

# S U J E T

## BASES DE LA PROGRAMMATION ORIENTÉE OBJET

### – CRAZY CIRCUS –

## 1 Présentation du jeu

Le jeu a pour but de déterminer quel est le meilleur dompteur parmi les joueurs. Il s'agit de réussir un exercice de domptage mettant en œuvre trois animaux : un lion, un ours blanc et un éléphant. Ces animaux sont répartis sur deux podiums – l'un est bleu, l'autre est rouge. Lorsque deux ou trois animaux sont sur le même podium, ils forment une pile (au sens commun du terme).

Les animaux réagissent uniquement à 5 ordres distincts :

- **KI** – L'animal se trouvant en haut de la pile du podium bleu saute pour rejoindre le sommet de la pile du podium rouge.
- **LO** – L'animal se trouvant en haut de la pile du podium rouge saute pour rejoindre le sommet de la pile du podium bleu.
- **SO** – Les deux animaux se trouvant au sommet des piles des deux podiums échangent leur place.
- **NI** – L'animal se trouvant en bas de la pile du podium bleu monte et se place en haut de la pile de ce même podium.
- **MA** – L'animal se trouvant en bas de la pile du podium rouge monte et se place en haut de la pile de ce même podium.

Le but du jeu est de trouver le plus rapidement la bonne séquence d'ordres qui, réalisés par les animaux, les conduiront d'une situation de départ à une situation à atteindre. Le joueur ayant trouvé le premier une bonne séquence remporte le tour. Un joueur ayant donné une mauvaise séquence ne peut plus en proposer d'autre pour ce tour. Si, durant un tour, il n'y a plus qu'un joueur à ne pas avoir proposé de séquence, celui-ci remporte le tour.

La situation initiale de début du jeu et la situation à atteindre à chaque tour sont déterminées en tirant au hasard l'une des 24 cartes représentant chaque situation possible. La situation atteinte après un tour est la situation de départ du tour qui suit. Une carte une fois tirée n'est pas remise en jeu et la partie se termine lorsque les cartes sont épuisées. Le joueur ayant remporté le plus de tours gagne la partie.

## 2 Travail à faire

L'application à développer doit permettre à des joueurs de s'affronter. Le programme devra gérer une partie dans sa totalité.

Les identités des joueurs (leurs prénoms ou surnom par exemple) doivent être reçus sur la ligne de commande au lancement du programme. Le programme doit afficher la situation de jeu sous la forme reproduite ci-dessous.

La situation de départ est donnée en haut à gauche (ici le lion est sous l'ours sur le podium bleu alors que l'éléphant est seul sur le podium rouge) et la situation à atteindre est indiquée à droite (ici le lion est sur l'éléphant sur le podium bleu alors que l'ours est seul sur le podium rouge). Les différents ordres possibles sont rappelés en bas (les sauts d'un sommet à l'autre sont représentés par des flèches et un déplacement du bas vers le haut est symbolisé par le caractère (^)). Les séquences KIMALONI et SONI sont deux solutions pour cette situation de jeu.

OURS			LION	
LION	ELEPHANT		ELEPHANT	OURS
----	----	==>	----	----
BLEU	ROUGE		BLEU	ROUGE

-----

KI : BLEU --> ROUGE	NI : BLEU ^
LO : BLEU <-- ROUGE	MA : ROUGE ^
SO : BLEU <-> ROUGE	

Lorsqu'un joueur veut jouer, il saisit son identité suivie de la séquence d'ordres qu'il propose. Si DP est une identité connue, le joueur peut saisir 'DP KIMALONI' par exemple.

Si l'identité n'est pas connue, la séquence doit être ignorée et un message d'erreur affiché. Si le joueur n'avait pas le droit de jouer (*i.e.* il a déjà fait une erreur) ou que la séquence n'est pas la bonne, un message informatif doit être affiché par le programme. Si la séquence est correcte, le joueur remporte un point, une nouvelle situation de jeu est choisie aléatoirement par le programme parmi celles qui n'ont pas encore été jouées et un nouveau tour peut débuter.

En fin de partie, le score et le rang des joueurs sont affichés (par score décroissant et par ordre alphabétique en cas d'égalité).