PROJET LANGAGE C & SYSTEMES

eXia Games

Rentrez dans une nouvelle dimension du jeu

Table des matières

1. Introduction

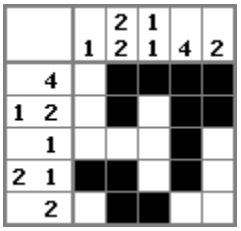
France, 03/12/2014.

Grâce à l'immense succès du jeu CesiMillions, l'entreprise a décidé de se lancer dans le développement d'autres jeux interactifs : le jeu du hanjie et quelques autres plus traditionnels (memory et sudoku losange). Ces nouveaux jeux demandent la mise en place d'un ou plusieurs logiciels. Les étudiants de première année de l'Exia.Cesi dans toute la France vont être mis à contribution car ils ont déjà une bonne expérience sur un module du jeu CesiMillions implémenté lors du prosit 5 du module Algo&C.

2. Le jeu du « Hanjie »

Le jeu du hanjie est un jeu solitaire qui permet de trouver un dessin à partir d'une grille comme celles ci-dessous.

Le but d'un hanjie est de noircir les cases de la grille afin de faire apparaître une image, un dessin. Les nombres à gauche et au-dessus de la grille sont là pour vous aider à déduire les cases à noircir.



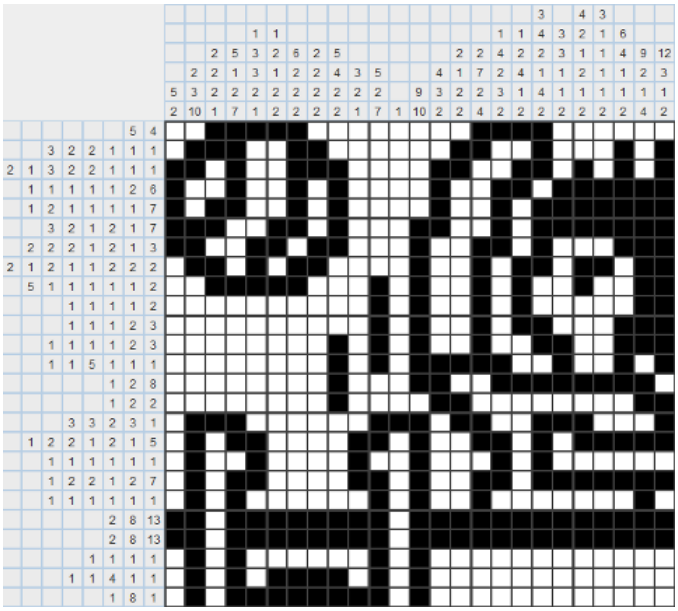
« Hanjie très simple – il peut difficilement nuire à la santé. ☺»

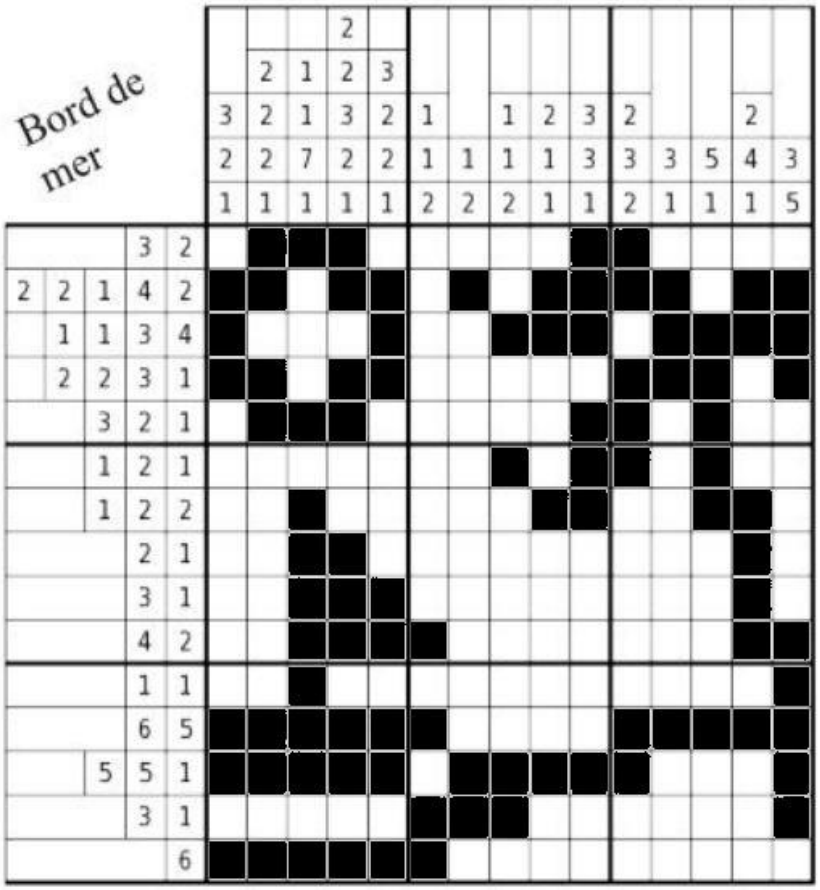
Les nombres présents à gauche de la grille indiquent le nombre de cases à noircir sur la ligne correspondante.
Les nombres présents en haut de la grille indiquent le nombre de cases à noircir sur la colonne correspondante.
Les chiffres représentent uniquement les séries des cases à noircir, pas le nombre de blancs entre les cases noires.

