PROJET INFORMATIQUE

Sommaire:

I-Description et fonctionnement général du programme:

Le programme permet de lancer une interface générée par le biais de la bibliothèque <ncurses> et exécute le *Jeu de la vie* (Automate cellulaire établie par Conway: *Game Of Life*) suivant les spécifications demandées au préalable à l'utilisateur.

Description du programme étape par étape:

A l'exécution du programme, un message de bienvenue est affiché ainsi qu' un message d'information destiné à l'utilisateur indiquant qu'il pourra relancer le *Jeu de la vie* à tout moment en appuyant sur la touche ' r ' (comme *restart)* et quitter le *Jeu de la vie* en appuyant sur la touche ' q ' (comme *quit).*Ces options sont utilisables quelque soit les options choisis par l'utilisateur.

Puis le programme demande à l'utilisateur de choisir si il veut lancer le *Jeu de la vie* en mode normal, avec la variante Days and Nigth ou s'il souhaite visualiser des structures classiques en mode normale.
Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé au mode de jeu (0=Normal ; 1=Days and Nigth ; 2=structures classiques). Si l'utilisateur saisi un chiffre invalide (comme 5 ou 12 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide.

<u>Si l'utilisateur souhaite lancer le jeu en mode normale ou Days and Nigth:</u>

Le programme demande à l'utilisateur de choisir si il veut lancer le *Jeu de la vie* en *mode pas à pas* ou en *mode continu.*
Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé au mode de jeu (0=mode pas à pas ; 1=mode continu). Si l'utilisateur saisi un chiffre invalide (comme 5 ou 12 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide.

Puis le programme demande à l'utilisateur de choisir si il veut personnaliser son *Jeu de la vie.*
Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé à la réponse voulue (0=non-> je souhaite prendre des valeurs par défaut ; 1=oui-> je veux personnaliser mon jeu). Si l'utilisateur saisi un chiffre invalide (comme 5 ou 12 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide.

-Si l'utilisateur a choisit de ne pas personnaliser son *Jeu de la vie*:
alors le programme lance le *Jeu de la vie* avec une couleur des cellules prise par défaut, ainsi qu'un temps d'animation pris par défaut dans le cas où l'utilisateur aurait choisit le mode continu. Bien entendu, il lance le *Jeu de la vie* dans les modes choisis par l'utilisateur précédemment.

-Si l'utilisateur a choisit de personnaliser son *Jeu de la vie* et qu'il avait choisit auparavant le *mode pas à pas*:
alors le programme demande à l'utilisateur de choisir la couleur des cellules.
Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé à une couleur des cellules. Si l'utilisateur saisi un chiffre invalide (comme 12 ou 54 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide. Puis il lance le *Jeu de la vie* dans les modes précédemment choisis par l'utilisateur et avec la couleur des cellules choisie, pour le tableau du haut (tableau courant que l'on appellera *damier 1* par opposition au damier des évolutions que l'on appellera *damier 2*).

-Si l'utilisateur a choisit de personnaliser son *Jeu de la vie* et qu'il avait choisit auparavant le *mode continu*:
alors le programme demande à l'utilisateur de choisir le temps d'animation (on propose une valeur de 500 à l'utilisateur et on l'informe qu'une valeur plus grande ralentira le jeu et qu'à l'inverse, une valeur plus faible accélérera le jeu). Puis le programme demande à l'utilisateur de choisir la couleur des cellules.
Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé à une couleur des cellules. Si l'utilisateur saisi un chiffre invalide (comme 12 ou 54 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide. Puis il lance le *Jeu de la vie* dans les modes précédemment choisis par l'utilisateur avec un temps d'animation et une couleur des

cellules choisie par l'utilisateur lors de la personnalisation de son jeu.

Rappel important:
L'utilisateur peut relancer le *Jeu de la vie* à tout moment en appuyant sur la touche ' r ' (comme *restart)* et quitter le *Jeu de la vie* en appuyant sur la touche ' q ' (comme *quit).*Ces options sont utilisables quelque soit les options choisis par l'utilisateur.

Si l'utilisateur souhaite visualiser des structures classiques:

Le programme demande à l'utilisateur de choisir quel type de structures il souhaite visualiser: Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé au type de structures qu'il souhaite visualiser (0=structures stables ; 1=structures périodiques ; 2=vaissaux). Si l'utilisateur saisi un chiffre invalide (comme 5 ou 12 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide.

Quel que soit la structure classique choisit par l'utilisateur, le programme demande de choisir quelles structures-respectivement 0=stables, 1=périodiques ou 2=vaissaux-, il souhaite visualiser. Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé au type de structures -respectivement stables, périodiques ou vaissaux- qu'il souhaite visualiser. Si l'utilisateur saisi un chiffre invalide (comme 5 ou 12 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide.

Puis le programme demande à l'utilisateur de choisir si il veut

personnaliser sa structure classique.
Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé à la réponse voulue (0=non-> je souhaite prendre des valeurs par défaut ; 1=oui-> je veux personnaliser mon jeu). Si l'utilisateur saisi un chiffre invalide (comme 5 ou 12 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide.

