

# Rapport TP4

## But :

L'objectif de ce tp est d'implémenter le Chomp, un jeu combinatoire abstrait à deux joueurs et nous allons utiliser les notions suivantes : tableau statique à deux dimensions, les types structurés et énumérés, interface *ncurses* en gestion tour par tour, comparaison entre programmation par fichier unique et par modules et la compilation à partir d'un fichier *bash*.

## Description du projet :

Le projet se découpe en 6 modules :

- Un module Chomp qui va servir de fonction principale du jeu et qui va appeler tous les autres modules nécessaires pour le programme.
- Un module Graphique qui va afficher le plateau du jeu et transférer des données récupéré sur la fenêtre.
- Un module Coup qui va créer un nouveau type structuré qui stockera les coordonnées x et y d'un coup de l'un des joueurs.
- Un module Joueur qui va créer un nouveau type énuméré qui contiendra deux valeurs qui représentent le joueur 1 et le joueur 2.
- Un module Tablette qui va créer un nouveau type structuré qui va stocker une tableau statique à deux dimension qui représentera le plateau de jeu.
- Un module Position qui va créer un nouveau type structuré qui va stocker le plateau de jeu Tablette et la valeur du joueur Joueur en cours.

Prototypes	Descriptions	Renvois	Dépendances
Joueur partie(int num_partie, int nb_partie);	Lance la partie du jeu et l'affiche. Elle récupère son le numéro de la partie dans 'num_partie' et le nombre total de partie dans 'nb_partie'	Renvoie la valeur du joueur qui a gagné la partie en cours	<i>ncurses.h</i> <i>Position.h</i> <i>Graphique.h</i> <i>Coup.h</i> <i>Joueur.h</i>
void match(void);	Lance le match et demande aux joueurs le nombre de partie qu'ils souhaitent	Ne renvoie rien	<i>ncurses.h</i> <i>Chomp.h</i>
int est_legal(Position pos, Coup coup);	Vérifie si le Coup 'coup' est valide dans la Position 'pos'	Renvoie 1 si le coup est valide et 0 sinon	<i>Position.h</i> <i>Coup.h</i>

void afficher_position(Position pos);	Affiche la Position 'pos' dans la fenêtre standard	Ne renvoie rien	<i>Graphique.h Position.h ncurses.h</i>
Coup lire_coup(Position pos);	Lit un clic souris et l'associe à un Coup et vérifie si le coup est valide dans la Position 'pos'	Renvoie le coup valide récupéré	<i>Coup.h ncurses.h Position.h</i>
Joueur adversaire(Joueur joueur);	Donne l'adversaire du Joueur 'joueur'	Renvoie la valeur du joueur inverse de 'joueur'	<i>Joueur.h</i>
Tablette creer_tablette(void);	Crée une Tablette initialisée	Renvoie une variable de type Tablette avec le champs chocolat où toutes ses cases sont initialisées à 1	<i>Tablette.h</i>

On peut compiler le programme à l'aide du fichier *bash* Compilation.sh en exécutant la commande :  
sh Compilation.sh. L'exécutable s'appelle tp4 et s'exécute comme-suit : ./tp4.

Les avantages de la programmation modulaire sont : le partage du travail qui peut s'organiser plus facilement, rend plus compréhensible un projet car il le partitionne presque par fonctionnalité et d'autre.

Son désavantage le plus important est sa longueur pour des codes simples il demande parfois trop d'effort pour des choses qui n'en ont possiblement pas la nécessités.