

I. Guide de projet AI12 : Application de jeu de stratégie Nomad en réseau

1. Jeu de plateau Nomad

Informations

- **Licence** : Creative Commons CC-BY
- **Auteur** : Kirk Mitchell
- **Vidéo de présentation** : <https://youtu.be/8sWKO99TegA>

Pièces et plateau

- Le plateau consiste en une grille de 13 x 13 cases
- Il y a 5 pièces **Tour**
- Il y a 165 tuiles, rouge d'un côté et blanc de l'autre

Règles du jeu

- Le plateau est vide au début du jeu.
- Deux joueurs placent alternativement leurs pièces sur le plateau. Un joueur place les pièces du côté rouge, l'autre du côté blanc.
- Le joueur rouge joue en premier.
- Le premier coup consiste à placer une tour sur le plateau.
- Les tours peuvent être posées sur n'importe quelle case non occupée du plateau, mais éloignées d'au moins trois cases des autres tours.
- Quand toutes les tours de la partie ont été placées (de 2 à 5), les tuiles sont jouées.
- Une tuile peut être jouée sur n'importe quelle case vide du plateau.
- Une tuile peut être placée sur une case contenant une pile de tuiles (quelque soit la couleur de la pile) si le joueur possède une pile de tuile adjacente au moins aussi grande que la pile ciblée.
- Un joueur possède une pile de tuile si la tuile la plus haute est de sa couleur.
- Deux cases adjacentes sont des cases avec un côté commun (pas d'adjacence diagonale).
- Une tuile ne peut pas être jouée sur une Tour.
- Un joueur peut passer son tour de jeu.
- Une tour est dite connectée pour un joueur s'il y a un chemin ininterrompu de tuiles ou de piles de tuiles adjacentes de la couleur du joueur de la tour à une autre tour.
- Quand on compte les connexions, on compte le nombre de tours connectées, pas le nombre de chemins entre les tours.

Fin d'une partie

- Le jeu se termine quand un joueur a connecté toutes les tours, et que l'autre joueur ne peut pas interrompre une connexion par un simple coup.
- Le jeu se termine quand toutes les tuiles ont été jouées. Le joueur avec le plus de tour connectées a gagné.
- Le jeu se termine quand les deux joueurs passent successivement leurs tours de jeu. Le joueur avec le plus de tour connectées a gagné.

2. Fonctions du logiciel

Créer un profil

- **Déroulement** : L'utilisateur choisit un pseudo et un mot de passe, créant un profil local sur sa machine. Des données personnelles optionnelles (nom, prénom, âge, avatar, etc.) sont également demandées.
- **Remarque** : Attention, rien n'empêche a priori deux utilisateurs d'avoir le même pseudo (par exemple avec une application client/serveur en créant un compte de même pseudo sur deux serveurs différents, puis l'un des deux se connectant par la suite au serveur de l'autre).

Connexion et découverte d'utilisateurs

- **Déroulement** : L'utilisateur se connecte à l'application par son pseudo et mot de passe. L'application affiche la liste des utilisateurs distants actuellement connectés, et la liste des parties en attente ou en cours.
- **Piste de travail** : L'application peut être de type client/serveur ou de type client décentralisé. Attention à bien étudier les **nombreux** avantages et défauts de chacun avant de choisir (l'argumentaire du choix fera partie du dossier de conception). Pour un type décentralisé, la découverte d'autres applications se fait à travers des noeuds initiaux dont les adresses sont spécifiées par l'utilisateur. Pour un type client/serveur, l'adresse d'un serveur est supposée connue. Plusieurs serveurs de jeu différents peuvent fonctionner en parallèle sans se connaître, et un utilisateur peut changer de serveur de jeu sans perdre ses données personnelles.

Créer une partie

- **Déroulement** : L'utilisateur peut créer une partie de jeu. Il spécifie le nombre de tour à placer pour la partie, donne un nom à la partie, et elle est créée. Les joueurs peuvent toujours *chatter*.
- **Fonctionnement facultatif**: Le créateur d'une partie décide lors de sa création si la partie accepte ou non des spectateurs, et si les spectateurs ont le droit de chatter.

Jouer une partie

- **Déroulement** : La partie suit les règles décrites précédemment. Le créateur de la partie peut arrêter la partie à tout moment.

Voir le profil d'autres utilisateurs

- **Déroulement** : L'utilisateur peut à tout moment voir le profil d'autres utilisateurs, son nombre de parties jouées, et ses résultats (parties perdues, gagnées, ou abandonnées).

Exporter et importer un profil

- **Déroulement** : L'utilisateur peut exporter son profil local pour l'importer et jouer à partir d'une autre machine cliente.

3. Moyens

Environnement informatique

- Gestion de projet :
 - Ganttproject
 - Microsoft Office Project

- dotProject
- gestion d'issues Github
- kanban picasoft (type trello) : <https://kanban.picasoft.net/>
- UML
 - PlantUML (recommandé) : plantuml.com, planttext.com
 - Dia (outil **très** basique)
- Environnement de programmation
 - Netbeans
 - Eclipse
 - sous Java 1.8
- Gestion de version :
 - Gitlab utc (recommandé)
 - SVN (sur tuxa.sme.utc)
 - plugin SVN, Git et CVS sous Netbeans
 - outil Tortoise SVN
 - SVN sur Eclipse ?
- Autres logiciels de l'UTC:
 - Microsoft Office,
 - owncloud : répertoire partagé pour l'échange de documents, **pas** la gestion de version
 - etc.

