J’ai créé un nouveau dossier *Projet* car on a décidé de repartir sans CTAT.

# Général

Le sudoku se compose d’instance d’éléments Case

# Interface (Gilles)

L’interface restera celle établie par Gilles.

Il faudra gérer l’affichage d’un sudoku comme suit :

*Case* sudoku[][] = new *Case*[9][9] ;

Les Case se compose d’un chiffre principal (=0 si la case est blanche) et d’un table de 9 bool (*true* si la chiffre peut être dans cette case)

* Voir le fichier Case.java

On affichera donc par case le chiffre se trouvant dans la case OU l’ensemble des chiffre hypothétique contenu dans le tableau.

Le top serait d’établir plusieurs fonctions au clic gauche de la souris :

* Un mode crayon dans laquelle l’utilisateur peut cliquer sur des boutons représentant les hypothèses à l’intérieur des cases,
* Un mode stylo pour lorsque l’on clique sur une case, on puisse rentrer un chiffre à valider,
* Un mode gum pour effacer le chiffre et revenir à l’affichage des hypothèses.

Gilles vois si c’est jouable.

# Tuteur (Jules)

Le tuteur réagira dans 3 cas avec des phrases toute faites

1. Le joueur entre une réponse fausse
   * On informe le joueur qu’il a faux,
   * On lui indique la règle violé
2. Le joueur rentre une réponse incertaine (potentiellement bonne mais n’appartenant à aucune stratégies)
   * On lui dit de faire attention,
   * On lui dit que c’est un imprudent,
   * On lui dit qu’il peut résoudre le sudoku sans recourir à l’aléatoire (car on se concentre sur les sudoku faciles pour le moment
3. Le joueur ne joue pas pendant un moment
   * On regarde les coups possibles et on le renseigne dans la région où il peut jouer ?
   * Ou on l’informe sur la stratégie à appliquer (genre *essayes de repérer dans tes hypothèse les couples de chiffre présents dans une même région*)

# Stratégies (Maxime)

Vocabulaire :

* **Case** : plus petit élément du sudoku. Une case peut être vide ou bien peut contenir un chiffre inscrit ou bien contenir entre 1 et 9 chiffres hypothétiques. Il existe en tout 81 cases.
* **Grille** : ensemble des cases du sudoku. Une grille de sudoku est divisé en colonnes, ligne et région.
* **Chiffre**: valeur que peut prendre une case, entre 1 et 9
* **Colonne**: ensemble des 9 cases qui se trouvent à la verticale par rapport à une case. Il existe en tout 9 colonnes qui sont numérotés de 1 à 9, de la plus à gauche à la plus à droite.
* **Ligne**: ensemble des 9 cases qui se trouvent à l’horizontale par rapport à une case. Il existe en tout 9 lignes qui sont numérotés de 1 à 9, de la plus à gauche à la plus à droite.
* **Région**: ensemble des 9 cases qui se trouvent dans le même carré qu’une case. Il existe en tout 9 régions qui sont numérotés de 1 à 9, selon le sens de lecture de gauche à droite d’abord puis de haut en bas.
* **Unité**: colonne, ligne, ou région d’une case.
* ***XXX donné***: valeur d’entrée pour l’algorithme. (chiffre, numéro de colonne/ligne/région)
* ***Chiffre inscrit dans une case***: valeur de la case correspondant à celle de la solution (de l’étudiant, du sudoku réel) (ie : valeur définitive)(ie : valeur inscrite au stylo)(nb : autre vocabulaire possible ; chiffre résolu, chiffre validé…)
* ***Chiffre valable selon la règle YYY***: selon la règle YYY (ligne/colonne/région) qui est appliqué, le chiffre peut être un candidat pour être un chiffre hypothétique.
* ***Chiffre hypothétique***: peut être candidat pour être un chiffre inscrit.

Règles de bases : (principe de base du sudoku)

1. **Règle de la colonne**

Pour un chiffre donné, pour une case donnée,

Si pour chaque case de la colonne de la case donnée, le chiffre donné n’est pas inscrit dans la case,

Alors le chiffre donné est un chiffre valable selon la règle de la colonne pour la case donnée.

1. **Règle de la ligne**

Pour un chiffre donné, pour une case donnée,

Si pour chaque case de la ligne de la case donnée, le chiffre donné n’est pas inscrit dans la case,

Alors le chiffre donné est un chiffre valable selon la règle de la ligne pour la case donnée.

