BORDEAUX INP Ensc

Projet Informatique 2022

Le mémo à N-uplets

“Nuplémo”

Sommaire :

Introduction :

Dans le cadre de notre projet d'informatique du premier semestre, nous avons dû mettre en place la réalisation d'un jeu codé en C#.

Celui-ci se nomme le "nuplémo", et il s'agit d'un jeu dont l'objectif est de mémoriser l'emplacement de n-uplets de lettres constitués de m-cartes identiques de chaque lettre, dans une grille de dimensions n*m. Habituellement, ce jeu se nomme le "mémo" et se joue seul.

I/ Choix de la modélisation

Afin de respecter les contraintes et objectifs fixés par le projet d'informatique 2022, nous avons choisi de modéliser les cartes de mémory par un tableau rempli de caractères. Le dos d'une carte est ainsi représenté par le caractère * , tandis que la face est un type de caractère parmi les lettres minuscules, majuscules, les caractères usuels (!,:,(,'et d'autres...) et les chiffres(0 à 9). Le nombre maximal de N-uplet est donc fixé à 10, 14, 26 ou 94 suivant les choix du joueur, nous avons de plus fait attention à ne pas ré-utiliser les caractères spéciaux * et [espace].

Ainsi, l'utilisateur lorsqu'il choisit un type de caractères donné, il se voit prévenu du nombre maximal de N-uplet qu'il peut utiliser.

Exemple de grille de nuplémo de dimensions 4*3 avec les quadruplets de lettre 'a', 'k' et 's'

a	k	s	a
a	s	k	s
s	k	a	k

Les grilles cachées de nuplémo seront composées d'étoiles *. En voici un exemple de dimension 4*3 :

*	*	*	*
*	*	*	*
*	*	*	*

L'affichage de la grille se fait au cours de la partie et à chaque découverte d'une carte par un joueur, afin que celui-ci ait la possibilité de la mémoriser pour un prochain tour. La carte choisie est en couleur pour permettre à l'utilisateur de mieux se souvenir du déroulement de la partie. Pour que le jeu reste un jeu de mémoire, nous avons décidé d'effacer le terminal après 3 secondes d'affichage lors de la fin d'une tentative. De plus, le nombre de N-uplets restant à découvrir et le nombre de tentatives faites par le joueur sont donnés à titre informatif à chaque début de tour. Le nombre maximal de tentatives a été dans un premier temps déterminé arbitrairement, puis nous avons décidé de laisser ce choix au joueur.

Dans le cas d'une partie multijoueur, les scores sont comptabilisés à chaque fin de tour. Afin de garder un enchaînement clair, le joueur devant jouer est annoncé avant la demande de la carte à retourner.

De même, nous avons ajouté la règle suivante, fidèle au jeu du memory « classique » : Si un joueur forme un N-uplet(classiquement une paire), il rejoue. En fin de partie multijoueur, le gagnant est annoncé par le pseudonyme rentré en début de partie.

Dans un premier temps, les dimensions du tableau modélisant les cartes disposées sur la table sont le nombre de cartes formant un N-uplet, par le nombre de N-uplets. Cependant, l'option proposée d'optimisation de la taille du tableau nous a permis de le rendre le plus carré possible.

Nous avons également mis un système de sauvegarde en place. L'utilisateur au cours de la partie peut appuyer sur la touche "p" de son clavier pour mettre le jeu en pause. Un menu s'affiche ensuite et 3 options lui sont proposées:

- Retourner à la partie
- Sauvegarder la partie en cours
- Quitter le jeu

Concernant la sauvegarde. Nous enregistrons dans un fichier tous les éléments nous semblants indispensables pour un bon fonctionnement de la partie; mais également pour que l'utilisateur puisse se souvenir d'où il s'était arrêté au niveau de la partie. Ces informations sont donc :

- Le nombre de ligne du plateau
- Le nombre de colonne du plateau
- Le nombre de cartes identiques formant le n-Uplets
- Le nombre de n-uplets sur le plateau
- Le plateau de jeu courant
- L plateau de jeu final
- Le nombre de tentatives maximales
- Le nombre de tentatives qui se sont effectuées
- Le nombre de joueurs
- Quel est le joueur qui doit jouer lors de ce tour
- La dernière ligne choisi
- La dernière colonne choisi
- Le nombre de carte d'un même n-Uplets déjà retourné mais non complet
- Le nombre de n-Uplets qu'il reste dans le tableau courant
- Les pseudonymes des joueurs
- Le tableau des scores

Ainsi, si l'utilisateur décide de quitter la partie et de la reprendre ultérieurement il pourra utiliser cette sauvegarde. Si une sauvegarde existe, elle sera toujours proposée à l'utilisateur au début du jeu. Il peut la reprendre ou pas. Si une partie sauvegarder s'achève alors cette dernière est supprimée.