Vous trouverez de plus amples informations sur ce jeu ainsi qu'une série d'astuces pour le résoudre aux adresses suivantes :

http://xunor.free.fr/enigmes/hanjie/
http://www.hanjie-star.fr/picross/regles-et-tutoriels/regles-du-jeu.html

Voici quelques autres exemples un peu plus difficiles :





« Hanjie plus difficile – il peut vite être un peu casse-pied. 💣☹ »

3. Cahier des charges fonctionnel

Spécifications communes pour tous les jeux que l'entreprise souhaite développer :

- Choix du niveau : débutant, intermédiaire, expérimenté.
- Affichage du temps passé pour résoudre le jeu, sans limitation de temps
- Lancer le jeu.
- A la fin de la partie et de l'affichage du résultat, vous devez revenir au menu principal du jeu.
- Chargement et enregistrement de l'historique
- Affichage de l'historique et différents tris possibles en fonction des critères.
- Possibilité d'enregistrer dans un fichier binaire le jeu à n'importe quel moment de la partie et de pouvoir le recharger également pour continuer la partie. (*avec les connaissances acquises avec le prosit de primitives système*). Quand on enregistre un jeu, la partie s'arrête et on revient au menu principal.

Dans un premier temps, l'entreprise veut se concentré que sur le jeu du **hanjie**. Voici les fonctionnalités spécifiques pour ce jeu **:**

- Proposer 3 grilles possibles par niveau.
- Chargement en mémoire du fichier correspondant
- Affichage de la grille
- Pour afficher les cases du hanjie, vous devez donner deux coordonnées de la façon suivante :
  - « Griser » un ensemble de cases dans la même ligne : B3 G10
  - « Griser » un ensemble de cases dans la même colonne : B3 B7
  - « Griser » une seule case : D4 D4
- En cas d'erreur, si vous vous êtes trompé et que vous avez grisé une case (ou plusieurs) et que vous voulez la « dégriser », il suffit de saisir à nouveau ses coordonnées. Cette action effacera la case si elle était « grisée ».
- L'algorithme doit détecter la fin de la partie quand toutes les cases ont été grisées

1. *Structures de données*

Vous devez réfléchir à une structure pour représenter le jeu en mémoire.

Le choix de la structure de données est le plus important à faire au début du projet car de cela dépendront tous les algorithmes que vous allez implémenter.

Vous trouverez une grille en annexe qui vous aidera à formaliser les structures des données ainsi que les prototypes des fonctions.

Voici quelques conseils :

- Pensez à des tableaux à deux dimensions (statiques ou dynamiques)
- Pour le contenu de chaque case du tableau, pensez à faire des « struct » avec l'ensemble des informations dont vous aurez besoin.
- L'utilisation des pointeurs peut sans doute vous simplifier l'implémentation de certains algorithmes et vous faciliter la manipulation de ces structures de données.

