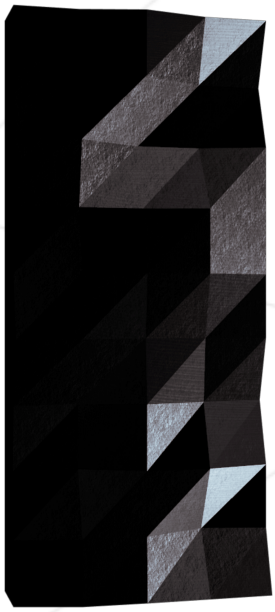


Domino

Rush Algo

Ly ly@42.fr
42 Staff pedago@42.fr

Résumé: Ce document est le sujet du rush Domino.

Table des matières

I Préambule **2**

II Introduction **3**

III Objectifs **4**

IV Consignes générales **5**

V Partie obligatoire **6**

V.1 Les règles 6

V.2 La VM 6

V.3 Exemple 7

VI Partie bonus **9**

VII Rendu et peer-évaluation **10**

Chapitre I

Préambule

Un crocodile s'en allait à la guerre
Disait adieu à ses petits enfants
Traînant la queue la queue
Dans la poussière
Il s'en allait combattre les éléphants

Ah les crocrocros les crocrocros les crocodiles
Sur les bords du Nil ils sont partis n'en parlons plus
Ah les crocrocros les crocrocros les crocodiles
Sur les bords du Nil ils sont partis n'en parlons plus

Il fredonnait une marche militaire
Dont il mâchait les mots à grosses dents
Quand il ouvrait la gueule tout entière
On croyait voir ses ennemis dedans

Ah les crocrocros les crocrocros les crocodiles
Sur les bords du Nil ils sont partis n'en parlons plus
Ah les crocrocros les crocrocros les crocodiles
Sur les bords du Nil ils sont partis n'en parlons plus

Un éléphant parut et sur la terre
Se prépara un combat de géants
Mais près de là courait une rivière
Le crocodile s'y jeta subitement

Ah les crocrocros les crocrocros les crocodiles
Sur les bords du Nil ils sont partis n'en parlons plus
Ah les crocrocros les crocrocros les crocodiles
Sur les bords du Nil ils sont partis n'en parlons plus

Chapitre II

Introduction

Ce rush consiste en la réalisation d'une intelligence artificielle capable de jouer aux dominos.
Elle peut faire le café si vous le souhaitez, mais je ne bois pas de café, donc contentez-vous des dominos.

Chapitre III

Objectifs

L'objectif de ce projet est de faire en sorte que vos IA s'étripent à mort comme Fred peut le faire avec sa grand-mère les dimanches d'hiver.
Oui, vous l'avez compris, il y aura compétition entre vos IAs.
Ce rush vous apprendra à faire preuve de rigueur (le domino ne plaisante pas) et vous permettra de développer vos skills en algorithmie de pointe.

Chapitre IV

Consignes générales

Les consignes suivantes feront toutes parti du barème de soutenance. Soyez très attentifs et soigneux dans leur application car elles sont sanctionnées par un 0 sans appel.

- Au cas ou quelqu'un en douterait, ce rush est bien entendu à écrire en `C`.
- Ce projet ne sera corrigé que par des humains.
- Votre projet doit être à la Norme. Une erreur de Norme est éliminatoire, y compris dans votre `libft`.
- Vous devez gérer les erreurs de façon sensible. En aucun cas votre programme ne doit quitter de façon inattendue (Segmentation fault, bus error, double free, affichage de non sens, etc) ou partir en boucle infinie, que ce soit dans la partie obligatoire ou dans la partie bonus. Une telle erreur est éliminatoire. Non, vous n'avez pas besoin de `Valgrind` pour cela.
- Toute mémoire allouée sur le tas doit être libérée proprement. Oublier de désallouer de la mémoire est éliminatoire.
- La valeur de retour de tous vos appels système doit être vérifiée. Un oubli est éliminatoire. On tolérera de ne pas verifier la valeur de retour de `write(2)`. On définit par "appel système" toute fonction dont le man se trouve dans la section 2 sur les Macs de l'école.
- Vous pouvez utiliser votre bibliothèque `libft`. Pour cela, vous devez en rendre les sources à la racine de votre dépot dans un dossier nommé `libft`. Votre bibliothèque devra être compilée en même temps que votre rendu et linkée avec celui-çi. Votre `libft` doit être à la Norme. Utiliser votre `libft` pour contourner la Norme est éliminatoire.

Chapitre V

Partie obligatoire

V.1 Les règles

Pour ce rush, nous allons implémenter des règles particulières du domino : celles du 5 partout (revues par mes soins).

- Pour commencer à jouer, vous devez poser un double.
- A votre tour, trois possibilités s'offrent à vous :

  1. Poser un domino
  2. Bouder
  3. Piocher

- A la fin de votre tour, si la somme des extrêmités libres des dominos sont égales à un multiple de 5, ces points sont ajoutés à votre score.
- Le premier joueur ayant posé toutes ses pièces gagne les points de chacun des dominos restants dans la main de chaque joueur.
- Le joueur ayant totalisé le plus de points remporte la partie.
- La partie se déroule sur une base de 6 (dominos de 0 :0 à 6 :6)

V.2 La VM

Pour simuler ce combat acharné, une VM vous est fournie.
Vous devez lui passer en paramètre deux programmes.
Elle se lance comme suit :

```
$./dominoVM "./monIAdefou" "./IAquivamepoutrer"
```

Vous communiquerez avec cette VM par le biais du stdin/stdout.

