

BUT1 – SAÈ S1.02

COMPARAISON D'APPROCHES ALGORITHMIQUES

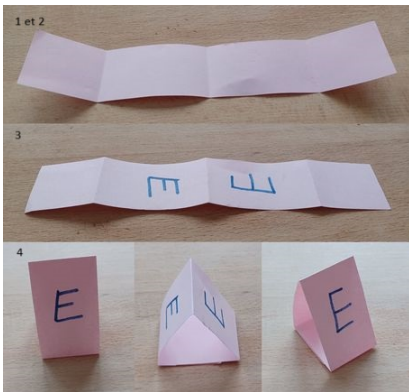
– OCTO–VERSO –

Le but du projet est de développer un logiciel permettant à deux joueurs de disputer une partie de OCTO–VERSO. L'application doit veiller au respect des règles du jeu et gérer la totalité du déroulement de la partie jusqu'à l'annonce du gagnant.

OCTO–VERSO est un jeu de lettres créé par Laurence Alsac en 2010 et est édité par la société "Envie de jouer".

1 Règles du jeu

Le matériel de jeu est constitué d'un paquet de 88 lettres de l'alphabet. Chaque lettre est portée par un chevalet où elle est imprimée des deux cotés. Le plus simple pour imaginer les chevalets est d'en construire un soit même :



1. Découper une bande de papier étroite.
2. La plier en son milieu dans le sens de la longueur et rabattre les deux extrémités vers l'intérieur au deux tiers.
3. Le chevalet va être obtenu en accolant les deux extrémités, écrire la même lettre sur les deux faces externes.
4. Accoler les deux extrémités en les scotchant.

Les lettres portées par les 88 chevalets se répartissent comme suit :

lettre	qté	lettre	qté	lettre	qté	lettre	qté	lettre	qté	lettre	qté
A	9	E	14	I	7	N	6	R	6	V	2
B	1	F	1	J	1	O	5	S	7		
C	2	G	1	L	5	P	2	T	6		
D	3	H	1	M	3	Q	1	U	5		

Table 1: Nombre de chevalets pour chaque lettre de l'alphabet

Nous pouvons remarquer que les lettres les plus rares du dictionnaire français (**K, X, Y, Z et W**) ont été écartées.

Le matériel de jeu contient aussi un rail rotatif sur lequel 8 chevalets peuvent être alignés côte à côte pour former des mots.

Initialement, **12 chevalets tirés au hasard sont remis à chacun des deux joueurs et les chevalets restant forment la pioche.** Pour commencer et remplir le rail, les joueurs forment chacun un mot de 4 lettres à partir des 12 qu'ils détiennent et **les deux mots sont posés côte à côte sur le rail dans l'ordre alphabétique.** Par exemple, si le premier joueur propose CIEL et que le second joueur propose BLEU, le rail contiendra les lettres BLEUCIEL.

Le but du jeu est d'être le premier joueur à se débarrasser de tous ses chevalets.

Les joueurs jouent à tour de rôle et le joueur ayant proposé le plus petit mot de 4 lettres commence. Lorsque c'est son tour, le joueur doit former un nouveau mot à partir de ses chevalets en les glissant à une extrémité du rail. L'ajout de ces nouveaux chevalets va provoquer l'expulsion d'un nombre de chevalets équivalents à l'autre extrémité. **Les chevalets expulsés sont récupérés par le joueur adverse.**

Comment former un mot ? Les lettres doivent être composées à partir des chevalets du joueur et d'au moins deux chevalets de l'extrémité du rail. Par exemple, si le rail contient BLEUCIEL et que le joueur détient les chevalets nécessaires, il peut introduire ANCER à droite pour former (EL)ANCER ou introduire ENSEM à gauche pour former ENSEM(BLE).

Si le joueur décide de former le mot ELANCER, il introduit 4 de ses chevalets (ANCER) à droite et en emploie 2 à cette extrémité du rail (EL). Les 4 chevalets se trouvant à l'autre extrémité du rail (BLEU) sont expulsés sur la gauche et le rail contient alors la suite CIELANCER. Le joueur courant s'est débarrassé de 4 de ses chevalets et le joueur adverse a récupéré les 4 expulsés à gauche du rail.

L'originalité du jeu vient du fait que le rail est rotatif. Si le rail contient BLEUCIEL et qu'il est retourné, les lettres qui le composent sont lues à l'envers et forment la suite LEICUELB.

Ce côté du rail peut lui aussi être employé par le joueur pour y poser un mot. Il peut ainsi introduire EUFS à droite pour former (ELB)EUF (il est très fort et il sait que les elbeufs sont des draps fins fabriqués dans la ville d'Elbeuf). Il peut aussi introduire FACI à gauche pour former FACI(LE) (mais là c'est beaucoup plus simple).

Attention, un même mot ne peut être joué qu'une seule fois au cours d'une même partie. De même, seul les mots présents dans le dictionnaire sont autorisés.