2. Format des fichiers

2.1. Historique des jeux

Vous devez gérer un fichier qui enregistrera l'historique des parties. Vous devez stocker dans ce fichier :

- Le nom du joueur ou son alias.
- Le nom du jeu, son type, le niveau.
- La date et l'heure du jeu.
- Le temps réalisé

Voici un exemple de fichier à construire, commun à tous les jeux :

```
#Format de chaque colonne.
#Séparateur ; entre chaque colonne.
#Dernière colonne, \n (saut de ligne)
#
#Date format jj/mm/yyyy;
#Heure HH:MM:ss;
#Alias (taille max 50 caractères);
#Jeux : 1=memory, 2=sudoku, 3=hanjie;
#Niveau : 1=débutant, 2=intermédiaire, 3=expériemnté
#Temps HH:MM:ss
10/10/2014;15:15:10;Luc SkyWalker;1;1;00:00:27
10/10/2014;05:25:45;Capitain Spot;2;2;00:01:46
11/12/2014;13:07:23;Luc SkyWalker;3;3;00:10:37
11/12/2014;15:15:12;Capitain Spot;2;1;00:00:56
15/01/2015;15:15:04;Teal'c;1;2;00:00:10
15/01/2015;15:15:30;Princesse Anna;3;1;01:04:43
```

L'enregistrement se fait à chaque fin du jeu. Maintenir ce fichier trié par date/heure
NOTA : ce format est donné à titre d'exemple. Vous pouvez vous en inspirer ou créer le vôtre si celui-ci ne vous convient pas.

2.2. Les fichiers « images » pour le Hanjie

Pour chaque image hanjie, vous devez construire un fichier PBM ASCII avec la syntaxe suivante **à respecter scrupuleusement** :

- Le nombre magique du format : il dépend du format et de la variante (binaire ou ASCII).
- Un caractère d'espacement (espace, tabulation, nouvelle ligne)
- La largeur de l'image (codée en caractères ASCII)
- Un caractère d'espacement
- La hauteur de l'image (codée en caractères ASCII)
- Un caractère d'espacement
- Les données binaires de l'image :
  - L'image est codée ligne par ligne en partant du haut
  - Chaque ligne est codée de gauche à droite
- Toutes les lignes commençant par # sont ignorées.

Le format de fichier PBM est utilisé pour des images noir et blanc. Un pixel noir est codé par un caractère 1, un pixel blanc est codé par un caractère 0. Un fichier PBM ASCII a pour nombre magique P1. Dans les données binaires de l'image, les caractères d'espacement à l'intérieur sont ignorés. Aucune ligne ne doit dépasser 70 caractères.

Exemple

```
P1
# Un exemple bitmap de la lettre "J"
6 10
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 0 0 1 0
1 0 0 0 1 0
0 1 1 1 0 0
0 0 0 0 0 0
0 0 0 0 0 0
```

Pour plus d'information : http://fr.wikipedia.org/wiki/Portable_pixmap

Vous trouverez plusieurs fichiers « image hanjie » dans les ressources. Vous pouvez en créer autant que vous le souhaitez.

3. Menus, saisie de données et affichages

Vous devez implémenter un menu principal pour chaque logiciel avec les fonctionnalités attendues ci-dessus.

3.1. Menus et sous-menus

Un menu initial sera présenté avec les fonctionnalités suivantes :

- S'identifier : donner son alias ou son nom
- Choix du niveau du jeu : débutant, intermédiaire, expérimenté.
- Choix de la taille d'affichage des résultats

D'autres sous-menus seront certainement nécessaires et leur implémentation reste entre vos mains.

Pour les affichages, vous devez utiliser certains caractères UFT-8 :

- Pour les blocs, http://www.fileformat.info/info/unicode/block/block_elements/utf8test.htm
- Pour les contours des tableaux, http://www.fileformat.info/info/unicode/block/box_drawing/utf8test.htm

Comment afficher ces caractères en C en utilisant « printf » classique ? Très simplement :

```
// par exemple pour le symbole ▉ on a : UTF-8 (hex)   0xE2 0x96 0x88 (e29688)
// et on place donc :
printf("%c%c%c\n", 0xE2, 0x95, 0x90);
```

Créez-vous des fonctions d'affichages qui vous serviront, dans le futur, pour tous les jeux.

Si vous n'arrivez pas à utiliser ces caractères-là, choisissez n'importe quel autre caractère ASCII ou UTF-8. Le caractère utilisé n'influera en rien dans la note finale du projet.