II/ Structuration du code

Le code proposé est composé des sous-fonctions suivantes :

Nom de la sous-fonction	Entrées	Sorties	Commentaire
DemanderCaract*	/	int	Renvoie un entier représentant le type de caractères choisi par le joueur
DemanderN	/	int	Renvoie le nombre de cartes formant un N-uplet choisi par le joueur
DemanderNuplets	/	int	Renvoie le nombre de N-uplet choisi par le joueur
OptimiserLaTaille*	(int nbCartesParUplets, int nbUplets)	int[]	Renvoie un tableau des 2 entiers diviseurs du produit des paramètres tels que ceux-ci soient les plus proches de sa racine carrée.
InitierTabGrille	(int nbCartesParUplets, int nbUplets)	char[,]	Renvoie le tableau “initial” correspondant à l’organisation des cartes lorsqu’elles sont face cachée
InitierTabCaractère	(int nbCartesParUplets, int nbUplets, int typeCaract)	char[,]	Renvoie le tableau “final” correspondant à l’organisation des cartes lorsqu’elles sont toutes retournées
DemanderNbTentativesMax*	(int int nbCartesParUplets, int nbUplets)	int	Renvoie le nombre maximal de tentatives si ce réglage est choisi par le joueur, et int.MaxValue si ce dernier décide de ne pas en avoir
NombreJoueurs*	/	int	Renvoie le nombre le joueur
DemanderPseudonymes*	(int nbrJoueurs)	string[]	Renvoie le tableau des pseudonymes choisis par chaque joueur
AfficherTab	char[,] tableau	void	Permet l’affichage d’un tableau sous la forme d’un plateau de

			cartes
AfficherTabCouleur	(char[,] tableau, int lignechoisi, int colonnnechoisi)	void	Permet l'affichage d'un tableau sous la forme d'un plateau de cartes et de mettre en bleu la dernière carte retournée au cours de la partie
DemanderLaCase	(ref int ligne, ref int colonne,tableau,char[,])	bool	Permet de demander à l'utilisateur une ligne et une colonne du tableau à révéler. De plus, l'utilisateur peut à tout moment mettre le jeu en pause. Renvoie true si l'utilisateur à mis sur pause, false sinon
Menu	(int nbCartesParUplets, int nbUplets, char[,] tableau, char[,] tableauResult, int nbTentativesRestantes, int nbMaxTentatives, int nbrJoueurs, int tourJoueurs, string[] pseudos, int[] tableauScore, int reste)	bool	Affiche le menu pause du jeu. L'utilisateur peut demander de sauvegarder, reprendre ou quitter la partie. Renvoie true si l'utilisateur veut quitter la partie, false si il veut la reprendre.
VerifierLaDisponibilite	(char[,] tableau, int ligne, int colonne)	bool	Renvoie true si les coordonnées rentrés en paramètres correspondent à une carte face cachée et false si celle-ci est retournée
Jouer	(ref char[,] tableau, char[,] tableauResult, int nbCartesParUplets, int nbUplets, int nbrJoueurs, string[] pseudos, int nbrTentatives, int tourJoueurs, ref int[] tableauScore, int nbMaxTentatives, int reste)	bool	Fonction qui s'occupe du déroulement du jeu. Le(s) joueur(s) tour par tour retourne les cases une à une. La tentative d'un joueur s'achève si il retourne une carte ne faisant pas partie du même NbUplets. Elle continue si toutes les cartes d'un même NbUplets sont trouvées. La partie s'achève quand toutes les cartes ont été retournées ou si l'utilisateur (ou les utilisateurs) a ordonné un nombre de tentatives maximum et que cette dernière est à 0. On annonce s' il y a un

			vainqueur à la fin. Cette fonction retourne true si l'utilisateur souhaite quitter une partie, false sinon .
TesterLaVictoire	char[,] tab	bool	Renvoie true si l'ensemble des cartes ont été retournées, false si ce n'est pas le cas
Sauvegarde*	(char[,] tableau,char[,] tableauResult, int nbCartesParUplets, int nbUplets, int nbTentativesRestantes, int nbMaxTentatives, int nbrJoueurs, int tourJoueurs, string[] pseudos, int[] tableauScore, int reste)	void	Fonction qui créer un fichier " PartiePrecedente.txt " Dans cette dernière on stocke toutes les données relatives à la partie.
SuppressionSauvegarde*	(char[,] tableauResult, int nbrJoueurs, string[] pseudos, int[] tableauScore, string path)	bool	Compare si il existe une partie déjà sauvegardée avec la partie qui vient de se terminer pour savoir si on supprime la sauvegarde. Retourne true si la partie doit être supprimer, false sinon ou si il n'y a pas de partie précédente sauvegardée.
TrierTab	(ref int[] tab, ref string[] pseudos)	void	Trie le tableau des pseudonymes des joueurs en fonction de leurs scores de manière décroissante
Gagnant*	(int tableauScore, string[] pseudos)	void	Si il y a plusieurs joueurs affiche le joueur ayant retourné le plus de cartes de NbUplets
Memory		bool	Fonction qui permet de mettre en place les éléments du jeu (plateau, demande de NbUplets et autre) via une sauvegarde ou en commençant à zéro. A la fin de la partie demande à l'utilisateur s' il veut recommencer une partie. Renvoie true si l'utilisateur veut quitter le jeu, false sinon.

*** optionnel**

III/ Organisation

- ➢ Séance 1 : Prévision des sous programmes, discussion des justifications et répartitions des tâches

- ➢ Séance 2 (en autonomie) : Mise au point sur les fonctionnalités de base restantes à coder et début du codage des options

- ➢ Séance 3 : Codage des options restantes et mise en place du document justificatif

- ➢ Séance 4 : Rédaction du document justificatif et Finalisation du projet

Les différentes tâches de la version de base du mémory ont été distribuées de façon la plus équitable possible entre les membres du groupe. En ce qui concerne les options, elles ont été réparties en fonction de l'avancée de chacun sur les tâches qui lui ont précédemment été attribuées. Afin de garder une certaine coordination entre les membres du groupe, un document partagé (google doc) a été créé pour communiquer le nom des variables, des fonctions, leurs types et éventuellement leurs paramètres pour permettre à chacun d'avancer à son propre rythme. Le but étant aussi d'avoir un document « référence » sur l'avancée des autres membres du groupe.

Des mises au point sur l'avancée ont été faites durant les horaires dédiés au projet informatique en parallèle de l'avancée du travail.