



# global AI bootcamp

2-4 March 2023

## ChatTDD ou comment faire écrire des tests à Chat GPT



**Youen Péron**

Architecte Technique - CGI



# Les modèles de langage pour écrire du code

2015 : RNN

2017 : SeqGAN

2021 : Copilot / Codex : Completion automatique du code



2023 : ChatGPT : agent conversationnel parlant plusieurs langues dont python !







Lantao Yu<sup>†</sup>, Weinan Zhang<sup>†\*</sup>, Jun Wang<sup>‡</sup>, Yong Yu<sup>†</sup>  
<sup>†</sup>Shanghai Jiao Tong University, <sup>‡</sup>University College London  
{yulantao,wnzhang,yyu}@apex.sjtu.edu.cn, j.wang@cs.ucl.ac.uk



# Exemple de problème classique : le bowling

 propose moi une fonction python pour calculer les scores d'une partie de bowling 

 Voici une proposition de fonction Python pour calculer les scores d'une partie de bowling :  

```
python  Copy code

def calculate_bowling_scores(frames):
    total_score = 0
    frame_index = 0

    for frame in range(10):
        if frames[frame_index] == 10: # Strike
            total_score += 10 + frames[frame_index + 1] + frames[frame_index + 2]
            frame_index += 1
        elif frames[frame_index] + frames[frame_index + 1] == 10: # Spare
            total_score += 10 + frames[frame_index + 2]
            frame_index += 2
        else: # Open frame
            total_score += frames[frame_index] + frames[frame_index + 1]
            frame_index += 2

    return total_score
```

Cette fonction prend une liste `frames` contenant les scores de chaque lancer dans l'ordre chronologique. Par exemple, si le joueur a marqué 9, 1, 5, 5, 10, 8, 0, 10, 10, 9, 1, 10, 10, 10, 8, la liste `frames` serait `[9, 1, 5, 5, 10, 8, 0, 10, 10, 9, 1, 10, 10, 10, 8]`.



Il a les concepts de

- Frames
- Strikes
- Spare
- Open frame
- 10 pins

Il me propose des données de test.


# Où a-t-il trouvé cette connaissance ?





 bowling  Pull requests Issues Codespaces Marketplace Explore

Repositories	12K
Code	4M
Commits	54K
Issues	16K
Discussions	28
Packages	0
Marketplace	0
Topics	15
Wikis	1K
Users	401

12,504 repository results

 [makersacademy/bowling-challenge](#)  
Calculate Scores for Ten Pin Bowling  
★ 15 Updated on Oct 24, 2022

 [next-step/java-bowling](#)  
볼링 게임 점수판 구현을 위한 저장소  
★ 14 ● Java Updated on Aug 7, 2022

 [12meses12katas/Abril-Bowling](#)  
Kata del mes de Abril de 2011: Bowling Game  
★ 64 ● JavaScript Updated on Jun 17, 2015





# Son code est-il valide ?

- Autant que le vôtre !
- Les risques :
  - La spécification n'est pas claire
  - Il n'a pas bien interprété la question
  - Il a oublié un cas d'usage
  - Le code n'est pas optimisé
  - Il va vous proposer une solution même si c'est au-dessus de ses capacités !
  - Son code est très crédible (il masque bien les bugs !)



# La nouvelle règle de Bowling "One Point"



User Story: Ajouter la règle du "one point" pour les joueurs qui ne font pas de spare ou de strike.

En tant que joueur de bowling,  
je veux pouvoir lancer un troisième tir si je ne fais pas de spare ou de strike,  
pour que si je fais tomber qu'une seule quille avec mon troisième tir, je gagne un "one point" et un bonus,  
et que mon score final soit correctement calculé.

Critères d'acceptation:

- Si le joueur ne fait pas de spare ou de strike, il peut lancer un troisième tir.
- Si le joueur fait tomber qu'une seule quille avec son troisième tir, il gagne un "one point" et un bonus égal au produit du nombre de quilles tombées lors du premier et du deuxième tir.
- Le score final doit être correctement calculé, en incluant les "one points" et les bonus pour les joueurs qui en gagnent.

# Implémentation par ChatGPT



Voici une proposition de modification de la fonction `calculate\_bowling\_scores` pour implémenter la règle du "one point":

python

Copy code

```
def calculate_bowling_scores(frames):  
    total_score = 0  
    frame_index = 0  
  
    for frame in range(10):  
        if frames[frame_index] == 10: # Strike  
            total_score += 10 + frames[frame_index + 1] + frames[frame_index + 2]  
            frame_index += 1  
        elif frames[frame_index] + frames[frame_index + 1] == 10: # Spare  
            total_score += 10 + frames[frame_index + 2]  
            frame_index += 2  
        else: # Open frame  
            total_score += frames[frame_index] + frames[frame_index + 1]  
            if frame == 9 and frames[frame_index + 1] + frames[frame_index + 2] == 10:  
                total_score += 1 + frames[frame_index + 1] * frames[frame_index + 2]  
                break  
            elif frame == 9:  
                frame_index += 2  
            else:  
                frame_index += 2  
  
    return total_score
```





# TDD : Test Driven Development

- Objectif : définir des contrats interfaces
- Effets de bords :
  - 99 % de coverage
  - Concentration optimisée
  - Pas de sur-qualité
- Méthode :
  - 1 – écrire un test
  - 2 – coder (sans réfléchir)
  - 3 – refactorer (prendre de la hauteur)





# Démo : Bowling en ChatTDD

- Objectif : US « One Point »
- On conserve la boucle TDD
- On intègre ChatGPT à chaque étape
- Vous me proposez des modifications ou des prompts



# Débriefing

MERCI

Crédits images :  
DALL-E

