

Cycle de vie d'un
projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de
tests

Puissance 4 Test unitaire

Christophe Viroulaud

Première - NSI

Lang 11



Constat

Le projet *Puissance 4* est composé de plusieurs fichiers et contient de nombreuses fonctions.

Cycle de vie d'un
projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de
tests

Comment détecter les erreurs dans un programme
informatique ?

1. Cycle de vie d'un projet

2. Test unitaire

3. Autres phases de tests

Cycle de vie d'un
projet

Test unitaire

Rôle

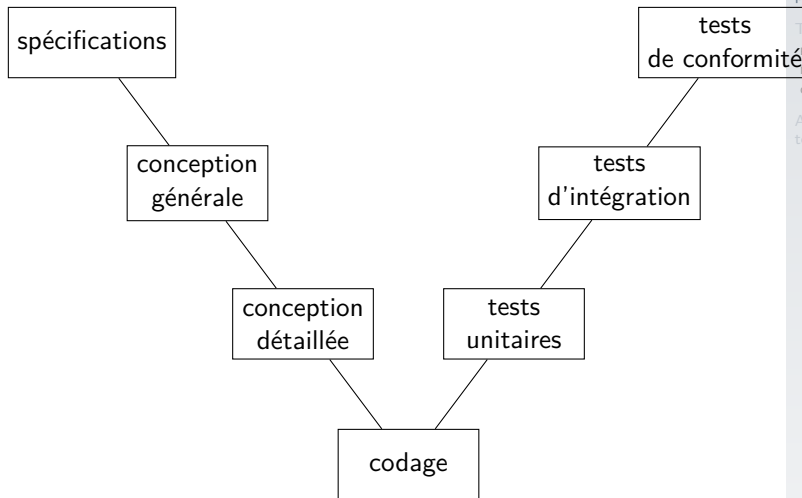
Mise en place

Construction des tests

Autres phases de
tests

Cycle de vie d'un projet

Puissance 4
Test unitaire



Cycle de vie d'un projet

Test unitaire

Rôle
Mise en place

Construction des tests

Autres phases de tests

À retenir

- ▶ Un projet est découpé en plusieurs étapes.
- ▶ Chaque étape peut être réalisée par des équipes différentes.
- ▶ Les tests peuvent prendre plus de la moitié du temps de réalisation.

Cycle de vie d'un
projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de
tests

1. Cycle de vie d'un projet

2. Test unitaire

2.1 Rôle

2.2 Mise en place

2.3 Construction des tests

3. Autres phases de tests

Cycle de vie d'un projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de tests

Spécifications :
règles du jeu

Conception générale :
déroulé du jeu

Conception détaillée :
découpage en fonctions
signatures des fonctions

Tests unitaires :
vérification
comportement
de chaque fonction

Codage :
écriture du programme principal
écriture des fonctions

Conception détaillée :
découpage en fonctions
signatures fonctions



Tests unitaires :
vérification
comportement
de chaque fonction

À retenir

Un **test unitaire** permet de garantir le comportement d'une fonction **dans chaque cas de figure**.

1. Cycle de vie d'un projet

2. Test unitaire

2.1 Rôle

2.2 Mise en place

2.3 Construction des tests

3. Autres phases de tests

Cycle de vie d'un
projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de
tests

À retenir

Tester une fonction consiste à effectuer des appels qui prennent en compte l'ensemble des cas de figure du code de la fonction.

Il faut donc écrire un programme principal qui appelle chaque fonction avec des arguments choisis.

Hors programme

La ligne

```
1 if __name__ == "__main__":
```

permet d'exécuter le programme principal uniquement si le fichier est exécuté directement (et pas importé).

Activité 1 :

1. Dans le fichier `fonctions_placement.py`, écrire le code suivant :

```
1  if __name__ == "__main__":  
2      # provoque une erreur  
3      placer_jeton([], LARGEUR, ROUGE)
```

Code 1 – On essaie de placer un jeton ROUGE dans une grille vide.

2. Exécuter le fichier `fonctions_placement.py` directement.
3. Exécuter le fichier `puissance4_console.py`

```
1 while lig+1 < HAUTEUR and grille[lig+1][col] == VIDE:  
2 IndexError: list index out of range
```

Code 2 – Exécution du fichier `fonctions_placement.py`
directement.

Remarque

Quand le fichier `puissance4_console.py` est exécuté,
le test n'est pas effectué.

1. Cycle de vie d'un projet

2. Test unitaire

2.1 Rôle

2.2 Mise en place

2.3 Construction des tests

3. Autres phases de tests

Cycle de vie d'un
projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de
tests

Cycle de vie d'un
projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de
tests

Pour effectuer des tests, il faut mettre en place des situations.