-Si l'utilisateur a choisit de ne pas personnaliser sa structure: alors le programme lance la structure voulue avec une couleur des cellules prise par défaut, ainsi qu'un temps d'animation pris par défaut.

-Si l'utilisateur a choisit de personnaliser sa structure:
alors le programme demande à l'utilisateur de choisir le temps d'animation (on propose une valeur de 500 à l'utilisateur et on l'informe qu'une valeur plus grande ralentira le jeu et qu'à l'inverse, une valeur plus faible accélérera le jeu). Puis le programme demande à l'utilisateur de choisir la couleur des cellules.
Pour ce faire, l'utilisateur doit saisir un chiffre qui est associé à une couleur des cellules. Si l'utilisateur saisi un chiffre invalide (comme 12 ou 54 par exemple) alors le programme repose la question à l'utilisateur jusqu'à ce qu'il saisisse un chiffre valide. Puis il lance la structures avec un temps d'animation et une couleur des cellules choisie par l'utilisateur lors de la personnalisation de son jeu.

Remarque importante:
L'utilisateur peut quitter le *Jeu de la vie* en appuyant sur la touche ' q ' (comme *quit).* Cependant, l'option restart est désactivée lorsque l'on se trouve dans le mode des *structures classiques*. En

effet, cette option n'a que peu d'intéret dans ce mode.
L'option *quit* est utilisable quelque soit les options choisis par l'utilisateur.
De plus, lorsque l'on se trouve dans le mode des *structures classiques,* la visualisation s'effectue autommatiquement en mode continu.
Enfin, il est recommandé de choisir un temps d'animation de 5000 pour une bonne visualisation des strufctures périodiques.

II-CHOIX DE PROGRAMMATION:

Mode Normal:

Mode pas à pas	Mode continu
->On crée une matrice de type char que l'on appelera «*monde*»,de dimension TAILLEY*TAILLEX qui constitura le damier 1 (i.e: la grille avec la population courante => damier courant). ->On crée une matrice de type char que l'on appelera «*monde2*»,de dimension TAILLEY2*TAILLEX2 qui constitura le damier 2 (i.e: la grille avec les modifications à venir => damier des évolutions).	->On crée une matrice de type char que l'on appelera «*monde_c*»,de dimension (TAILLEY_C)*(TAILLEX_C) qui constitura le damier où les cellules évoluront.
->On initialise ncurses. On définit les couleurs que l'on utilisera par la suite pour les couleurs des cellules.	->On initialise ncurses. On définit les couleurs que l'on utilisera par la suite pour les couleurs des cellules.
->On crée une fonction qui fermera le terminal de ncurses.	->On crée une fonction qui fermera le terminal de ncurses.
->On remplit au hasard le damier1 de 0 ou de 1 grâce à la fonction *damier_hasard*. Les 1 symboliseront une	->On remplit au hasard le damier de 0 ou de 1 grâce à la fonction *damier_hasard_c*. Les 1 symboliseront

cellules en vie et les 0 symboliseront les cellules mortes. ->On remplit au hasard le damier2 de 0 ou de 1 grâce à la fonction *damier_hasard2*. Les 1 symboliseront une cellules en vie et les 0 symboliseront les cellules mortes. Ainsi, à l'instant 0, la disposition du damier 1 et du damier 2 sont identiques.	une cellules en vie et les 0 symboliseront les cellules mortes.
->On implémente une fonction *affiche* où l'on gère la coloration des cellules des damiers. Cette fonction prend en argument la couleur avec laquelle on veut colorer les cellules.	->On implémente une fonction *affiche_c* où l'on gère la coloration des cellules du damier. Cette fonction prend en argument la couleur avec laquelle on veut colorer les cellules.
->On implémente une fonction *nombre_voisins*, qui parcourt le damier et qui calcule pour chaque cellules le nombre de voisins vivants adjacents à la cellule qui servira à appliquer les règles de Conway.	->On implémente une fonction *nombre_voisins_c*, qui parcourt le damier et qui calcule pour chaque cellules le nombre de voisins vivants adjacents à la cellule qui servira à appliquer les règles de Conway.
->On fabrique une fonction *génération_suivantes* qui implémente les règles de Conway dans le but de determiner la génération suivante pour le damier 1. ->On fabrique une fonction *génération_suivantes2* qui implémente les règles de Conway dans le but de determiner la génération suivante pour le damier 2 et qui gère les couleurs suivant les différents statuts d'une cellule à la génération suivantes.	->On fabrique une fonction *génération_suivantes_c* qui implémente les règles de Conway dans le but de determiner la génération suivante pour le damier.
**********	->On implémente une fonction *temporise* qui permet de gérer la vitesse d'animation.

