IHM : Projet

Régis WITZrwitz@unistra.fr

Introduction

Ce projet est destiné à mettre en pratique les connaissances et techniques acquises lors du cours magistral et des TP. Il peut cependant vous aussi permettre d'expérimenter avec de nouveaux thèmes et de nouveaux outils, si vous jugez ceux-ci plus appropriés à la tâche énoncée ou plus intéressants pour la carrière que vous envisagez.

Énoncé

Ce projet vise à proposer une implémentation de **Love Letter**, un jeu de cartes de Seiji Kanai pour 2 à 4 joueurs. Les règles du jeu sont décrites dans ce PDF. Vous pouvez aussi les découvrir en dans cette vidéo d'une dizaine de minutes.

Groupes

Ce projet devra être réalisé en groupes de **5 étudiants**. Essayez de composer un groupe fonctionnel, ou chacun peut accomplir sa part de travail.

Les étudiants solo ne seront pas acceptés, et ce pour deux raisons :

- Les contraintes de planning pour les soutenances et pour la correction.
- Savoir convaincre, partager les responsabilités, organiser des séances de travail communes bref, travailler en groupe, est un set de compétences professionnelles vital, qu'il vous faut développer.

Si vous avez du mal à trouver un groupe, ou si votre groupe recherche des membres supplémentaires, vous pouvez par exemple utiliser ce sujet spécialement dédié sur Moodle, ou tout autre canal de communication de votre choix.

Implémentation

L'implémentation du projet passe par deux modules obligatoires : noyau logiciel et interface graphique. Afin de vous permettre d'explorer certains domaines à

votre convenance, et d'optimiser votre note en conséquence, ce sujet propose aussi trois modules facultatifs : IA, réseau et extensions.

Les technologies utilisées sont laissées libres. Il vous faudra cependant tenir compte des contraintes suivantes :

- Tout le groupe doit être d'accord sur le bien-fondé des technologies choisies.
- Votre projet doit pouvoir être intégralement buildé, et évidemment tourner, sur les machines sises en salle de TP (T01, T02 et T03).

Noyau logiciel (Module 0)

Le projet s'appuiera sur un noyau logiciel en charge des règles du jeu.

- Les règles du jeu devront être implémentées en respectant scrupuleusement les détails énoncés dans les liens donnés dans l'énoncé.
- L'implémentation du noyau logiciel devra être indépendante et offrir des interfaces claires avec les autres modules. En particulier, aucune notion d'IHM, d'IA ou de réseau ne devra transpirer dans la conception du noyau.
- Comme vous le remarquerez en faisant vos recherches, les créateurs et la communauté des joueurs de Love Letter ont créé beaucoup de cartes supplémentaires pouvant être intégrées au jeu. Les cartes de base suivantes sont les seules à devoir obligatoirement être implémentées:
  - Garde/*Guard* : valeur 1, 5 exemplaires, fait perdre si devine
  - Prêtre/*Priest* : valeur 2, 2 exemplaires, regarde la carte
  - Baron/*Baron* : valeur 3, 2 exemplaires, compare les cartes
  - Servante/*Handmaid* : valeur 4, 2 exemplaires, protège
  - Prince/*Prince* : valeur: 5, 2 exemplaires, fait défausser
  - Roi/*King* : valeur 6, 1 exemplaire, échange les cartes
  - Comtesse/*Countess* : valeur 7, 1 exemplaire, à défausser si Roi ou Prince
  - Princesse/*Princess* : valeur 8, 1 exemplaire, perdu si défaussé

  Vous pouvez en implémenter d'autres, mais cela devra se faire via une ou plusieurs extensions.

Interface graphique (Module 1)

Comme vous le remarquerez faisant vos recherches, Love Letter est disponible en de nombreuses versions plus ou moins officielles. Chaque version « reskinne » les cartes, altérant leur nom, leur illustration et, plus rarement, leur pouvoir. De la même manière, vous êtes absolument libres de définir les détails et le *look and feel* de votre interface graphique. Gardez cependant à l'esprit les points suivants :

- Le résultat final devra démontrer au maximum les qualités d'une bonne interface homme-machine telles qu'abordées en cours.
- En particulier, votre interface devra être localisée pour au moins deux pays parlant des langues différentes.

- Le profil de la majorité d'entre vous n'est certes pas celui d'un artiste ou d'un graphiste. Cependant, cela ne doit pas vous empêcher pas de trouver des solutions techniquement à votre portée et maximisant l'attrait esthétique de votre interface.
- Si vous désirez différencier les cartes, vous pouvez reprendre les esthétiques suivantes :
  - édition originale
  - édition européenne

Intelligence Artificielle (Module 2, facultatif)

Ce module est facultatif.

Love Letter se joue par défaut de 2 à 4 joueurs. Si vous voulez permettre à un joueur unique de jouer à votre application, il vous faut donc proposer au moins une Intelligence Artificielle qui prendra la place du ou des joueurs manquants.

Cette IA peut être très bête, ou à l'inverse toujours jouer de manière optimale. Pour une meilleure expérience ludique pour le joueur humain, elle pourra aussi s'adapter à son style de jeu, afin de lui proposer un défi intéressant.

Optionnellement, il pourrait être possible de choisir parmi plusieurs IA aux styles de jeu différents.

