

## 🎯 Objectif

Terminez le jeu avec un **score** plus élevé que votre adversaire.  
Trois joueurs s'affrontent dans les **olympiades de l'arcade**.

Chaque joueur contrôle un agent dans **quatre** mini-jeux **simultanément**. Gagnez un maximum de **médailles** dans les quatre jeux pour obtenir le score le plus élevé.

## ✓ Règles

Chaque joueur est connecté à **quatre** console de jeux différentes, et chacune de ces consoles exécute un **mini-jeu** différent. Votre code peut lire les 8 **registres** utilisés en interne par les consoles : **GPU**, contenant une chaîne de caractères, et **reg0** à **reg6** contenant des entiers. Ce que ces valeurs représentent est différent pour chaque jeu.

Le jeu se joue en tours. À chaque tour, les trois joueurs effectuent l'une des quatre actions possibles : **UP**, **DOWN**, **LEFT** ou **RIGHT**.

Lorsqu'une action est effectuée par un joueur, son agent dans **chaque** mini-jeu effectue la même action, car leur manette a été connectée aux 4 machines en même temps.

## Gagner des médailles

Les quatre mini-jeux se jouent en boucle tout au long du jeu. À chaque manche d'un mini-jeu, vous pouvez obtenir une médaille d'or, d'argent ou de bronze. Entre les manches, il y a un tour de **réinitialisation** où le mini-jeu est inactif.

À la fin du jeu, le score de chaque joueur pour chaque mini-jeu est calculé en fonction du nombre total de médailles gagnées, avec cette formule :

**score\_mini\_jeu = nb\_medailles\_argent + nb\_medailles\_or \* 3**

Les scores des **quatre** mini-jeux sont **multipliés ensemble** pour déterminer le **score global**.

⚠ Cela signifie donc que n'avoir que des médailles de bronze dans un mini-jeu vous donnera un score final de 0 !

Pendant un tour de réinitialisation, le registre **GPU** affichera **"GAME\_OVER"**.

En cas d'égalité dans un mini-jeu, les joueurs ex æquo remporteront la même médaille la plus haute. Par exemple, si deux joueurs sont à égalité pour la première place, ils recevront tous les deux une médaille d'or et le troisième joueur recevra une médaille **de bronze**.

## Mini-jeu 1 : Course de Haies

Ce mini-jeu est une course entre les trois agents. Chaque agent est sur la même piste de course générée aléatoirement.

La piste de course est composée de **30 cases**, les agents commencent sur la première case, et la dernière case est la ligne d'arrivée. Une case peut contenir une **haie** que les agents doivent **sauter** sinon ils **heurteront** celle-ci et seront **étourdis** pour les **3** prochains tours. Un agent étourdi ne bougera pas quelle que soit l'action effectuée.

À chaque tour, les agents peuvent effectuer l'une des actions suivantes :

- **UP** : sauter par-dessus une case, en ignorant toute haie sur la case suivante et en avançant de **2** cases au total.
- **LEFT** : avancer d'**1** case.
- **DOWN** : avancer de **2** cases.
- **RIGHT** : avancer de **3** cases.

Heurter une haie interrompra le mouvement de l'agent, l'arrêtant sur la même case que la haie.

Lorsque l'un des agents atteint la **ligne d'arrivée**, la manche se termine. Les joueurs reçoivent une médaille en fonction de la position dans la course de leur agent, et la manche suivante commence après un tour de **réinitialisation**.

Registre	Description	Exemple
GPU	Représentation ASCII de la piste de course. . pour un espace vide. # pour une haie.	....#...#.....
reg0	position de l'agent du joueur 1	0
reg1	position de l'agent du joueur 2	6
reg2	position de l'agent du joueur 3	12
reg3	décompte d'étourdissement pour le joueur 1	1
reg4	décompte d'étourdissement pour le joueur 2	0
reg5	décompte d'étourdissement pour le joueur 3	2
reg6	non utilisé	

Le **décompte d'étourdissement** est le nombre de tours restants en étant étourdi (3 , puis 2 , puis 1). 0 signifie que l'agent n'est pas étourdi.

### Mini-jeu 2 : Tir à l'arc

Chaque joueur contrôle un curseur avec une coordonnée x et une coordonnée y. À chaque tour, les joueurs choisissent une direction, puis le curseur se déplace dans cette direction selon la force actuelle du **vent**. Après 10 à 15 tours, les joueurs remportent des médailles en fonction de leur proximité avec la coordonnée (0,0) en distance euclidienne.

Les coordonnées x et y sont restreintes dans [-20;20].

Registre	Description	Exemple
GPU	Une série d'entiers, indiquant la puissance du vent pour les tours à venir. L'entier à l'index 0 est la force du vent actuelle.	9914113315261
reg0	Coordonnée x pour le joueur 1	0
reg1	Coordonnée y pour le joueur 1	-10
reg2	Coordonnée x pour le joueur 2	5
reg3	Coordonnée y pour le joueur 2	8
reg4	Coordonnée x pour le joueur 3	-2
reg5	Coordonnée y pour le joueur 3	20
reg6	inutilisé	

### Mini-jeu 3 : Roller de vitesse

Les joueurs courent sur une piste cyclique longue de 10 cases. Chaque joueur aura un attribut de **risk** allant de 0 à 5.

