
Projet de Structures Linéaires : Rapport final

Jeu : « TOTEM ! »

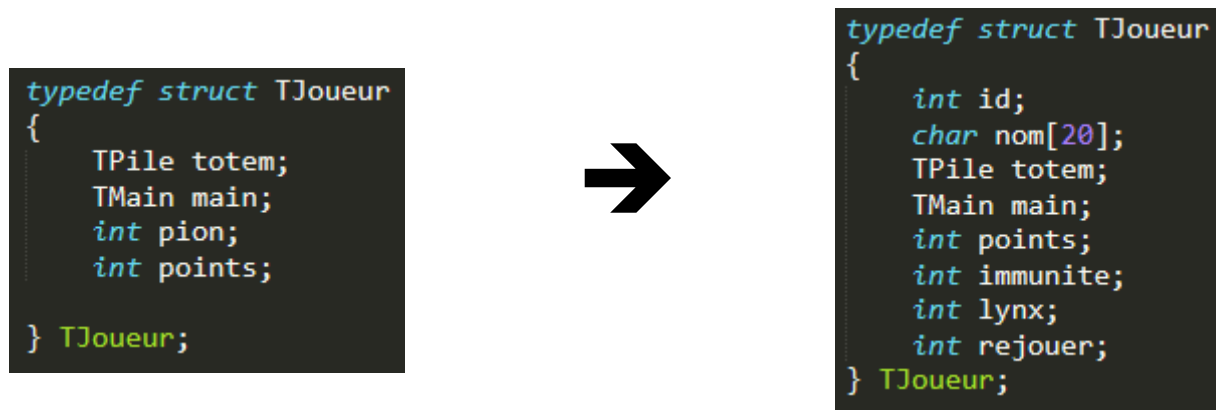
Etudiants : Killian BRONDY / Ludwig GRANDIN / Luca GARIC

Sommaire

I) Modifications des choix de représentation de données	3
II) Planning définitif	3
III) Jeux de tests	5

I) Modifications des choix de représentation de données

Les changements notables que nous avons effectués concernent principalement notre type TJoueur :



Tout d'abord, l'attribut **pion** a été renommé en « **id** » pour plus de visibilité. De plus, nous avons choisi d'ajouter un attribut nom au joueur pour rendre le programme plus convivial (lors de nos affichages qui annoncent le tour d'un joueur notamment).

Plusieurs données techniques sont apparues :

- **Immune** : Comme le précise le nom de cet attribut, le joueur peut bénéficier d'une immunité au vol de totem ou à la destruction de son totem lorsque le sommet de ce dernier correspond respectivement à la tête d'aigle (la variable prendra la valeur « 2 » signifiant la protection contre le vol de sommet) et à la tête de tortue (la variable prendra la valeur « 1 » signifiant la protection contre la destruction de sommet).
- **Lynx** : Cette variable fonctionne comme un mode activable lorsque le sommet du totem du joueur est une tête de lynx. En effet, cette variable prend seulement la valeur 1 ou 0. A 0, l'action du joueur sur la pioche sera normale, or lorsque cette variable passe à 1, le joueur sera en mesure de piocher trois cartes et en défausser deux.
- **Rejouer** : C'est aussi une donnée spéciale qui, comme la variable précédente, ne prend que deux valeurs (0 ou 1). A 1, le joueur joue une nouvelle fois. Cette variable est activable en contrant une carte « faux pas » avec une autre carte « faux pas ».

Comme changement mineur, nous avons renommé notre type TCellule en type TPilelem. Concernant les autres types, nous sommes restés fidèles à ce que nous avons proposé dans le rapport intermédiaire.

