

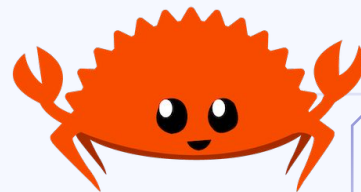
Règles communes aux projets

Rendu

Le projet Rust doit être stocké dans un dépôt Git public. Le lien est à envoyer à corentin.gauthier@eleves.ec-nantes.fr

Grille d'évaluation

- Le code compile.
- Les I/O du terminal affichent des résultats similaires à ceux des exemples.
- Chaque item doit être précédé d'un commentaire explicitant son rôle afin d'être représenté dans la documentation finale générée par `cargo --doc`.
- Toute fonction utilisant des items susceptibles de générer des erreurs doit renvoyer un type `Result<T>`.
- Toute fonction/méthode doit être testée par au moins un test unitaire
- Le code respecte les Styles Guides Rust (utiliser Clippy).



Jeu de duel

Présentation

Ce jeu se joue à tour par tour entre deux joueurs et utilisé exclusivement la touche **ENTREE** du clavier ainsi que le terminal en tant qu' I/O.

Il oppose deux joueurs disposant de caractéristique propres (nom, vitalité, vitesse, force) susceptibles de changer au cours de la partie. Chacune des caractéristiques communes aux joueurs peuvent-être paramétrés au lancement du programme mais doivent disposer de valeurs par défaut.

A chaque début de tour d'un joueur, un tableau d'objectif est généré et contient des nombres de 0 à 100 générés aléatoirement le joueur doit appuyer sur la touche entrée pour commencer à jouer.

Un compteur s'incrémentant de 0 à 100 doit alors apparaître. Le compteur est propre au joueur, son pas d'incrémentation (en ms) correspond à son membre "Vitesse". Lorsqu'une interruption est détectée par le programme (via l'appui de la touche **ENTREE**), le compteur se fige et affiche le nombre en cours puis démarre le compteur suivant sur une nouvelle ligne.

L'objectif du joueur est d'arrêter le compteur au plus près des différents nombres du tableau d'objectif. Le tour du joueur se termine lorsqu'une interruption est détectée pour chaque objectif.

Jeu de duel

Cahier des charges

- Les nombres du tableau objectifs doivent être générés aléatoirement entre 0 et 100 à chaque tour de jeu.
- Le code doit gérer le fait qu'un résultat de 95 pour un objectif de 15 entraîne une différence de 20 points et non de 80.
- Lorsque le compteur arrive à 100, il repasse à 0. Cela incrémente la variable "miss" dont la valeur par défaut est à 0.
- Les valeurs de "miss" et "compteur" doivent être up toutes les 30ms via un thread d'affichage dédié.
- A chaque objectif, le joueur gagne des points en fonction de la différence avec son score :

| Différence (absolue) | 0 | 1 à 5 | 6 à 10 | 11 à 20 | 21 à 50 | > 50 |
|-------------------------|--|---|---|---|---|--|
| Score | $(100 + \text{force}) / (\text{miss} + 1)$ | $(80 + \text{force}) / (\text{miss} + 1)$ | $(60 + \text{force}) / (\text{miss} + 1)$ | $(40 + \text{force}) / (\text{miss} + 1)$ | $(20 + \text{force}) / (\text{miss} + 1)$ | $(0 + \text{force}) / (\text{miss} + 1)$ |

- Le score final de fin de tour correspond à la moyenne de points sur chacun des objectifs.
- Le score final doit être arrondi à l'entier supérieur.
- A la fin d'une manche le joueur avec le plus de point gagne.
- Le perdant perd en vitalité la différence entre les deux scores.
- Le gagnant choisi un poison à appliquer au joueur qui impactera ses caractéristiques pour les prochaines manches (-5 de Vitesse OU -5 de Force).
- Les caractéristique d'un joueur doivent être affichés au début de son tour de jeu.
- La partie se termine lorsqu'un joueur n'a plus de vie.

