

Dossier de spécification des besoins logiciels

*MAHÉ François
POMERET-COQUOT Pierre*

Jeudi 9 mars 2017

DOSSIER DE SPÉCIFICATION DES BESOINS

LOGICIELS

BUT DU DOCUMENT

Le présent document tient lieu de Cahier des Charges. Il a été établi afin de lister les fonctionnalités attendues du logiciel, et ainsi assurer une parfaite adéquation entre les attentes du client et le travail fourni par l'équipe de développement

DESCRIPTION GLOBALE

L'équipe de développement s'engage à fournir au client un ensemble logiciel constituant le noyau fonctionnel (cœur de calcul) d'une application permettant de jouer à *Hex*. L'interface de cette application sera développée ultérieurement par le client, et dialoguera avec l'ensemble logiciel fourni.

L'ensemble logiciel développé est composé de modules :

- Bibliothèque native pour Java (JNI) permettant la manipulation d'une partie de Hex
- Bibliothèques natives pour Java (JNI) proposant des intelligences artificielles capable de jouer à Hex
- Démonstrateur (langage Java) manipulant ces bibliothèques, et proposant une interface minimale, en ligne de commande.

Le code source de ces bibliothèque restera propriété de l'équipe de développement, et leur utilisation conditionnée par un bail de un an, comme défini par le contrat. Le code du démonstrateur sera offert au client.

La garantie et le support seront assurés pendant toute la durée de ce bail.

ENVIRONNEMENT

L'ensemble logiciel fourni pourra être compilé sur toute machine (PC, Mac, UNIX)

Le démonstrateur (interface en ligne de commande) sera adapté à un ordinateur de bureau (muni d'un clavier et d'un écran de taille raisonnable). La recette (validation finale) se fera chez le client, sur une machine Unix du bâtiment U1.

L'intégration de ces bibliothèques au sein de l'interface développée par le client, et leur compilation, restent à la charge du client.

PROFILS DES UTILISATEURS

Les utilisateurs sont réputés compétents en informatique et capables d'utiliser une bibliothèque tiers d'après sa documentation.

Les utilisateurs finaux (clients du client) appartiennent à un panel très varié, et il revient au client d'adapter son interface à leurs caractéristiques.

SPÉCIFICATIONS GÉNÉRALE

DESCRIPTION DES SERVICES ATTENDUS

La gestion de parties (création, placement des pions, etc.) est assurée par la bibliothèque **cretinhex**

Les intelligences artificielles sont assurées par les bibliothèques **cretinhex_mmbot** (Algorithme du minimax) et **cretinhex_bridgebot** (détection de ponts)

Les facilités de jeu (interaction avec l'utilisateur) sont assurées par le démonstrateur **playhex** qui dialogue avec les bibliothèques.

La bibliothèque **cretinhex** et le démonstrateur **playhex** sont considérés comme obligatoires et l'équipe de développement s'engage à fournir leurs versions finales à la date de réception. Tout défaut de sa part entraînera les pénalités prévues par le contrat.

Les intelligences artificielles sont considérés comme souhaitées, sans qu'aucun engagement n'ait été pris par l'équipe de développement. Si ces bibliothèques sont réalisées lors de la livraison, elles seront fournies, et réglées par le client comme exprimé sur le devis. Le cas échéant, une intelligence « naïve » sera fournie (pour illustrer son implémentation), sans aucune conséquence sur le reste du travail fourni.

Une documentation complète des bibliothèques sera fournie lors de la livraison.

DESCRIPTION GÉNÉRALE DES FONCTIONS

Manipulation d'une partie de Hex (bibliothèque native **cretinhex**) :

- Création de parties : plateau de taille libre et choix du premier joueur
- Obtention des informations suivantes :
 - Numéro de tour
 - Joueur dont c'est le tour
 - L'occupation des cases
 - L'état de la partie (en cours, gagnée par J1, gagnée par J2)
- Historique : mémorisation, consultation et annulation des mouvements effectués par les joueurs
- Sauvegarde : écriture et lecture de fichiers textes au format imposé par le client : un fichier permettant d'enregistrer ou de charger une partie de Hex

Joueurs non humains : bibliothèques dédiées (une bibliothèque par IA)