1. **Règle de la région**

Pour un chiffre donné, pour une case donnée,

Si pour chaque case de la région de la case donnée, le chiffre donné n’est pas inscrit dans la case,

Alors le chiffre donné est un chiffre valable selon la règle de la région pour la case donnée.

Stratégies de bases (niveau facile)

1. **Stratégie de l’hypothèse**

Pour un chiffre donné, pour une case donné,

Si pour la case donnée, le chiffre donné est valable selon la règle de la colonne, selon la règle de la ligne et selon la règle de la région,

Alors le chiffre donné est un chiffre hypothétique.

1. **Stratégie de l’hypothèse seule explicite (anglais : naked single = célibataire nu)**

Pour une case donnée,

Si la case donnée ne contient qu’un seul chiffre hypothétique,

Alors le chiffre hypothétique est un chiffre inscrit.

Stratégies avancées (niveau normal voire difficile)

1. **Stratégie de l’hypothèse seule implicite (anglais : hidden single = célibataire caché)**

Pour une unité donnée, pour un chiffre donnée

Si le chiffre donné est un chiffre hypothétique pour une et seule case de l’unité donnée,

Alors le chiffre hypothétique est un chiffre inscrit.

1. **Stratégie des l’hypothèses couplés explicite (anglais : naked pair = pair nu)**

Commençons déjà avec celle du dessus…

On utilise Jess mettre à jours les possibilités et tester la violation des règles

Fonctionnalités :

* A partir de l’état actuel de la matrice, il nous renvoie une liste de coups possibles en appliquant la stratégie 1 (on verra pour la 2 après),

*Il faut voir si tu préfères stocker l’état de la matrice en interne ou si tu veux qu’on te la passe en argument*

* Faut que tu nous dise le type de retour que tu veux renvoyer (celui qui te facilite la vie, on fera avec ;) )
* A partir d’une action faite par l’utilisateur (un chiffre est rentré), Jess nous informe si la réponse viole les règles,
  + Si oui, c’est tout, Java fera le reste,
  + Si non bah c’est tout aussi le joueur a joué aléatoirement mais n’a pas faux (au tuteur de prendre le relais,

*Pareil ici à toi de voir si on te passe juste l’action faite par l’utilisateur pour mettre ta matrice interne à jour ou si tu prèfère qu’on te passe la nouvelle matrice potentiel en argument*

* Ici, renvoie juste un bool

# Apprenant (Ludovic)

Dépend des stratégies mais voilà les fonctions prévues :

* Stockage de la liste des connaissances,
* Stockage de l’avancement dans l’apprentissage de chacune d’entre elles,
* Stockage de l’utilisation de l’aléatoire.
* Stockage de la maitrise des stratégies.

# Boucle de jeu

Voilà comment se déroule une boucle de jeu :

* Au joueur de jouer,
  + S’il met trop de temps on envoie un message à l’aide du tuteur,
* Le joueur modifie ces hypothèses
  + On verra si on a le temps de le prendre en compte,
* Le joueur modifie le chiffre d’une case
  + On test si l’action appartient au actions de la stratégie 1 ou 2,
    - Si oui, on met à jour :
      * la matrice de l’état actuel du sudoku,
      * l’affichage,
      * L’apprentissage des connaissances (en fonction de la stratégie utilisé)
    - Si non,
      * On teste avec Jess
        + s’il a fait une erreur

Le tuteur lui dit son erreur,

On met à jour l’apprentissage des connaissances (en fonction de la règle violé)

On empêche la mise à jour de l’affichage ?

* + - * + S’il n’a pas faux

Le tuteur lui dit qu’il a pas besoin de répondre au hasard,

On met à jour l’utilisation du hasard (apprentissage)

On l’autorise à continuer ?

* On met à jour la liste des coups possibles par stratégie,
* On attend que le joueur joue.

Jules : Je suis d’accord pour ton déroulement sauf que les stratégies 1 et 2 (du doc Note.docx que tu as fait Ludo, les stratégie qu’on avait initialement prévu puis qu’on change finalement, cf. ce que j’ai rédigé) font partie de la stratégie de l’hypothèse seule explicite (deuxième point des strats de bases). Je pense qu’il serait pas mal de prendre en compte les hypothèses (si possible techniquement) car je pense que c’est difficile d’expliquer le processus cognitif humain du feeling que constitue les stratégie 1 et 2 initiales, et en plus de percevoir les connaissances de l’apprenant quoique c’est possible mais c’est pas mal d’inférence et assez limité comme perception.