Un joueur qui réussit à former un mot de 8 lettres (remplissant donc tout le rail) est récompensé. Il défausse un de ses chevaux qu'il choisit et qui rejoint la pioche. Dans l'exemple ci-dessus, c'est le cas lorsque le joueur forme le mot ENSEMBLE.

Une autre originalité du jeu est qu'il est ouvert. Mis à part les chevaux de la pioche, les deux joueurs voient les deux côtés du rail, les lettres portées par leurs chevaux mais aussi celles de l'adversaire. Ainsi si un joueur vient de former un mot composé de moins de 8 lettres alors que ses chevaux lui permettaient de former un de 8, son adversaire peut l'annoncer une fois le mot posé. Le coup du joueur est laissé en l'état mais, en récompense, son adversaire se défausse d'un de ses chevaux. Le mot de 8 lettres proposé par le joueur adverse ne pourra plus être joué dans cette partie.

Une dernière règle permet d'éviter les situations de blocage. Lorsque c'est son tour, à la place de former un mot, le joueur peut échanger un de ses chevaux par un autre pris au hasard dans la pioche.

2 Cahier des charges

L'application que vous devez développer doit permettre à deux joueurs de réaliser une partie dans sa totalité. Les joueurs sont désignés par leur numéro d'ordre (1 et 2).

Dans la suite du document, le texte en bleu est affiché par le programme et celui en rouge est saisi par l'un des joueurs.

Initialement, votre programme doit gérer le mélange des chevaux et la distribution des chevaux aux joueurs. Les lettres portées par les chevaux détenus par chaque joueur doivent ensuite être affichées classées par ordre alphabétique comme suit :

1 : ABCDEFGHIJLM
2 : AACDNOPQRSTU

Chaque joueur est ensuite invité à saisir un mot de 4 lettres.

1> AIDE
2> PURS

Si le mot saisi n'est pas valide (sa longueur n'est pas de 4 lettres, il n'est pas dans le dictionnaire ou les lettres détenues par le joueur ne sont pas suffisantes), votre programme redemande immédiatement la saisie d'un mot au même joueur sans autre forme d'explication.

1> AIED
1> ABIME
1> AGES
1> AIDE
2> PURS

À chaque étape du jeu, le programme doit à présent afficher la situation courante du jeu, c'est à dire, les lettres (classées par ordre alphabétique) étant portées par les chevaux détenus par chacun des joueurs ainsi que les lettres portées par les chevaux présents sur le rail.

1 : BCFGHJLM
2 : AACDNOQT
R : AIDEPURS
V : SRUPEDIA

Les lettres sur le rail sont présentées dans les 2 ordres possibles. Les lettres R (pour recto) et V (pour verso) vont permettre de désigner le point d'insertion du mot.

Le joueur devant jouer est alors invité à saisir son coup par l'affichage de 1> ou 2> selon le joueur courant.

Le joueur peut décider entre (1) former un mot, (2) échanger un de ses chevaux avec un autre de la pioche.

(1) Un nouveau mot est formé en indiquant le côté du rail où il doit être posé par la lettre correspondante (R ou V) suivi du mot. Les lettres prises sur le cheval doivent nécessairement être mise entre parenthèse désignant ainsi à quelle extrémité les autres chevaux doivent être introduits. Par exemple, dans la situation précédente, le joueur 1 qui est le premier à jouer peut saisir R CH(AI) ou V (DIA)L.

Là encore, si le coup saisi n'est pas valide (le mot est trop long pour tenir sur le rail, il n'appartient pas au dictionnaire ou a déjà été joué précédemment, les lettres prises à cette extrémité du rail ne correspondent pas ou sont en nombre insuffisant, les nouvelles lettres introduites ne sont pas couvertes par les chevalets du joueur), un nouveau coup est demandé au joueur toujours sans autre forme d'explication.

```
1> R CH(AT)
1> V (A)H
1> R CH(AI)
```

(2) L'échange d'un de ses chevalets est provoqué en saisissant le signe - suivi de la lettre correspondante. Là encore, un nouveau coup est demandé en cas d'erreur (la lettre n'est pas présente sur les chevalets du joueur).

```
1> - Q
1> - J
```

Le cas échéant, le joueur peut signaler un mot de 8 lettres qu'aurait pu faire l'adversaire au coup précédent. Le signalement est réalisé en saisissant un coup normal mais où le côté du rail concerné est précisé par une lettre minuscule (**r** ou **v**).

Un tel signalement n'est possible que si le joueur adverse a bien formé un mot de moins de 8 lettres au coup précédant, que le joueur courant n'a pas déjà fait un signalement pour ce coup et que le mot de 8 lettres proposé aurait vraiment pu être joué. Attention, ce mot fait référence à l'état du rail et des chevalets de l'adversaire avant que celui-ci ait formé son dernier mot.

Si un joueur a formé un mot de 8 lettres ou qu'il en a signalé un qu'aurait pu jouer son adversaire, il lui est demandé de choisir quel chevalet il veut écarter. L'invitation est précédée d'un tiret (-1> ou -2>) et le joueur doit saisir une lettre. Si celle-ci n'est portée par aucun de ses chevalets, il est invité à recommencer.