- Respectant toutes la même interface (et proposant les mêmes services)
- Implémentant une stratégie Minimax (**cretinhex_mmbot**) ou par détection de ponts (**cretinhex_bridgebot**)
- Permettant le remplacement et/ou l'ajout de nouvelles intelligences artificielles

Démonstrateur **cretinplay** : application Java implémentant les fonctionnalités que le client désire intégrer à son interface, pour valider et illustrer les fonctionnalités attendues

- Interaction avec l'utilisateur :
 - Système de menus permettant la réalisation de toutes les actions citées
 - Affichage du plateau et des jetons placés
 - Affichage des informations du tour (numéro, joueur)
 - Saisies au clavier (placement de pion, taille de plateau, etc.)
- Communication avec la bibliothèque :
 - Création de parties, obtention d'informations
 - Consultation de l'historique, annulation du dernier coup
 - Chargement et sauvegarde de partie dans des fichiers
- Compétences autonomes :
 - Vérification de la validité des coups joués
 - Abandon de partie
 - Jeu Humain vs Humain ou Humain vs CPU

Le client souhaite pouvoir gérer une liste des parties grâce à une base de donnée. L'équipe de développement lui propose la possibilité de nommer à sa guise les fichiers de sauvegarde, et ainsi les associer facilement aux entrées de sa base de donnée.

EXIGENCES OPÉRATIONNELLES

CONTRAINTES D'EXPLOITATION

Les bibliothèques doivent pouvoir être compilées sur les systèmes courants (Windows, Mac, Unix)

MODES DE FONCTIONNEMENT

Utilisation des bibliothèques par appels de leurs fonctions

Utilisation du démonstrateur au clavier, dans un terminal

CAPACITÉS

La bibliothèque **cretinhex** peut gérer toute taille de grilles (la seule limite étant la mémoire de la machine exécutant le programme).

Les intelligences artificielles ne pourront traiter dans un temps raisonnable que des grilles petites (minimax) ou moyennes (bridge).

L'affichage de ces grilles dans le démonstrateur est limité par la taille de l'écran. Une grille de 20x20 pourra être affichée sur un écran standard.

PERFORMANCES

Pour une grille 11x11 :

- Le démonstrateur et les appels à la bibliothèque **cretinhex** seront quasi instantanées
- Les appels à la bibliothèque **cretinhex_bridgebot** seront inférieurs à 5 secondes sur un PC grand public récent.

La bibliothèque **cretinhex_mmbot** *ne devrait pas* pouvoir travailler en 11x11 dans un temps raisonnable.

SÛRETÉ DE FONCTIONNEMENT

La bibliothèque **cretinhex** garantie un fonctionnement sans erreur par le respect de ses spécifications. Ces dernières sont vérifiées par des assertions qui peuvent être désactivées à la compilation afin d'améliorer la rapidité d'exécution.

Par ailleurs, des séries de tests seront proposées au client pour confirmer le bon fonctionnement des fonctions : des sauvegardes de grilles-types seront chargées en séries pour leur appliquer les différentes fonctions, et vérifier leurs résultats.

EXIGENCES TECHNIQUES

LOGICIELS IMPOSÉS

Les bibliothèques fournies doivent être utilisables par une application Java. Elles seront développées en C, et une surcouche permettra de les appeler avec JNI (Java Native Interface). Elles pourront ainsi être compilées en .dll ou en .so

ALGORITHMES IMPOSÉS

La bibliothèque **cretinhex_mmbot** utilisera l'algorithme du Minimax pour trouver les meilleures solutions. Elle sera adaptée aux plateaux de petite taille.

Les bibliothèques **cretinhex** et **cretinhex_bridgebot** utiliseront une représentation de graphe pour leur calculs, ce qui sera bien plus effectif.

INTERFACE AVEC LE MATÉRIEL

N/A

INTERFACE AVEC D'AUTRES LOGICIELS



N/A

ARCHITECTURE MATÉRIELLE OPÉRATIONNELLE

N/A

SCENARIO D'UTILISATION

Le client utilisera les fonctions de ces bibliothèques pour développer sa propre interface de jeu.

Le démonstrateur sera principalement utilisé pour la recette et pour illustrer l'implémentation des fonctionnalités.