À chaque tour, une liste des 4 actions sera fournie dans un ordre aléatoire dans le GPU, par exemple **ULDR** (pour **UP**, **LEFT**, **DOWN**, **RIGHT**), cela s'appelle l'**ordre de risque**. Effectuer l'action à un indice plus élevé déplacera le joueur vers l'avant sur plus de cases. Mais choisir le mouvement le plus rapide n'est pas sans risque...



- L'action à l'index **0** déplacera votre joueur d'une case et **diminuera** votre **risk** de **1**
- L'action à l'index **1** déplacera votre joueur de **2** cases
- L'action à l'index **2** déplacera votre joueur de **2** cases mais augmentera votre **risk** de **1**
- L'action à l'index **3** déplacera votre joueur de **3** cases mais augmentera votre **risk** de **2**

De plus, si après un mouvement un joueur se retrouve sur le même espace qu'un adversaire, leur **risk** à tous les deux est augmenté de **2** ! Si le risque d'un joueur atteint **5** ou plus, le joueur est **étourdi** pour les **2** prochains tours et son **risk** est réinitialisé à **0**.

Registre	Description	Exemple
GPU	Ordre de risque pour ce tour	<b>URLD</b>
reg0	cases parcourues par le joueur 1	2
reg1	cases parcourues par le joueur 2	9
reg2	cases parcourues par le joueur 3	21
reg3	risque du joueur 1 ou décompte d'étourdissement en nombre négatif s'il est étourdi	4
reg4	risque du joueur 2 ou décompte d'étourdissement en nombre négatif s'il est étourdi	-1
reg5	risque du joueur 3 ou décompte d'étourdissement en nombre négatif s'il est étourdi	0
reg6	tours restants	14

Vous pouvez déterminer si deux joueurs partagent une case en comparant leurs **cases parcourues modulo 10**

### Mini-jeu 4 : Plongée artistique

Les joueurs doivent réaliser une série de figures correspondante à la séquence de directions donnée au début de chaque partie. On appelle cette série l'**objectif**.

À chaque tour où l'action d'un joueur correspond à la direction de l'objectif de ce tour, le joueur augmentera son **multiplicateur de combo** puis gagnera des points égaux à celui-ci. Le multiplicateur de combo commence à **0** et augmente de **1** pour chaque tour consécutif où l'action du joueur correspond à l'objectif. Il **se réinitialise** à **0** lorsque l'action du joueur ne correspond pas à l'objectif.

Registre	Description	Exemple
GPU	Objectif de plongée de cette partie	<b>UUUDDLLLULDRLL</b>
reg0	points du joueur 1	7
reg1	points du joueur 2	4
reg2	points du joueur 3	0
reg3	combo du joueur 1	1
reg4	combo du joueur 2	0
reg5	combo du joueur 3	9
reg6	<i>inutilisé</i>	



#### Condition de victoire

Vous avez un **score global** plus élevé après **100** tours.

#### Condition de défaite



Votre programme ne fournit pas une commande dans le temps imparti ou fournit une commande non reconnue.



## Conseils de débogage

- Appuyez sur l'icône d'engrenage sur le visualiseur pour accéder à des options d'affichage supplémentaires.
- Utilisez le clavier pour contrôler l'action : espace pour jouer/pause, flèches pour avancer d'une trame à la fois.

## ■ Protocole de Jeu

### Entrées d'Initialisation

**Première ligne :** `playerIdx` un entier pour indiquer quel agent vous contrôlez dans les mini-jeux.

**Ligne suivante :** le nombre de mini-jeux fonctionnant simultanément. Pour cette ligue, c'est `4`.

### Entrées pour Un Tour de Jeu

**Les 3 premières lignes :** une ligne par joueur, ordonnée par `playerIdx`. Une chaîne de caractères `scoreInfo` contenant une répartition du score global de chaque joueur. Elle contient `13` entiers. Le premier entier représentant le **score global** actuel du joueur suivi par trois entiers par mini-jeu : `nb_medailles_or`, `nb_medailles_argent`, `nb_medailles_bronze`.

**Les `nbGames` lignes suivantes :** une ligne pour chaque mini-jeu, contenant les huit registres séparés par des espaces :

- `gpu` une chaîne de caractères
- `reg0` un entier
- `reg1` un entier
- `reg2` un entier
- `reg3` un entier
- `reg4` un entier
- `reg5` un entier
- `reg6` un entier

Leurs valeurs dépendront du jeu. Les registres non utilisés seront toujours `-1`.

### Sortie

L'une des chaînes suivantes :

- `UP`
- `RIGHT`
- `DOWN`
- `LEFT`

L'effet dépendra du jeu.

### Contraintes

$0 \leq \text{playerIdx} \leq 2$

$1 \leq$

nbGames  $\leq 4$  (dans toutes les ligues)

Temps de réponse par tour  $\leq 50$  ms

Temps de réponse pour le premier tour  $\leq 1000$  ms

### Code source

Vous pouvez voir le code source de ce jeu sur [ce repo GitHub](#).