Une trace d'exécution d'une partie complète est donnée en annexe. Notez que cette trace correspond à une partie ayant été filmée et commentée (le lien de la vidéo est dans l'annexe).

Une commande spéciale permet aux joueurs en perdition d'obtenir un peu (beaucoup !) d'aide. Si, alors qu'il doit former un mot, le joueur saisit la lettre **H**, alors tous les meilleurs coups possibles sont affichés. Un coup est considéré comme meilleur qu'un autre s'il permet au joueur de se débarrasser de plus de lettres. Les coups doivent être affichés dans le format de saisie (), un par ligne et classés selon l'ordre alphabétique des mots joués.

Cette commande spéciale est optionnelle et l'avoir programmé donnera lieu à un bonus.

Vous disposez d'un dictionnaire de la langue française. Il est composé de 369085 mots. Votre application devra faire l'hypothèse que le fichier texte correspondant se trouve dans le répertoire courant où est lancé l'application (surtout ne pas mettre de chemin absolu pour désigner le fichier).

Les coups sont saisis sur une seule ligne de texte. Les espaces servent de séparateur et peuvent être répétés. Toute commande de format incorrect doit être ignorée et cela indépendamment de la cause de l'erreur.

À la fin de la partie (*i.e.* un joueur n'a plus de chevalet en sa possession), le programme se termine automatiquement sans affichage particulier (on sait que le dernier joueur ayant joué a gagné).

3 Qui, quoi et quand?

Votre projet doit être fait en binôme. Les groupes de 3 ne seront pas acceptés. Évitez de faire votre projet tout seul (soit vous êtes très fort et des personnes ont besoin de votre aide, soit vous avez des difficultés et il faut vous faire aider).

Avant le **vendredi 10 janvier 2025 à 23h59**, vous devez déposer sur MOODLE une archive (.zip) contenant votre rapport (.pdf) ainsi que tous les fichiers sources (.cpp et .h) de votre application. Ces derniers doivent être réunis, au sein de l'archive, dans un dossier nommé **src**.

Vous devez porter une attention particulière à la rédaction de votre dossier. Sa qualité est déterminante pour l'évaluation de votre travail. Votre dossier doit être un unique document pdf dont la composition est la suivante :

- Une page de garde indiquant le nom et le **groupe** des membres du binôme, l'objet du dossier.
- Une table des matières de l'ensemble du dossier.
- Une brève introduction du projet.
- Le graphe de dépendance des fichiers sources de vos applications (*cf.* cours n°8). Tous les composants (qu'ils soient réutilisés ou développés) de vos applications devront figurer sur le graphe.
- Le code source des tests unitaires que vous aurez écrits (précisez quels tests passent et lesquels échouent).
- Un bilan du projet (les difficultés rencontrées, ce qui est réussi, ce qui peut être amélioré).
- En annexe, le code complet de vos sources (pensez à présenter les fichiers selon un ordre logique).

Nous vous rappelons que le critère principal de notation est la structuration de votre code. Votre rapport doit mettre en avant la qualité de celle-ci.

Tous les éléments (constantes, types et fonctions) déclarés au sein d'un fichier `.h` doivent être documentés.

Vous pouvez employer les structures de données vues en cours ou développer les vôtres. Vous pouvez vous servir de toute la bibliothèque standard du C.

La correction de votre code sera testé automatiquement par nos soins. Votre programme doit respecter strictement les formats d'affichage et de saisie présentés ci-dessus sous peine d'échouer lors des tests.

Annexe

<pre> 1 : ACEEELMNRUV 2 : ABDEILNORRTU 1> LAVE 2> BORD 1 : CEEMNRUV 2 : AEILNRTU R : BORDLAVE V : EVALDROB 2> V (ROB)INET 1 : ACEEELMNRUV 2 : ALRU R : TENIBORD V : DROBINET 1> R (BORD)URES 1 : ACEELMN 2 : AEILNRTU R : BORDURES V : SERUDROB </pre>	<pre> -1> V 1 : ACEELMN 2 : AEILNRTU R : BORDURES V : SERUDROB 2> V REALI(SER) 1 : ABCDEELMNORU 2 : NTU R : RESILAER V : REALISER -2> U 1 : ABCDEELMNORU 2 : NT R : RESILAER V : REALISER 1> R (ER)ABLE </pre>	<pre> 1 : CDEMNRU 2 : EINRST R : LAERABLE V : ELBAREAL 2> r RECOUD(RE) -2> N 1 : CDEMNRU 2 : EIRST R : LAERABLE V : ELBAREAL 2> V (REAL)ISTE 1 : ABCDEELMNORU 2 : R R : ETSILAER V : REALISTE -2> R </pre>
--	--	--

Figure 1: Exemple de partie – en rouge, les données saisies par les joueurs, en bleu, les messages affichés par le programme. La vidéo commentée de cette partie est accessible ici : https://www.youtube.com/watch?v=A4F2e_z1U0c.