Autres logiciels (non fournis par l'UTC)

- Outils UML libres utilisables sans installation : White Star UML, Websequence diagrams (outil en ligne), Modelio, ArgoUML (léger), Topcased, etc.
- SVN repository gratuit (si tuxa rame ou plante) : Google Code, Unfuddle, OpenSVN
- Git : github
- Logiciel de gestion de projet libre : dotproject, etc.
- Outils de qualimétrie : Sonar (sonarqube, sonarlint), XRadat, etc.

Echange d'information

- Discussions pendant les séances (formelles / informelles)
- SVN, CVS, Git (code source et documentation)
- Mattermost : https://team.picasoft.net/signup_user_complete/?id=wctr7zo3tj8kueuym6adcfgiqc
- Moodle : forum étudiant
- Autres forums ?
- Mail
- autres possibilités : slac, discord, google groups, facebook, etc.

Tutoriels et docs

Qwant/DuckDuckGo est votre ami !

- La JAVAdoc : INDISPENSABLE
 - en anglais (Java 1.8) : <https://docs.oracle.com/javase/8/docs/api/>
- JAVA
 - en anglais : *tutoriel oracle* <http://download.oracle.com/javase/tutorial/>
 - en français : <http://jmdoudoux.developpez.com/cours/developpons/java/>
- Points particuliers :
 - Serialization (pour transporter des objets JAVA à travers le réseau)
 - en anglais : <http://www.oracle.com/technetwork/articles/java/javaserial-1536170.html>

- en français : <http://ydisanto.developpez.com/tutoriels/j2se/serialisation/partie1/>
- IHM en JavaFX (en anglais)
 - http://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm
 - <https://code.makery.ch/fr/library/javafx-tutorial/>

4. Déroulement du projet

Découpage structurel imposé

Quatre sous-projets :

1. Communication : Ce module est responsable de l'acheminement des données dans le logiciel, et de l'envoi et de la réception des messages réseaux. Si le choix d'une architecture serveur est faite, il devra aussi implémenter l'IHM serveur (modification du port de connexion, par exemple).
2. Gestion et traitement des données : Ce module comprend les structures de données liées au profil utilisateur, et aux parties de jeu. Il est responsable de leur chargement et de leur sauvegarde et de l'implémentation des règles du jeu.
3. IHM-Main : Ce module gère l'interface graphique principale de l'application : connexion, liste des utilisateurs et des parties, création de partie, profil utilisateurs, etc.
4. IHM-Game : Ce module gère l'interface graphique pour jouer une partie, ainsi que le chat de chaque partie.

Organisation

Six responsables pour chaque équipe :

1. Directeur de projet : responsable communication avec le client et les autres équipes de sous-projet
2. Manageur de projet : responsable gestion de projet (estimation des charges, planification, mise à jour des indicateurs)
3. Concepteur : responsable de la conception
4. Développeur : responsable du développement
5. Chef qualité : responsable de la qualité (documentation, tests, qualité du code, fautes d'orthographe, etc.)
6. Etude : responsable des solutions techniques (y compris leur prototypage)

Possibilité de changer de rôle en cours de semestre par demande justifiée.

Tout le monde participe à toutes les tâches !

(mais c'est la faute du responsable si ça ne fonctionne pas)

Délivrables

- Plan de management de projet
 - a. Découpage du projet
 - b. Estimation des charges
 - c. Planification
 - d. Déroulement du projet (problème, reprise des étapes d'analyse, nouvelle planification, etc.)
 - e. Analyse de risque
 - f. Actions qualité
 - g. Réflexion sur l'ingénierie soutenable du logiciel (pour les UTCéens FISE)
 - h. Pour chaque membre du module : fiche de travail individuelle indiquant précisément les tâches effectuées, le temps passés sur chacune, le temps initial prévu, et quelques commentaires (facilité de réalisation, problèmes rencontrés, etc.).
 - i. Auto-notation de chaque sous-projet (12 points par membre à répartir comme entre les membres du sous-projet, sauf soi-même).
 - j. Bilan

- Dossier de conception
 - a. Choix de conception : question, solutions possibles, argumentation et décision
 - b. Diagramme des cas d'utilisation
 - c. Diagrammes de séquences (pas forcément tous)
 - d. Interfaces
 - e. Diagramme des classes
- Dossier de réalisation
 - a. Choix technologique : question, solutions possibles, argumentation et décision
 - b. Points durs (difficultés particulières rencontrées)
 - c. Points particuliers (exemples d'algorithme, approfondissements possibles, etc.)
- Code source et exécutable (y compris les bibliothèques nécessaires, s'il y en a, la documentation et les manuels utilisateurs)

Echéances

- Pour le vendredi 8 octobre : Dossier de conception, et première version du plan de management de projet
- Pour le vendredi 7 janvier : Plan de management de projet (version finale), dossier de réalisation, code source et exécutable

Plan possible pour la soutenance (première semaine de janvier)

- Organisation générale (10mn) :
organisation et déroulement globaux du projet, cas d'utilisation et classes générales de l'application
- Présentation de chaque sous-groupe (30mn) :
 - présentation de chaque responsable (5mn)
 - question (10mn)
Attention : une question peut être posée à n'importe quelle personne sur n'importe quel sujet !
- Démonstration de l'application (20mn)

Longue durée : environ 3 heures = le résultat de 1000 heures de travail (voire plus)