En fait le joueur peut remplir une case au feeling uniquement s’il applique la stratégie de base numéro 2. Sans hypothèse c’est possible.

Appliquer la stratégie normale 1 au feeling ça me semble assez chaud – je dis ça je suis novice.

Puis comme je disais, à la première connexion lui demander s’il connaît les règles, si oui ok (on verra bien) si non on lui explique en brut.

Pour ma part, je pense qu’il serait intéressant de laisser à l’utilisateur la possibilité de soit remplir une hypothèse soit remplir un chiffre.

Si l’hypothèse est correct, Ok, cela correspond à la strat de base 1 puis au règles de bases.

Si l’hypothèse n’est pas correct, lui avertir s’il la laisse comme ça au bout d’un certains temps, car cela serait de l’inattention (sur le jeu moi ça m’arrive de placer l’hypothèse puis de l’éliminer quelques seconde plus tard car je n’avais pas encore tout checké : d’abord une région, puis une ligne, puis une colonne)

Si l’utilisateur fait plein d’erreur d’hypothèse, lui expliquer les règles qu’il n’a donc pas saisies ! (« t’es un noob ma parole ? »)

Si le chiffre est correct, il faut checker si cela correspond à la strat de base 2. (seul chiffre possible)

Si cela correspond bien à la strat de base 2, le féliciter (« mais quelle vision !! »)

Si cela ne correspond pas à la strat de base 2 (et que c’est correct), lui demander s’il a fait ça au hasard (ça me semble difficile de faire ça sans les hypothèses, à voir avec un vrai pro n’est-ce-pas Gilles !!)

Si ce n’est pas au hasard, le féliciter (« oh quelle intuition !! »)(nb : l’utilisateur pourrait alors exploiter la faille de tout remplir au hasard et de dire non c’n’est pas du hasard jusqu’au message de félicitation…)

Si c’est du hasard, on l’engueule (« commence par remplir les hypothèses tête d’œuf !! ») mais on ne remplis pas la case.

Si le chiffre n’est pas correct, il faut checker pourquoi

Si cela correspond à une violation des règles de bases selon l’état du plateau, on lui montre gentiment. (« tu louches, ma foi ? »)

Si cela correspond à autre chose, demander s’il a fait au hasard

Si c’est du hasard (« commence par remplir…)

Si ce n’est pas du hasard, soit on le laisse dans sa merde jusqu’à une contradiction, soit on l’engueule.

Ça fait pas mal de boulot déjà…

# **Architecture logicielle**

Fonctionnement principale : jouer un coup

1. Le module Expert lit la grille actuelle et obtient la liste des coups jouables selon la stratégie 1 ou 2.
2. L’utilisateur joue un coup sur la grille actuelle à l’aide du module Interface.
3. Le module Apprenant évolue selon la nature du coup joué en fonction des stratégies.
4. Le module Tuteur intervient selon l’évolution de l’apprenant.
5. La grille actuelle est mise à jour selon le coup joué.

Fonctionnement secondaire : éliminer des hypothèses

1. Le module Expert lit la grille actuelle et obtient la liste des hypothèses éliminables selon les différentes règles.
2. L’utilisateur élimine plusieurs hypothèses à l’aide du module Interface.
3. Le module Apprenant évolue selon la nature des hypothèses éliminées.
4. Le module Tuteur intervient au bout d’un certain moment.

Module Tuteur : fonctionnement

1. Première connexion de l’utilisateur
   1. Introduction du logiciel et de ses auteurs.
   2. Introduction du jeu du Sudoku.
   3. Explication des règles de bases du jeu de Sudoku.
2. Lorsque l’utilisateur élimine une hypothèse
   1. Pas d’intervention.
   2. Si l’apprenant a éliminé plus de X hypothèse :
      1. Définition de la stratégie 1.
      2. Définition de la stratégie 2.
3. Lorsque l’utilisateur joue un coup
   1. Si le coup joué est un coup jouable selon la stratégie 1 ou 2: Félicitation de l’apprenant en citant la stratégie
   2. Si le coup joué n’est pas un coup jouable selon la stratégie 1 ou 2 : Avertissement à l’apprenant.