Vous trouverez le protocole de communication à suivre en ressources du sujet.

La VM comprend aussi un certain nombre de message d'erreur commencant par "ko". Si vous obtenez un message "ko", la VM s'arrêtera et vous jugera responsable de tous ses maux. Vous serez donc le perdant.

`Votre IA devra bien attendre d'avoir reçu "go" avant de commencer à parler, autrement, vous aurez 0.`

Vous pouvez aussi utiliser la VM en mode training à l'aide de cette commande :

```
$./dominoVM --training
```

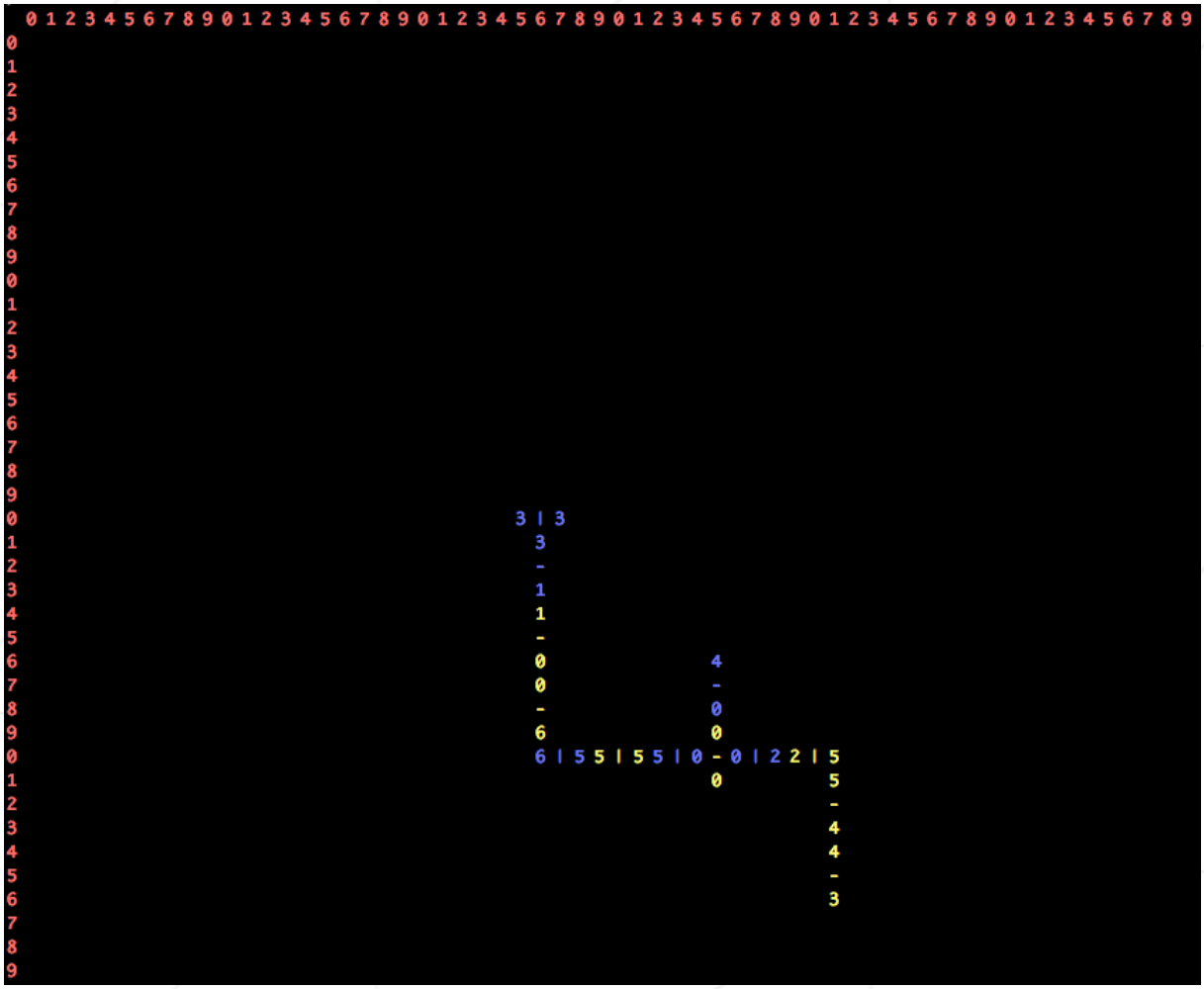
Cela vous permettra de vous familiariser avec son utilisation. En plus du reste, elle vous enverra les pièces disponibles dans votre stack à chaque tour. Vous faites les deux joueurs mais vous aurez un code couleur différent pour chaque joueur.

Vous pouvez entrer une taille de plateau pour le training, à l'aide de l'argument "–size :c :l".

```
$./dominoVM --training --size:30:25
```

V.3 Exemple

Voici un exemple de partie pour que vous sachiez bien comment positionner vos pièces, et surtout vos "coudes". Pas sur la table, dirait ma maman (so much fun here !).
Une extrêmité est un bout de domino qui n'est pas en contact avec un autre bout de domino.


3 | 3
3
-
1
1
-
0
0
-
6
4
-
0
0
6 | 5 5 | 5 5 | 0 - 0 | 2 2 | 5
0
5
-
4
4
-
3

Chapitre VI

Partie bonus

Le bonus portera sur la qualité des explications sur votre algo et sur son implémentation.

Chapitre VII

Rendu et peer-évaluation

Vous rendrez sur votre dépôt git tout fichier nécessaire à votre évalutation par vos pairs.