3.2. Affichages pour le Hanjie

			A	B	C	D	E
				2	1		
			1	2	1	4	2
1		4					
2	1	2					
3		1					
4	2	1					
5		2					

Grille Initiale

			A	B	C	D	E
				2	1		
			1	2	1	4	2
1		4		■	■	■	
2	1	2		■	■	■	
3		1				■	
4	2	1	■	■		■	
5		2		■			

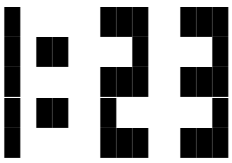
Grille pendant le jeu

ASTUCE : pensez bien aux nombres de colonnes et de lignes pour chaque case du « tableau de jeu » pour que le dessin soit visuellement correct.

IMPORTANT : Réfléchissez bien, pour les hanjie plus complexes, à la gestion de l'affichage des colonnes et des lignes si le nombre de grisées sur une ligne ou une colonne est supérieur à 9.

3.3. Affichage du temps

A la fin de la partie de n'importe quel jeu, l'affichage du temps (en secondes ou en minutes et secondes, voire en heures ☺!) se réalisera en simulant un afficheur digital avec une taille par défaut de 5 lignes x 3 colonnes pour chaque chiffre, avec une séparation de 2 colonnes entre deux chiffres. Par exemple :



La taille de l'affichage est paramétrable dans le menu principal. Les tailles possibles sont les suivantes : 5x3, 7x4, 9x5, 11x6, etc … **jusqu'à un maximum raisonnable pour la console**.

Pour la bonne réalisation de cette fonctionnalité, vous devez stocker uniquement que la modélisation des chiffres pour 5x3. Toutes les autres doivent être affichées que par des calculs à partir de la modélisation de base.

ASTUCE : la taille d'un chiffre d'un afficheur digitale, peut être modélisée par un tableau statique à deux dimensions avec des 1 et des 0.

3.4. Gérer le temps

La gestion du temps doit être utilisée pour :

- calculer le temps passé pour une partie.

La gestion du temps se fera avec la fonction « time() ». Avec cette fonction vous allez pouvoir calculer le temps passé pour résoudre un jeu en calculant le temps au début et à la fin de la partie.

4. Modalités d'évaluation

L'évaluation du projet se réalisera sur la base de critères suivants :

- **Soutenance, Supports et Documentation** :

	A
La soutenance (oral) [Comportement professionnel]	La soutenance est parfaitement préparée. La présentation est dynamique. Le passage de parole est fluide et pertinent.
Supports (Forme) [Comportement professionnel]	Le support de type Power Point est agréable et construit de manière pertinente et soignée. Le rapport est agréable à lire et aéré. Tous les documents sont soignés. Pas de faute d'orthographe.
Documentation (Fond) [Savoir]	Tous les documents sont construits de manière pertinente. Toutes les informations nécessaires apparaissent dans le rapport. Aucune erreur n'apparait dans les informations du rapport.

- **Efficacité & Fonctionnement** : vous allez gagner des points en fonction des fonctionnalités implémentées. Concentrez-vous sur le jeu du hanjie. A lui tout seul peut vous donner 100 points. Voici la grille.

JEUX	FONCTIONNALITE	NOMBRE DE POINTS MAX
HANJIE	Menu(s) et options disponible(s)	5
	Affichage du temps à la fin de chaque jeu selon spécification demandée	15
	Historique des parties et des jeux	10
	Choix possible entre plusieurs niveaux et fichiers	10
	Chargement en mémoire d'un fichier hanjie	10
	Affichage console d'un hanjie selon spécification fonctionnelle	15
	Joueur au jeu - saisie de coordonnées à griser et affichage sur la console	10
	Vérification de la fin d'une partie	10
	Enregistrement et rechargement d'un hanjie en cours	15
	TOTAL POINTS REALISES	**100**

Cette même grille d'efficacité sera utilisée par le jury pour évaluer les performances de groupe et individuelle.

Vous pouvez avoir des points supplémentaires si vous réalisez un ou plusieurs bonus **(voir chapitre 7 – BONUS)**:

BONUS	Memory	15
	Sudoku "losange"	15
	Hanjie en couleur	10

- **Choix techniques** implémentés :

