

Dossier de spécification des besoins logiciels

Nom : LYLE, MESTE, POQUILLON

Prénom : Thomas, Rémi, Mathieu

Groupe : L2InfoA4 – n°39

Version envisagée: V2

(un dépassement de version est envisageable).

DOSSIER DE SPÉCIFICATION DES BESOINS

LOGICIELS

BUT DU DOCUMENT

Le but de ce document est de lister toutes les fonctionnalités du futur logiciel et de son contexte d'utilisation (utilisateurs, autres composantes, matériel, etc.).

DESCRIPTION GLOBALE

Le projet consiste à concevoir une version numérique, interactive du jeu de société de hex ([details](#)), le jeu doit être conçu pour un usage local (un ou deux joueurs humains et possibilité d'affronter une intelligence artificielle non évoluée pour une partie à un joueur humain).

Concernant l'aspect utilisateur, il doit disposer d'une interface d'interaction (jouer / menu ...), mais il doit également proposer des fonctionnalités comme la possibilité de sauvegarder, charger des parties.

L'utilisateur doit pouvoir personnaliser la partie, modifier la taille du plateau de jeu, sélectionner le type de chaque joueurs (IA / humain), ainsi que la couleur qui commence.

Concernant l'aspect code de ce projet, l'intégration du jeu de hex doit être conçu de manière à respecter les conventions de code, une bonne lisibilité du code qui doit être commenté, et une bonne gestion et protection des structures de données.

De plus il doit être programmé de manière modulaire en divers langages qui ont un but propre a chacun :

- Le langage C va représenter la partie du traitement des informations et de la partie.
- Le JAVA va gérer les interactions avec l'utilisateur (saisie des coups, paramétrage de la partie, demande de charger / sauvegarder une partie...).
- Le SQL va gérer la gestion du stockage des données des parties sauvegarde.

ENVIRONNEMENT

Le programme doit fonctionner principalement sur l'environnement UNIX (Fedora) installé dans les salles de TP de l'Université Paul Sabatier.

PROFILS DES UTILISATEURS

Le programme du jeu de HEX doit être utilisable par des utilisateurs standards ayant des connaissances sommaires / basiques en informatique (méthodes d'interactions), sans formation spécifique à la programmation ou l'usage du logiciel.

SPÉCIFICATIONS GÉNÉRALE

DESCRIPTION DES SERVICES ATTENDUS

Le programme du jeu de HEX doit permettre principalement pour l'utilisateur de jouer une partie du jeu de HEX selon les mêmes principes que dans la vie réelle.

De plus le programme doit respecter plusieurs fonctionnalités :

- Permettre de modifier la taille de du plateau (largeur entre 10 et 20 inclus).
- Permettre de jouer contre un autre joueur humain sur le même programme (local).
- Permettre de jouer contre un joueur de type IA.
- Permettre de sauvegarder.
- Permettre de charger une partie ayant au préalable été sauvegardée.
- Gérer un affichage graphique des menus, de la partie, de la saisie des coups.

DESCRIPTION GÉNÉRALE DES FONCTIONS

- sauvegarderPartie(nomSauvegarde): Fonction qui va sauvegarder l'état et les variables de la partie dans la situation actuelle dans un fichier text de sauvegarde nommé par la variable nomSauvegarde.sav selon le modèle requis.
- chargerPartie(nomSauvegarde): Fonction qui va charger le contenu du fichier désigné par la variable nomSauvegarde afin d'extraire toutes les informations sur la situation d'une partie sauvegardée et le programme va reprendre la partie directement d'en l'état décrit par la sauvegarde.
- CréerPartie (taille): Fonction qui va initialiser une nouvelle partie avec un plateau de largeur/longueur taille.
- QuitterPartie : Cette fonction va être l'interaction lorsqu'un joueur souhaite quitter une partie en cours et non terminée, elle consiste à faire une vérification du choix de l'utilisateur quant à sa volonté d'abandonner la partie en cours. Mais elle propose également à l'utilisateur la possibilité de sauvegarde la partie en cours afin de pouvoir la reprendre ultérieurement.
- retourArriere : Fonction qui annule le dernier coup jouer sur le plateau si la partie n'est pas terminée ou que au moins un coup a été joué.
- AfficherPlateau: Fonction probablement en JAVA qui va gérer l'affichage du plateau à partir du type de donnée transmis par le C grâce à l'interfaçage de JNI.
- Jouer(couleur, coordonnée) : Fonction qui va modifier le plateau et joueur le pion de « couleur » au la position de la coordonnée si le coup est valide.

- vainqueur(plateau) : Fonction qui à partir du plateau après avoir joué un coup va déterminer s'il y a un vainqueur et mettre fin à la partie en cours.
- coupIa(plateau) : Fonction qui va générer un coup à jouer par l'ia, le coup sera plus ou moins évolué en fonction de la version.

EXIGENCES OPÉRATIONNELLES

CONTRAINTES D'EXPLOITATION

Le programme doit fonctionner sur les ordinateurs (Fedora) des salles de TP informatique de l'Université Paul Sabatier. Il doit être programmé en C et en JAVA.

