manque de connaissances ?
  - peur de donner une mauvaise réponse ?
  - c'est difficile de choisir ?
  - « c'est quoi qui est le mieux ? ? ? »
  - « faut que je fasse quoi ? ? ? »

Le déroulé

Écriture de la documentation des fonctions

1. 1 action = 1 fonction
2. Écriture de la documentation (on ne la trouve jamais sur internet ...)
3. Écriture des assertions sur les préconditions
4. Écriture des assertions sur les postconditions
5. Écriture d'un test exécutable avec `doctest`
6. On réutilise la Time-line (nom variable, type, exemple)
7. Façon détournée de se réapproprier le cahier des charges et la Time-line
8. **Partie non-technique, qui ne demande pas de savoir coder**

```python
def verifier(jeu, coordonnees):
    """
    renvoie True si les 2 cases contiennent le même élément et False sinon

    Parameters
    ----------
    jeu : dict
        les clés sont des lettres et les valeurs sont des tableaux
    coordonnees : list
        un tableau de p-uplets.

    Returns
    -------
    reponse : bool

    Exemple
    -------
    >>> verifier({'A': [1, '?'], 'B': ['?', 2]}, [('A', 1), ('B', 2)])
    False
    >>> verifier({'A': [1, '?'], 'B': [1, '?']}, [('A', 1), ('B', 1)])
    True
    """
    # préconditions
    assert type(jeu) is dict
    assert type(jeu['A']) is list
    assert 'A' in jeu
    assert len(jeu) == len(jeu['A'])

    assert type(coordonnees) is list
    assert type(coordonnees[0]) is tuple
    assert len(coordonnees) == 2
    assert len(coordonnees[0]) == 2
    assert len(coordonnees[1]) == 2

    # code
```


Le déroulé

Écriture de la documentation des fonctions - bilan

1. Partie documentation : plutôt réussie pour un élève appliqué et sérieux (pas le cas de tout le monde)
2. Partie tests : la notion d'appel de fonction est difficile (« je mets quoi entre les parenthèses ? »)
   - on revoit la notion de variable locale et de variable globale
   - on revoit la notion d'exemple (c'est juste un truc pour essayer ...)
   - on appuie sur la Time-line et l'utilité d'écrire sur du papier **EN PREMIER**
3. Partie préconditions et postconditions : plutôt réussie car il y a peu d'instructions à connaître (`type()`, `len()`, `==`, `>`, ...)

Le déroulé

Codage des fonctions

1. **Le corrigé de l'étape d'avant a été fourni à tous les élèves pour leur permettre d'avancer sans trop de décalage dans le rythme.**
2. Écriture du code
3. Tests avec les exemples et `doctest`

Bilan

- Répartition des fonctions par le chef de projet sous surveillance de l'enseignant et suivant les compétences de chacun
- Certaines fonctions sont difficiles à coder pour les moins bons (boucle, test)
- Assez peu d'entraide à l'intérieur du groupe - les élèves restent sur leur blocage
- Difficile pour l'enseignant de répondre à toutes les sollicictations (surtout s'il y en a 32 en même temps ...)

Le déroulé

Utilisation des fonctions dans le programme principal

1. **Le corrigé de l'étape d'avant a été fourni à tous les élèves pour leur permettre d'avancer sans trop de décalage dans le rythme.**
2. Écriture du programme principal

Bilan

- En général c'est LE meilleur élève qui prend les choses en main.
- Pas toujours d'aboutissement final.
- Partie souvent décevante.

L'évaluation

Quelques mots sur l'évaluation 😬 😱

- Grille d'évaluation et échéancier fournis dès le départ
- 4 des étapes de validation (et de notation) :
  - Analyse du projet (pas de code)
  - Documentation - préconditions - postconditions
  - Codage des fonctions
  - Programme principal
- Le code représente moins de la moitié des points
- Présentation orale ? Rapport écrit ?

10/10/2022	Copie de la time-line ; entrées-sorties ; répartition des tâches.
17/10/2022	fichier python **INDIVIDUEL** avec : ▷ la documentation complète comprenant : ◦ ce que fait la fonction ◦ la définition complète des arguments (nom, type, usage) ◦ deux ou trois exemples d'exécution. ▷ les assertions sur les préconditions. ▷ les assertions sur les postconditions.
24/10/2022	fichier python **INDIVIDUEL** avec la documentation complète, les assertions sur les préconditions, les assertions sur les postconditions **ET** le code complet vérifiant les exemples.
31/10/2022	dossier **COLLECTIF** réunissant le travail des élèves **ET** le programme principal même s'il n'est pas complet.

Arborescence attendue

```
/
└── Memory
    ├── fonctions_eleve_1.py
    ├── fonctions_eleve_2.py
    ├── fonctions_eleve_3.py
    ├── fonctions_eleve_4.py
    └── memory.py
```

ascal

Annexe 3 - Critères d'évaluation

Critères d'évaluation		Note
Rédaction de la time-line	Complète : 3 Partielle mais en quantité/qualité suffisante : 2 Partielle et en quantité/qualité insuffisante : 1 Fournie au groupe : 0	
Répartition des tâches	Sans aucune aide : 2 Avec une aide : 1 Fournie au groupe : 0	
Rédaction de la documentation	Complète et explicite : 2 Incomplète ou peu explicite : 1 Non fournie : 0	
Rédaction des post-conditions	Complète : 3 Partielle mais en quantité/qualité suffisante : 2 Partielle et en quantité/qualité insuffisante : 1 Non fournie : 0	
Rédaction des préconditions	Complète : 2 Partielle : 1 Non fournie : 0	
Écriture du code	Valide totalement le cahier des charges : 4 Valide majoritairement le cahier des charges : 3 Valide minoritairement le cahier des charges : 2 Ne valide pas le cahier des charges : 1 Code non fourni : 0	
Écriture du code	Le code est clair **et** structuré : 2 Le code est clair **ou** structuré : 1 Le code n'est **ni** clair **ni** structuré : 0	
Écriture du code du programme principal	Toutes les fonctions sont utilisées : 2 Quelques fonctions sont utilisées : 1 Aucune fonction n'est utilisée : 0	

Bilan

Côté élève

Ce que les élèves aiment :

- On peut parler 🤩 voire faire autre chose ...
- On a de l'aide à la maison (famille, parents, amis, internet, chatGPT, ...)
- On a une bonne note

Ce que les élèves n'aiment pas :

- C'est difficile
- C'est long - lassitude au bout de 3 semaines

Bilan

Côté prof

Ce que mon collègue et moi aimons :

- On peut parler 🤩 voire faire autre chose ...
- La relation au « savoir » est réellement différente, surtout en terminale
- On apprend des choses par les très bons
- Il y a une réelle progression des élèves en matière de programmation et/ou de comportement

Ce que mon collègue et moi n'aimons pas :

- C'est chronophage
- On retrouve nos corrigés chez les élèves l'année d'après.
- On perd les élèves qui ne jouent pas le jeu

The End