Mode Days and Night:

Mode pas à pas	Mode continu
->On crée une matrice de type char que l'on appelera «*monde*»,de dimension TAILLEY*TAILLEX qui constitura le damier 1 (i.e: la grille avec la population courante => damier courant). ->On crée une matrice de type char que l'on appelera «*monde2*»,de dimension TAILLEY2*TAILLEX2 qui constitura le damier 2 (i.e: la grille avec les modifications à venir => damier des évolutions).	->On crée une matrice de type char que l'on appelera «*monde_c*»,de dimension (TAILLEY_C)*(TAILLEX_C) qui constitura le damier où les cellules évoluront.
->On initialise ncurses. On définit les couleurs que l'on utilisera par la suite pour les couleurs des cellules.	->On initialise ncurses. On définit les couleurs que l'on utilisera par la suite pour les couleurs des cellules.
->On crée une fonction qui fermera le terminal de ncurses.	->On crée une fonction qui fermera le terminal de ncurses.
->On remplit au hasard le damier1 de 0 ou de 1 grâce à la fonction *damier_hasard*. Les 1 symboliseront une cellules en vie et les 0 symboliseront les cellules mortes. ->On remplit au hasard le damier2 de 0 ou de 1 grâce à la fonction *damier_hasard2*. Les 1 symboliseront une cellules en vie et les 0 symboliseront les cellules mortes. Ainsi, à l'instant 0, la disposition du damier 1 et du damier 2 sont identiques.	->On remplit au hasard le damier de 0 ou de 1 grâce à la fonction *damier_hasard_c*. Les 1 symboliseront une cellules en vie et les 0 symboliseront les cellules mortes.
->On implémente une fonction *affiche* où l'on gère la coloration des cellules du damier 1. Cette fonction prend en argument la couleur avec laquelle on veut colorer les cellules.	->On implémente une fonction *affiche_c* où l'on gère la coloration des cellules du damier. Cette fonction prend en argument la couleur avec laquelle on veut colorer les cellules.
->On implémente une fonction *nombre_voisins*, qui parcourt le damier et qui calcule pour chaque cellules le	->On implémente une fonction *nombre_voisins_c*, qui parcourt le damier et qui calcule pour chaque cellules le

nombre de voisins vivants adjacents à la cellule qui servira à appliquer les règles de Conway.	nombre de voisins vivants adjacents à la cellule qui servira à appliquer les règles de Conway.
->On fabrique une fonction *génération_suivantes_DN* qui implémente les règles de Conway selon la variante Days and Night dans le but de determiner la génération suivante pour le damier 1. ->On fabrique une fonction *génération_suivantes2_DN* qui implémente les règles de Conway selon la variante Days and Night dans le but de determiner la génération suivante pour le damier 2 et qui gère les couleurs suivant les différents statuts d'une cellule à la génération suivantes.	->On fabrique une fonction *génération_suivantes_DN_c* qui implémente les règles de Conway selon la variante Days and Night dans le but de determiner la génération suivante pour le damier.
**********	->On implémente une fonction *temporise* qui permet de gérer la vitesse d'animation.

III-<u>RÉSULTATS DE LA PARTIE 3:</u>

<u>Structures stable dans le mode normale</u>

Normal	Days and Nigth
Carré	Reste stable
Tube	Toute la population disparait après la première itération
Bâteau	Toute la population disparait après la deuxième itération
Navire	Périodique de période 4
Serpent	Toute la population disparait après la deuxième itération

Barge	Toute la population disparait après la deuxième itération
Porte-avion	Toute la population disparait après la première itération
Ruche	Toute la population disparait après la première itération
Miche de pain	Toute la population disparait après la première itération
Hameçon	Toute la population disparait après la deuxième itération
Canoë	Toute la population disparait après la deuxième itération
Longue barge	Toute la population disparait après la deuxième itération (Après la première itération, on obtient un tube. Puis il disparaît à la génération suivante)
Long navire	Toute la population disparait après la troisième itération (Après la première itération, on obtient une longue barge. Puis elle disparaît apès deux générations)
Mare	Toute la population disparait après la première itération
Long canoë	Toute la population disparait après la deuxième itération
Double hameçon	Toute la population disparait après la deuxième itération

Structures périodiques dans le mode normale

Normal	Days and Nigth
Clignotant	Toute la population disparait après la deuxième itération
Grenouille	Toute la population disparait après la deuxième itération
Horloge	Après neuf itérations, on obtient 3 carrés stables
Pentadecathlon	Toute la population disparait après la quatrième itération
Galaxie de kok	Toute la population disparait après la huitième itération
Croix	Après la quatorzième itérature,on aboutie à une forme stable
Octogone	Toute la population disparait après la deuxième itération
Fontaine	Toute la population disparait après la dixième itération
Diagonale	Périodique de période 4 après la deuxième itération

Vaisseaux dans le mode normale

Normal	Days and Nigth
Planeur	Toute la population disparait après la sixième itération
Vaisseaux LWSS	Toute la population disparait après la cinquième itération
Vaisseaux HWSS	Périodique de période 4 après 19 itérations
Canon	Après la 17-ième itération, il reste deux carrés stables