ITEM	A
STRUCTURES DES DONNES (coeff 3)	Utilisation des structures dynamiques (tableaux ou listes chaînées) Utilisation des pointeurs Les algorithmes utilisés sont fonctionnels sur ces structures
LECTURE/ECRITURE DU FICHIER (coeff 3)	La lecture des fichiers permet le chargement correcte des données en mémoire L'écriture dans un fichier se réalise correctement La structure du fichier d'échange est respecté
FONCTIONNALITES DE BASE (coeff 2)	Utilisation des printf pour l'affichage des menus Utilisation correcte de la fonction scanf ou équivalent pour lire les données saisies par l'utilisateur à la console Utilisation correcte des boucles et conditions pour l'affichage des tableaux de jeux (hanjie, memory, sudoku)
UTILISATION DE FONCTIONS (coeff 2)	Le code n'est pas redondant Le code est découpé en fonctions Les noms des fonctions et des paramètres sont assez parlants. Les fonctions sont utilisées au moins une fois
CODE COMMENTE (coeff 1)	Les commentaires représentent plus de 10% du code.
DECOUPAGE EN FICHIERS (coeff 1)	Le programme est découpé en 3 fichiers .c ou plus. Le contenu de fichiers est bien identifié (affichage, actions, etc …) Les structures de données et les prototypes des fonctions se trouvent dans le .h

5. Livrables techniques attendus

Les livrables sont les suivants :

- Un exécutable pour le hanjie (**OBLIGATOIRE**).
- Un exécutable par jeu (**SI BONUS REALISES**).

Pour chaque projet, le code doit être organisé de la façon suivante :

- Fichier pour les affichages (.c et .h) : contient toutes les fonctions/procédures nécessaires pour les affichages à l'écran, les saisies des informations, la validation de ces informations d'entrée et la gestion des messages d'erreurs.
- Fichier pour les actions (.c et .h) : contient les structures de données et les fonctions/procédures nécessaires pour les calculs
- Fichier main.c contenant la fonction main et les appels aux fonctionnalités demandées.

6. Informations complémentaires

1. Contraintes techniques

Tous les développements doivent être réalisés sur un ***environnement Linux***. Lors de la soutenance vous devrez montrer tous les développements et les démonstrations sur Linux.

ATTENTION : aucun projet ne sera recevable s'il est développé sur un autre environnement que Linux. La note de groupe sera automatiquement « **D** ».

Vous êtes libre d'utiliser ou pas un IDE sur Linux (Qt est installé sur la VM livrée pour le module Langage C).

2. Conseils

Ce projet n'a aucune complexité ni en « mathématiques » ni en algorithmique. Concentrez-vous sur la programmation en C, les structures de données, le découpage du code et les affichages de chacune des demandes et tout se passera très bien. **Concentrez-vous sur les fonctionnalités obligatoires** et vous ferez les **BONUS** s'il vous reste du temps.

Les spécifications de ce projet vous permettent d'assigner des fonctionnalités distinctes à chaque personne du groupe, si vous le souhaitez. **Cela permettra aux jurys de vous évaluer mieux et plus facilement sur votre contribution individuelle**.

ASTUCE : par exemple**,** en attendant que le chargement du fichier hanjie soit réalisé, simulez le contenu de ce fichier « en dur » en mémoire.

Pour avoir une meilleure note, vous devrez montrer vos connaissances sur les structures d'allocation dynamique et une bonne maîtrise de structures de données utilisées. Le **BONUS** ne pourra qu'améliorer votre note finale.

Si vous implémentez des algorithmes de tri et de recherche, vous pouvez, dans un premier temps utiliser des algos simples (bulle, insertion, sélection) et dans un deuxième temps implémenter un algorithme plus efficace (merge, quicksort). De même pour la recherche (séquentielle puis dichotomique), mais, attention, pour que la recherche dichotomique soit efficace, la liste doit être préalablement triée.

Vous pouvez utiliser n'importe quelle structure de données utilisée dans le module Algo/C. Réfléchissez bien à la plus adaptée à chaque problématique.

IMPORTANT : Il y a 10 chiffres (0-9) à coder pour chaque taille d'affichage. Si on compte environ une dizaine de tailles différentes, ce n'est pas envisageable **de stocker tous les chiffres en mémoire pour toutes les tailles pour l'affichage du temps.** Vous devez réfléchir à des algorithmes qui vous permettront de réaliser le « zoom » à partir d'un élément de base codifié dans une structure de données dans le logiciel.

3. Organisation du projet

Le projet se déroule en groupes de **3 personnes** (maximum 4) du 4 au 12 décembre 2014

- Le vendredi 5 décembre, chaque groupe doit rendre le document décrivant l'analyse du problème et le choix des fonctions et variables (cf. document "Feuille d'avancement"). Ces choix seront argumentés lors d'un rendez-vous avec le tuteur et/ou « l'assistant tuteur d'A5 ».