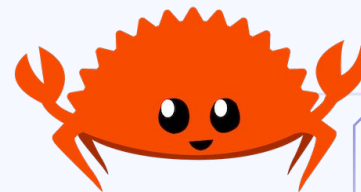


Jeu de duel

Exemple de déroulement d'un tour de jeu

```
$ cargo run --name1 Michel --name2 Jacques --vitality 50 --objectifs 5
##### Démarrage de la partie #####
## Manche 1 ##
Au tour de Michel (Vitality=50, Speed=50, Strength=50)
→ Objectifs : [50, 82, 74, 33, 95]
→ Appuyer sur ENTREE pour démarrer le tour..
→ Objectif 50 : Miss = 1 | Compteur = 36 // Score = (40 + 50) / 2 = 45
→ Objectif 82 : Miss = 0 | Compteur = 80 // Score = (80 + 50) / 1 = 130
→ Objectif 74 : Miss = 0 | Compteur = 70 // Score = (80 + 50) / 1 = 130
→ Objectif 33 : Miss = 1 | Compteur = 43 // Score = (60 + 50) / 2 = 55
→ Objectif 95 : Miss = 1 | Compteur = 90 // Score = (80 + 50) / 2 = 65
# Fin du tour #
→ Score moyen 85

Au tour de Jacques (Vitality=50, Speed=50, Strength=50)
[...]
```



Règles du jeu de duel

Exemple de déroulement d'une partie

```
$ cargo run --name1 Michel --name2 Jacques --vitality 50 --objectifs 5
##### Démarrage de la partie #####
## Manche 1 ##
Au tour de Michel (Vitality=50, Speed=50, Strength=50)
→ Objectifs : [50, 82, 74, 33, 95]
→ Appuyer sur ENTREE pour démarrer le tour..
→ [...]
→ Score moyen 82

Au tour de Jacques (Vitality=50, Speed=50, Strength=50)
→ Objectifs : [12, 88, 48, 56, 42]
→ Appuyer sur ENTREE pour démarrer le tour..
→ [...]
→ Score moyen 95

Jacques gagne la manche. Michel perd 13 points de vitalité.
Jacques vous devez choisir quel poison appliquer à Michel :
→ 1: -5 speed
→ 2: -5 strength
>2
## FIN Manche 1 ##
```



```
## Manche 2 ##
Au tour de Michel (Vitality=37, Speed=50, Strength=45)
→ Objectifs : [56, 22, 34, 63, 55]
→ Appuyer sur ENTREE pour démarrer le tour..
→ [...]
→ Score moyen 52

Au tour de Jacques (Vitality=50, Speed=50, Strength=50)
→ Objectifs : [16, 28, 98, 76, 22]
→ Appuyer sur ENTREE pour démarrer le tour..
→ [...]
→ Score moyen 92

Jacques gagne la manche. Michel perd 40 points de vitalité.
## FIN Manche 2 ##
##### Partie terminée #####
Relancer une partie ? [Y/N]
>y

##### Démarrage de la partie #####
[...]
```

Règles du jeu de duel

Variantes bonus

- Générer les objectifs sous forme Hashmap contenant les différentes lettres du clavier. L'appui sur la mauvaise touche entraîne un score de 0 sur l'objectif.
- Ex → `Objectifs: [(a: 13), (e: 45), (k: 87), (p: 64), (d: 30)]`
- Implémenter un mode avec plus de deux joueurs.

Conseils

- Crate `clap` pour la gestion des arguments en ligne de commande.
- Crate `rand` pour la génération d'objectifs aléatoires.
- Crate `log` et `env_logger` pour la gestion des différents niveaux de log avec les macros (`error!`, `warn!`, `info!`, `debug!` et `trace!`).
- Commencer par écrire les tests unitaire.