II) Planning Définitif

	Tâche	Procédures	date	Personne
	Rédiger librairies		22/11/2019	Killian / Ludwig / Luca
Gestion Piles	Faire proc est_pile_vide	est_pile_vide	22/11/2019	Luca
	Faire proc empiler	empiler	22/11/2019	Luca
	Faire proc depiler	depiler	22/11/2019	Luca / Ludwig
	Faire proc init_pile	init_pile	22/11/2019	Luca
	Faire proc afficher_pile	afficher_pile	22/11/2019	Luca
	Faire proc taille_pile	taille_pile	22/11/2019	Luca
Gestion Fin du jeu	Faire proc liberer_pile	liberer_pile	18/12/2019	Luca
	Supprimer les totems, les mains des joueurs et la pioche pour la fin de partie.	nettoyage_partie	27/12/2019	Killian
	Faire proc liberer_main	liberer_main	18/12/2019	Luca
Gestion début de Partie	Faire proc init_pioche	init_pioche	29/11/2019	Killian
	Faire proc retirer_carte	retirer_carte	24/11/2019	Killian
	Faire le menu	main	30/12/2019	Luca / Ludwig / Killian
	Faire procédure d'initialisation de la liste des cartes	init_ListeCarte	22/11/2019	Killian
	Faire procédure pour mélanger les cartes	melanger	23/12/2019	Killian
	initialisation d'un tableau d'eniter pour determiner l'ordre d'empilage de la pioche	init_sans_doublons	23/12/2019	Killian
	Afficher les cartes de la pioche pour vérif	afficherCartes	23/12/2019	Killian
	Faire une fonction saisir entre	saisir_entre	26/12/2019	Killian
	Faire fonction pour distribuer les cartes	Distribuer_Cartes	26/12/2019	Ludwig
	Tirage au sort 1er Joueur	choix_premier_joueur	26/12/2019	Ludwig
Gestion Joueur	Initialiser la liste des joueurs.	init_Joueurs	21/12/2019	Killian
Gestion Points	initialiser_points	init_point	18/12/2019	Killian
	ajout_points	ajout_points	18/12/2019	Killian
	affichage_points	affichage_points	18/12/2019	Killian
Gestion Main / effet Cartes	faire pocédure piocher	Piocher	22/11/2019	Ludwig
	faire procédure DéposerCarteTotem	DéposerCarteTotem	18/12/2019	Ludwig
	Faire procédure Retrait_Carte_Main	Retrait_Carte_Main	18/12/2019	Ludwig
	Gérer le dépôt	JouerCarte	18/12/2019	Ludwig
	Gérer la défausse de carte	Defausser_Carte	18/12/2019	Ludwig
	Gérer effet Carte coup bas	JouerCarteCoup Bas	29/12/2019	Ludwig
	Gérer effet carte totem	JouerCarteTotem	27/12/2019	Luca / Ludwig
Gérer Tour	Proposer une action (deposer carte) ou defausser	tour_joueur	29/12/2019	Luca / Ludwig
	Contrer Vérifie si le joueur détient la carte qu'il veut jouer	verif_carte	29/12/2019	Luca

Aucun retard majeur n'est à signaler, le planning a été principalement respecté.

III) Jeux de tests

Nous avons codé et testé nos fonctions séparément. Malgré la présence de l'outil GIT dans notre projet, et la création d'un groupe de discussion sur l'application « Messenger », nous n'avons pas obtenu une organisation optimale. C'est au moment de la création du menu principal que nous nous en sommes rendu compte.

Ce constat est arrivé assez tardivement dans la réalisation du projet, de ce fait, nous n'avons pas de jeu de test à proposer.

Lorsque nous avons essayé de mettre nos travaux en communs, beaucoup de problèmes se sont posés, comme le manque de compréhension de certaines structures utilisées ou du fonctionnement de certaines procédures.

De plus, nous avons choisi de travailler sur plusieurs fichiers (utilisation de la notion de programmation modulaire) sensés regrouper les parties principales traitées par chaque membre du groupe.

- Les fonctions et procédures se trouvent dans le fichier `methodes.c`.
- Notre modèle de données est présent dans le fichier `modele_donnees.c`.
- Nos prototypes de fonctions et de procédures se trouvent dans le fichier `libraryProjet.h`.
- Nous avons mis en place un fichier `main.c` où se trouve un test simple de notre projet, **c'est ce fichier qui doit être compilé et exécuté.**
- Pour finir, nous avons laissé les traces de nos tests tout au long du projet dans un fichier nommé `test.c`. Ce fichier est entièrement commenté car, comme cité précédemment, il représente nos différents tests.

Notre manque d'expérience quant à l'utilisation de la programmation modulaire a sans doute accentué notre manque de cohésion.

Cependant notre programme compile sans erreur. La majorité de nos fonctions et procédures fonctionnent séparément.