- Les documents attendus (**ci-dessus**) sont à rendre le jeudi 11 décembre à 12h00 au plus tard en version électronique par mail et en version papier au tuteur responsable du projet.

    - Un dossier technique comprenant un rappel du projet, la répartition finale des tâches des membres du groupe, la description de la réalisation effective du projet et l'utilisation de l'interface et un bilan de groupe et individuel. En annexe, le groupe fournira aussi le code source commenté.

    - Le code source et les exécutables du projet (en .zip).

- L'après-midi du 11 décembre sera consacré à la préparation de la soutenance.

- Dans la journée du 12 décembre, chaque groupe devra présenter et défendre son travail au cours d'une soutenance orale de 20 minutes avec séance de questions-réponses d'une même durée. Chaque membre du groupe devra prendre la parole de manière équitable.

7. Bonus

Les bonus ne sont pas obligatoires. La réalisation d'un ou plusieurs bonus peut vous permettre d'améliorer votre note finale. La non réalisation ne vous empêchera pas d'atteindre « A » dans la note technique de groupe. La réalisation d'un des bonus sera prise en compte par le jury dans la **note technique & fonctionnel de groupe et individuelle.**

1. Memory

Le jeu consiste à trouver les couples d'images ou d'objets cachés sur un ensemble de « cartes ».

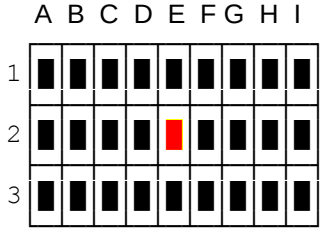
« L'abus d'alcool est dangereux pour la santé, à consommer avec modération »

L'objectif est de trouver tous les couples le plus rapidement possible.

En plus de spécifications communes du chapitre 3, voici les fonctionnalités spécifiques :

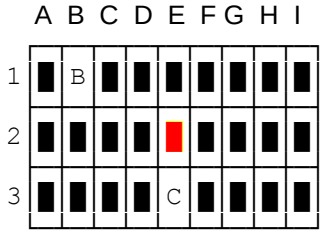
- En fonction du niveau, choisir un ensemble de couples (lettres, chiffres, caractères spéciaux, etc …). Le choix de l'ensemble de couples se fera par le programme et pas par le joueur.
  - Moins de 5 couples pour le niveau débutant, donc un tableau de 3x3 suffit
  - Entre 5 et 15 couples pour niveau intermédiaire
  - Plus de 15 couples pour le niveau expérimenté.
- La génération du memory doit être faite par le programme de façon aléatoire en plaçant les couples et aussi le(s) trou(s) s'il y en a.
- Pour signaler au jeu les cases à retourner il faut donner les coordonnées de deux cases, par exemple : A1 F3
- Si les deux cases retournées correspondent à un couple, vous ne cachez plus ces deux cases. Au contraire, s'il ne s'agit pas d'un couple, au bout d'un temps déterminé (2/3 secondes, par exemple) vous devez les retourner, supprimer tout affichage antérieur (pour éviter « des tricheries ») et afficher la grille au même état qu'avant le dernier « mouvement ». (Vous trouverez plus d'information ci-dessous)
- L'algorithme doit détecter la fin de la partie quand tous les couples ont été trouvés

Pour l'affichage du memory, au début de la partie, le tableau doit être comme suit :



Pour le jeu du memory, si votre tableau comporte un nombre impair de cases, vous devez en choisir une (ou plusieurs) et la marquer pour qu'elle ne soit pas utilisée pendant le jeu (cela signifie que cette case ne cache aucune valeur).

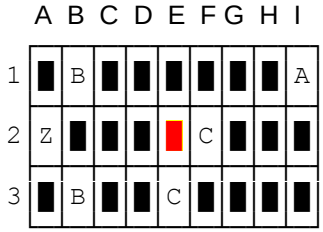
Au moment où le joueur découvre 2 cases, vous devez montrer les cases découvertes puis au bout de 2 ou 3 secondes les cacher à nouveau.