Pour que le programme soit opérationnel, il doit être exécuté dans un environnement disposant d'un interpréteur de programme C compilé, ainsi que java d'installé, le programme doit également disposer des droits d'écriture et de lecture du répertoire dans lequel il est placé pour le bon fonctionnement du système de sauvegarde des parties.

MODES DE FONCTIONNEMENT

Le mode de fonctionnement du programme est divisé en fonction des modules et des langages qui chacun ont un but propre :

- Le JAVA gère l'interaction avec l'utilisateur par l'affichage graphique, il permet à l'utilisateur d'interagir sur la partie par des saisies au clavier des clics sur les boutons de l'interface.

Remarque : nous n'excluons pas l'éventualité d'implémenter une saisie des coups par des clics sur l'interface si l'avancement dans le projet est suffisant.

- Le C gère le traitement de la partie (calcul de la validité des coups, ia, recherche de gagnant ...).
- Le SQL va gérer les informations concernant les joueurs (pseudo, email et année de naissance) et les parties.

CAPACITÉS

Le programme doit satisfaire son but principal qui est de permettre à l'utilisateur de jouer une partie de jeu de hex sur un ordinateur de manière intuitive et similaire au monde réel.

L'utilisateur doit pouvoir jouer une partie contre un utilisateur humain sur la même machine, ou une partie contre une intelligence artificielle basique. Il doit pouvoir effectuer les interactions avec le programme de manière intuitives et naturelles.

Il doit être en mesure de pouvoir sauvegarder une partie en cours, si cette dernière n'est pas terminée. Ainsi lorsqu'il relance le programme il pourra recharger et reprendre une partie qu'il avait été au préalable initié puis sauvegardé.

PERFORMANCES

Pour rendre le programme le plus efficace possible, nous allons élaborer des algorithmes de manière à avoir les complexités les plus faibles, favorisant ainsi un temps de calcul minimal. De plus nous tiendrons également compte de l'espace mémoire des différents types de variables utilisés afin de réduire l'espace mémoire utilisé par le programme.

SÛRETÉ DE FONCTIONNEMENT

Par soucis de sûreté de fonctionnement, nous allons concevoir le programme de manière à concevoir un modulaire code explicite, avec des noms de variables et de fonctions précis détaillant leurs rôles. De plus le code va être commenté de manière à ce que l'algorithme soit compréhensible dans le langage naturel.

Ces règles vont être appliqués aussi bien en JAVA et en C, le maximum de scénarios de fonctionnement vont être effectués ainsi que des tests unitaires pour chaque module.

Par sécurité de fonctionnement, et afin d'éviter les conflits lors des modifications apportés par chacun des membres du groupe, nous utiliserons un système de version GIT qui permettra des retours en arrière ainsi que signaler les conflits sur les modifications d'un fichier.

EXIGENCES TECHNIQUES

LOGICIELS IMPOSÉS

Aucun logiciel, n'est imposé pour la réalisation de ce projet.

Seul les langages de programmation C, JAVA et SQL ont été imposés

ALGORITHMES IMPOSÉS

Aucun de traitement spécifique n'est imposé, mais les algorithmes doivent par soucis d'optimisation avoir la complexité la plus faible possible.

INTERFACE AVEC LE MATÉRIEL

Le programme interagit avec le matériel (ordinateur sur lequel le programme est installé) de manière conventionnelle, saisie des informations au clavier, interaction de sélection avec la souris.

Le programme interagit de manière passive avec le matériel (sans l'action de l'utilisateur) pour la création des fichiers de sauvegarde (écriture sur le disque dur), ou pour la lecture des fichiers de sauvegarde (lecture du disque dur).

INTERFACE AVEC D'AUTRES LOGICIELS

Aucun, le programme ne doit pas pouvoir interagir avec des programmes externes, il s'agit d'un programme fermé sans communications externes (ni vers un autre logiciel, ni vers le réseau).

ARCHITECTURE MATÉRIELLE OPÉRATIONNELLE

Le programme doit fonctionner sur des machine UNIX (Fedora) des salles de TP de l'Université. Mais plus généralement il nécessite d'être exécuté dans un environnement sur lequel JAVA est installé.

SCENARII D'UTILISATIONS

- 1: Lancer une partie en mode joueur contre joueur, ils jouent jusqu'à se qu'il y est un gagnant ; fin de la partie.
- 2: Lancer une partie en mode joueur contre IA, le joueur saisie le nombre de case, joue 3 tours, sauvegarde et quitte.
- 3: L'utilisateur charge une partie à 2 joueurs, et joue puis fait un retour arrière pour changer son tour de jeu.
- 4 :L'utilisateur lance une partie contre une IA puis après 3 coups il tente de jouer un coups invalide.
- 5 : L'utilisateur qui est le premier à jouer tente de faire un retour en arrière.
- 6: L'utilisateur tente de créer une partie de taille 5.