

TP Test Unitaire – TDD

Objectif

Tests Unitaire ; TDD + Pair Programing = Ping Pong.
Implémenter le calcul du score dans une partie de tennis.
On se focalise sur un seul jeu (jeu/set/match)

Les règles du jeu

- Les points sont gagnés suivant cette séquence : 0 → 15 → 30 → 40 → Victoire
- Marquer avec un score de 40 et au moins 1 balle d'avance → Victoire
- Si les deux joueurs ont 40 points → Egalité
- Marquer avec un Egalité donne l'Avantage
- Marquer avec l'Avantage → Victoire
- L'avantage se perd si le joueur adverse gagne le point (on revient à Egalité) Spécifications
- Les joueurs peuvent marquer un point
- Une partie se termine avec un gagnant
- Pouvoir afficher le score après chaque point gagné (0-0, 0-15, 15-15, ...)

Par où commencer en QUnit ?

Ecrire un premier test

```
test('InitialisationApp', function() {  
    var jeu = new Jeu() ;  
});
```

... qui échoue => Ping (le clavier change de main)

Faire passer le test de la manière la plus simple possible

```
function Jeu(){  
} ;  
Le test passe
```

Prochaine étape : Ecrire un nouveau test

```
test('AfficherScoreDebutDePartie', function() {  
    var jeu = new Jeu() ;  
    equal("0-0",jeu.score()) ;  
});  
... qui échoue => Pong (le clavier change de main)
```

Prochaines étapes : développer l'application

Jouer au Ping Pong jusqu'à ce que tout le fonctionnel soit implémenté

Bonchon

Implémenter un Bouchon capable de sauvegarder le jeu en cours.

Injecter (ou affecter) ce bouchon à la classe de jeu.

Si le résultat en cours est « 15:15 »

jeu.save() → « 15:15 »

jeu.reset() → « 0:0 »

jeu.load() → « 15:15 »

Pour les courageux

gérer un match complet en intégrant les notions de set, le tie break, match.