Pour effacer toute la console, vous utiliserez la fonction système de la façon suivante :

```
system("clear");
```

Au milieu du jeu, vous devrez avoir des cases déjà découvertes et des cases encore cachées :



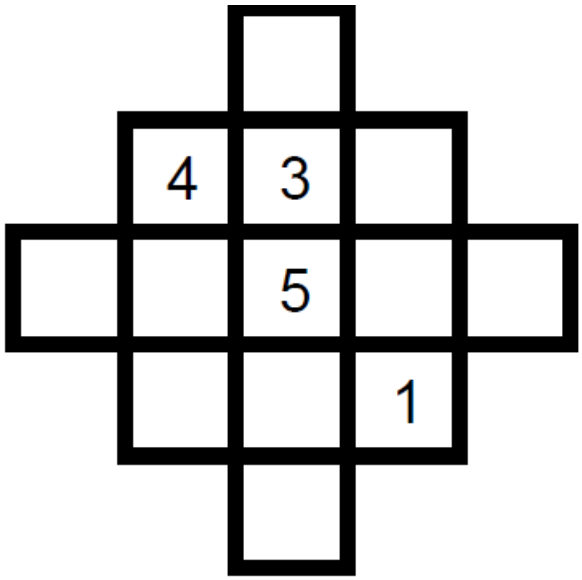
La gestion du temps doit être utilisée lors du :

- Effacement de la console après un « mouvement » au memory

Pour l'effacement de la console après un nombre de secondes, utilisez la primitive sleep() (*vu dans le WS 6*)

2. « Sudoku » Losange

L'objectif du jeu est de compléter les cases vides avec des chiffres du 1 à N (dans notre exemple de 1 à 5) de telle façon qu'aucun chiffre ne se répète ni dans les lignes, ni dans les colonnes ni dans les diagonales.



« L'abus des jeux peut nuire à la santé, à pratiquer avec modération »

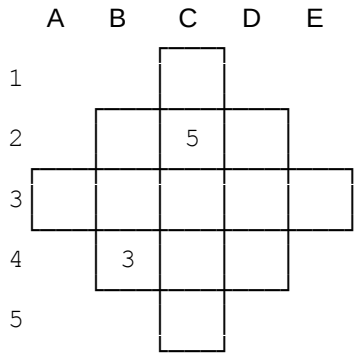
Ce jeu se joue uniquement en solitaire et l'objectif est de le résoudre le plus rapidement possible.

En plus de spécifications communes du point 3, voici les spécifications fonctionnelles :

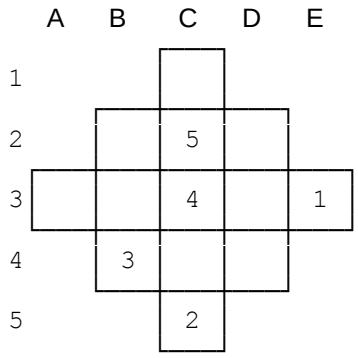
- En fonction du niveau, choisir un losange de taille
  - Débutant : 3x3
  - Intermédiaire : 5x5, 7x7
  - Expérimenté : 9x9
- La génération de la solution en mémoire doit être faite par le programme de façon aléatoire.
- Afficher plusieurs chiffres sur la grille de façon aléatoire pour faciliter la résolution du jeu. Le nombre de chiffres affichés peut varier en fonction du niveau choisi.
- Pour signaler au jeu le chiffre suivant à rajouter il faut donner les coordonnées d'une case plus le chiffre à rajouter dans cette case, par exemple : B1 4 (voir sous-chapitre 3 pour plus d'informations). Si vous vous êtes trompé et vous voulez changer le chiffre d'une case, il vous suffit de refaire la même chose que quand il n'y a pas encore un chiffre d'affiché (Par exemple, si je me suis trompé dans le mouvement antérieur, je tape B1 2).
- L'algorithme doit détecter la fin de la partie quand toutes cases sont remplies et que la solution a été trouvée.

ATTENTION : *dans certains cas, plusieurs solutions sont possibles. Vous pouvez donner par bonne une solution qui répond aux spécifications du jeu, même si celle-ci n'est pas exactement celle que vous avez générée en mémoire.*

Voici les affichages au début de partie et en cours de partie :