Activité 2 :

1. Dans le programme principal de `fonctions_placement.py`, écrire le code suivant :

```
1 grille_test = [ [VIDE for i in range(LARGEUR)]
2                  for j in range(HAUTEUR) ]
3
4 # rempli la première colonne
5 for i in range(HAUTEUR):
6     grille_test[i][0] = JAUNE
7 # place 2 jetons dans colonne 3
8 grille_test[5][3] = JAUNE
9 grille_test[4][3] = JAUNE
```

2. Tester alors la fonction `est_remplie` dans deux cas de figures.

```
1 if __name__ == "__main__":
2     grille_test = [ [VIDE for i in range(LARGEUR)]
3                     for j in range(HAUTEUR) ]
4
5     # rempli la première colonne
6     for i in range(HAUTEUR):
7         grille_test[i][0] = JAUNE
8     # place 2 jetons dans colonne 3
9     grille_test[5][3] = JAUNE
10    grille_test[4][3] = JAUNE
11
12    print( est_remplie(grille_test, 0) )
13    print( est_remplie(grille_test, 3) )
```

Activité 3 : Dans le fichier `fonctions_verif.py`, mettre en place des tests pour la fonction `verif_horizontale`.

Indication : prendre le temps de lire la documentation de la fonction pour déterminer les tests possibles.

```
1 if __name__ == "__main__":
2     grille_test = [ [VIDE for i in range(LARGEUR)]
3                     for j in range(HAUTEUR) ]
4
5     # place 4 JAUNE en haut de la grille
6     grille_test[0][3] = JAUNE
7     grille_test[0][4] = JAUNE
8     grille_test[0][5] = JAUNE
9     grille_test[0][6] = JAUNE
10
11     print( verif_horizontale(grille_test, JAUNE, 0, 6) )
```

Code 3 – Ce test n'est pas exhaustif.

À retenir

Mettre un jeu de tests unitaires en place est une tâche longue et difficile.

Cycle de vie d'un
projet

Test unitaire

Rôle

Mise en place

Construction des tests

Autres phases de
tests

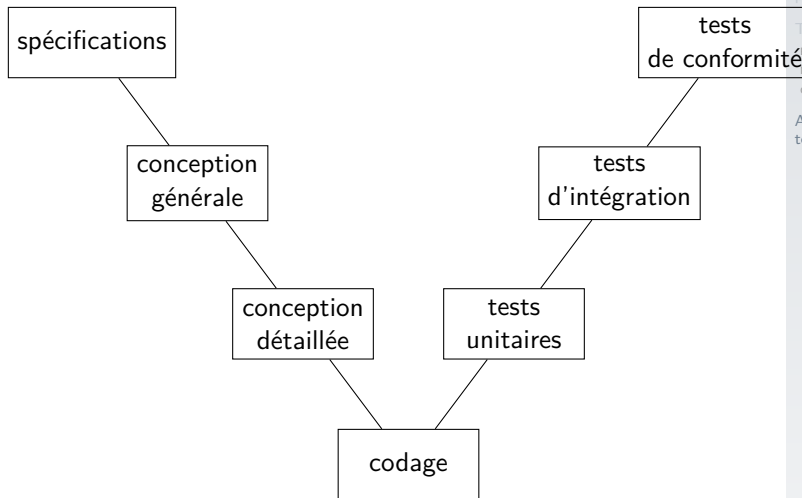
1. Cycle de vie d'un projet

2. Test unitaire

3. Autres phases de tests

Autres phases de tests

Puissance 4
Test unitaire



Cycle de vie d'un projet

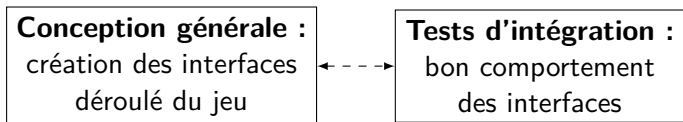
Test unitaire

Rôle

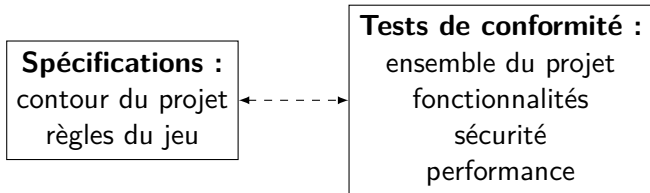
Mise en place

Construction des tests

Autres phases de tests



Exemple : Respect des différentes séquences du jeu
(positionnement graphique du jeton...)



Exemple : Respect des règles du jeu

Remarque

Ces deux autres phases de tests ne seront pas étudiées en détail en classe de première.

À retenir

Les phases de tests sont une étape indispensable du cycle de vie d'un projet : elles permettent de maintenir la cohérence du code lors des modifications, ajouts...