

TP - Functional Penalty

Programmation Fonctionnelle



Modalités de réalisation

- **Travail individuel.**
- Pour la qualité de votre apprentissage, je vous recommande de réaliser l'exercice sans employer l'**IA générative** (*ChatGPT, Gemini, Github Copilot...*), ou de façon modérée.
- L'**auto-documentation** fait partie de l'exercice.
- En cas de problème, **n'hésitez-pas à me consulter.**

Modalités de rendu

- **Indiquez votre prénom et votre nom** dans un fichier **README.md** placé à la racine de votre projet.
- **Fournissez l'adresse de votre dépôt Git distant** (par mail ou message privé sur Discord). Si le dépôt Git est privé, ajoutez **shrp777** ou **alex@shrp.dev** en tant que collaborateur du projet.
- **Rendu au plus tard le 17/9/2024 à 17h.**

Programmation Fonctionnelle

Alexandre Leroux (alex@shrp.dev) - 2024

Consignes

- Appliquez les concepts fondamentaux de la Programmation Fonctionnelle :
 - fonctions pures et totales,
 - immutabilité,
 - absence d'effets de bord,
 - absence d'instruction impérative...
- Appliquez les concepts :
 - de récursivité,
 - de gestion d'état via un store,
- L'emploi d'une base de données et d'une interface graphique ne sont ni nécessaires, ni demandés.

Objectifs

- **Ce TD a pour objectif de vous familiariser par la pratique aux différents concepts de Programmation Fonctionnelle.**
- **Employez le langage de votre choix, sous réserve qu'il soit adapté à la Programmation Fonctionnelle et typé statiquement** (de préférence TypeScript avec le runtime Bun, sinon Rust, Scala, Kotlin, Erlang, Elixir, Haskell, Clojure, C++...).
- Exploitez autant que possible le **typage statique** (variables, input / output des fonctions).
- Employez les **types** plutôt que des *classes*, *interfaces* et *enums* afin de définir les structures de données manipulées (*Extension*, *Union*, *Produit* et *Intersection*).
- Exploitez le **typage générique**,
- Employez les **méthodes de l'objet Array** inspirées du **paradigme déclaratif** : *map*, *reduce*, *filter*, *find*, *forEach*... (à adapter selon le langage sélectionné).
- Privilégiez les **expressions de type "one line"** (condition ternaire, chaînage...),
- **N'employez pas d'instructions impératives** (if, for...).
- **Évitez autant que possible les mutations de valeurs et les variables.** Privilégiez les constantes.
- Organisez votre programme à base de fonctions **pures et totales, composables et chaînables.**

- Privilégiez les **fonctions unaires**, pour ce faire, ayez recours à la **curryfication**.
- **Évitez les effets de bord** (accès à des variables globales ou à un état partagé, mutations de valeurs, levée d'exception...).
- **Évitez la répétition dans votre code** : privilégiez une approche atomique, afin de disposer de fonctions réutilisables et combinables,
- Chaque fonction doit être vérifiée par un **test unitaire** associé.

Sujet

Une séance de tirs au but se déroule à la fin d'un match de football en cas de score de parité après les prolongations.

Dans une séance de tirs au but, chaque équipe effectue un tir, l'une après l'autre.

Le score de la séance de tirs au but est mis à jour après chaque tir : si le tir au but est marqué, le score de l'équipe augmente d'une unité.

Scénario nominal : l'équipe dont le score est le plus élevé après 5 tirs a gagné.

Scénario alternatif 1 : si au cours de la séance des 5 tirs au but, une équipe dispose d'un avantage qui ne pourra pas être rattrapé par l'équipe adverse, la séance se termine immédiatement et la victoire lui est accordée (ex: 3/0, 4/1).

Scénario alternatif 2 : en cas d'égalité après les 5 tirs au but, la séance se poursuit. Si l'une des 2 équipes prend l'avantage après que chacune ait tiré, la victoire lui est accordée.

Travail à réaliser

- Simulez le résultat de chaque tir au hasard (pénalty marqué / pénalty manqué).
- Mettez à jour le score de la séance de tirs au but après chaque tir.
- Constituez un historique du déroulé de la séance de tirs au but, permettant de retracer l'évolution du score.
- Mettez en place une fonction permettant d'afficher l'historique de la séance de tirs au but dans le terminal, dans ce style :
 - Tir 1 | Score : 1/0 (équipe A : +1, équipe B : 0)
 - Tir 2 | Score : 1/1 (équipe A : 0, équipe B : +1)
 - Tir 3 | Score : 2/2 (équipe A : +1, équipe B : +1)
 - Tir 4 | Score : 3/3 (équipe A : +1, équipe B : +1)
 - Tir 5 | Score : 4/3 (équipe A : +1, équipe B : 0)
 - Victoire : Equipe A
- Faites en sorte que votre programme se répète tant que la séance de tirs au but n'a pas permis de déterminer un vainqueur.
- Vérifiez le fonctionnement de votre programme en implémentant des **tests unitaires**.