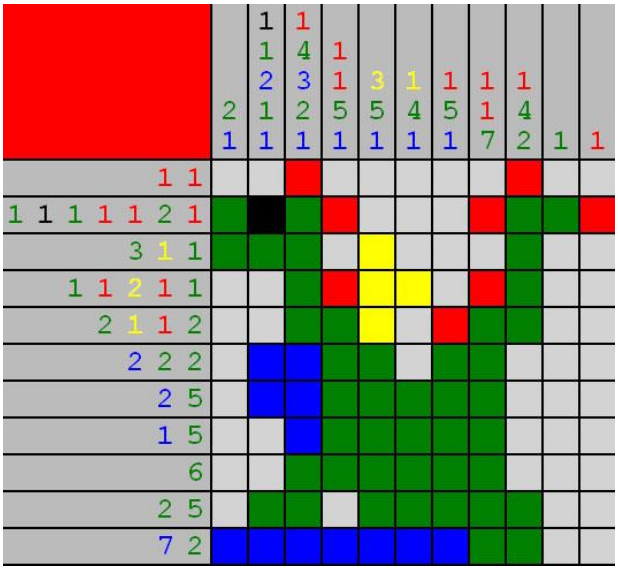


Grille initiale



Grille pendant le jeu

3. Hanjie en couleur



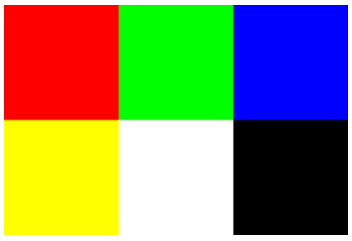
Le mécanisme est le même qu'avec le hanjie en noir sauf que le format utilisé sera le PPM.

Le format de fichier PPM est utilisé pour des images couleur. Chaque pixel est codé par trois valeurs (rouge, vert et bleu). Comme le format PGM, en plus des caractéristiques de largeur et hauteur, une valeur maximale utilisée pour coder les niveaux de couleur ; cette valeur doit être inférieure à 65536.

Un fichier ppm ASCII a pour nombre magique P3. Chaque pixel est codé en caractères ASCII, précédés et suivis par un caractère d'espacement. Aucune ligne ne doit dépasser 70 caractères.

Exemple

```
P3
# Le P3 signifie que les couleurs sont en ASCII,
# par 3 colonnes et 2 lignes,
3 2
# ayant 255 pour valeur maximum, et qu'elles sont en RGB.
255
255   0   0     0 255   0     0   0 255
255 255   0   255 255 255     0   0   0
```



http://fr.wikipedia.org/wiki/Portable_pixmap

Cet article sur les affichages en couleur sur Linux vous sera utile pour ce bonus.
http://www.linuxforums.org/forum/programming-scripting/88-color-console.html

"Text color output is not defined in ANSI C/C++. Instead the creators of the language left that to be operating system dependent. In Linux, to change text color you must issue what are known as terminal commands. To do this you just change your output statement to contain a terminal command."

```c
#include <stdio.h>

#define ANSI_COLOR_RED     "\x1b[31m"
#define ANSI_COLOR_GREEN   "\x1b[32m"
#define ANSI_COLOR_YELLOW  "\x1b[33m"
#define ANSI_COLOR_BLUE    "\x1b[34m"
#define ANSI_COLOR_MAGENTA "\x1b[35m"
#define ANSI_COLOR_CYAN    "\x1b[36m"
#define ANSI_COLOR_RESET   "\x1b[0m"

int main (int argc, char const *argv[]) {

  printf(ANSI_COLOR_RED     "This text is RED!"     ANSI_COLOR_RESET "\n");
  printf(ANSI_COLOR_GREEN   "This text is GREEN!"   ANSI_COLOR_RESET "\n");
  printf(ANSI_COLOR_YELLOW  "This text is YELLOW!"  ANSI_COLOR_RESET "\n");
  printf(ANSI_COLOR_BLUE    "This text is BLUE!"    ANSI_COLOR_RESET "\n");
  printf(ANSI_COLOR_MAGENTA "This text is MAGENTA!" ANSI_COLOR_RESET "\n");
  printf(ANSI_COLOR_CYAN    "This text is CYAN!"    ANSI_COLOR_RESET "\n");

  return 0;
}
```

All the colors that I have found are:

\033[22;30m - black
\033[22;31m - red
\033[22;32m - green
\033[22;33m - brown
\033[22;34m - blue
\033[22;35m - magenta
\033[22;36m - cyan
\033[22;37m - gray
\033[01;30m - dark gray
\033[01;31m - light red
\033[01;32m - light green
\033[01;33m - yellow
\033[01;34m - light blue
\033[01;35m - light magenta
\033[01;36m - light cyan
\033[01;37m - white

QUE LA FORCE SOIT AVEC VOUS !