Fonctionnalités réseau (Module 3, facultatif)

Ce module est facultatif.

Love Letter se joue par défaut de 2 à 4 joueurs. Vous pouvez choisir de permettre à plusieurs utilisateurs de lancer votre application chacun sur leur machine, et de se rejoindre pour jouer ensemble, automatiquement ou via le biais de salons ou d'un système de matchmaking, par exemple.

Le format des données échangées est laissé libre. Cependant, le protocole choisi devrait supporter toutes les fonctionnalités implémentées par votre application.

Extensions (Module 4, facultatif)

Ce module est facultatif.

Il consiste à concevoir le noyau logiciel de manière à ce qu'il puisse accepter un certain nombre d'extensions ou *plugins*. La nature des extensions n'est à priori pas définie à l'avance : un moteur d'extensions bien conçu permettra d'exposer le plus grand nombre de fonctionnalités, afin que des développeurs tiers puissent les modifier ou en ajouter de nouvelles.

Les extensions ne font pas partie du code du logiciel. À l'inverse, ils sont chargés au lancement de l'application ou dynamiquement, alors que celle-ci tourne. Par contre, toute extension doit évidemment respecter un formalisme clairement défini par vous.

Afin de démontrer l'efficacité de votre moteur, vous implémenterez une ou plusieurs extensions en explicitant leur but : nouveaux modes de jeu, ou nouvelles cartes (par exemple les nouveaux personnages de l'édition premium).

Optionnellement, vous pourrez aussi démontrer cette efficacité de manière encore plus convaincante en faisant charger par votre application une ou plusieurs extensions développés par un autre groupe. Vous préciserez l'autre groupe dans votre rapport, et cela ne sera évidemment pas considéré comme du plagiat. Par contre, il vous faudra vous mettre d'accord avec l'autre groupe sur les interfaces et structures de données utilisées. Il vous faudra aussi convenir ensemble d'une méthode de publication des extensions, via un dépôt Gitlab ou GitHub, par exemple.

Notation

Pour obtenir la moyenne, chaque groupe devra être capable de fournir :

- Un dépôt git contenant le **code source** de son application. Ce code source devra :
  - inclure tout ce qui est nécessaire pour que je puisse le builder (le cas échéant) en un seul script ;
  - permettre de produire une application tournant sans erreur sur une machine de TP ;
  - faire preuve de qualités de conception logicielle telles qu'un découpage modulaire et une bonne maintenabilité.
- Un **rapport** convaincant. Ce rapport devra, entre autres, détailler et justifier les choix ergonomiques et techniques qui auront été faits. De plus, vous devrez expliquer et justifier la manière dont vous vous êtes organisé en tant qu'équipe de développement : répartition du travail, etc.
  Le format et la maquette du rapport sont libres. *HTML*, *.pdf*, *.odt*, un seul fichier, plusieurs … votre seule contrainte est que j'arrive à le lire. En particulier, je n'ai pas de licence Microsoft™©® Word™©®, donc évitez les *.docx*.
- Une présentation convaincante de leur travail lors d'une **soutenance**. Cette soutenance pourra reprendre le contenu du rapport, et/ou développer certains points d'intérêt.
  Je vous communiquerai la durée et la date précise des soutenances dès que possible. En attendant, vous pouvez partir du principe que ces entretiens dureront 10 à 20 minutes, et que tout le groupe devra être présent.

Si vous faites tout cela correctement, vous avez 10. À ces impératifs de base se rajoutent tout un tas de domaines dans lesquels vous pouvez engranger des points supplémentaires. En voici la liste non exhaustive :

- Implémenter un ou plusieurs modules facultatifs. Consultez leur description respectives pour davantage de détails.
- Réaliser une IHM particulièrement ergonomique.
- Réaliser une IHM particulièrement attrayante (esthétique).
- Réaliser une IHM accessible aux mal voyants et/ou aux non voyants.
- Localiser votre IHM pour une culture « r2l » (par exemple arabe).
- Proposer plusieurs modes de jeu (solo, multijoueurs local ou réseau).
- Réaliser une application portable (précisez les systèmes cibles dans votre rapport, et démontrez cette portabilité lors de la soutenance).
- Protéger votre application par des tests (unitaires/fonctionnels) automatisés, utiliser un serveur d'intégration continue.

- Rédiger un rapport particulièrement détaillé et de qualité (schémas de conception, preuves de votre travail d'analyse, comparaison avec l'état de l'art, étude des alternatives, difficultés d'implémentation, solutions envisagées mais non retenues et pourquoi, etc).
- Proposer un manuel utilisateur convaincant, que ce soit au format « papier » (*ie.* PDF) ou intégré à votre application (manuel en ligne ou tutoriel).

Évidemment, tout groupe surpris à plagier le travail d'un autre sans lui accorder le crédit qui lui est dû subira les sanctions habituelles, appliquées avec la plus extrême sévérité.

Planning

- Début officiel du projet : **semaine du 5 mars**.
- La composition des groupes doit m'être communiquée par mail le **vendredi 16 mars** au plus tard. Je composerai des groupes en répartissant aléatoirement tous les étudiants ne faisant pas partie d'un groupe passé cette date.
- Livraison du code source et du rapport correspondant **dimanche 22 avril à 23:59:59** au plus tard.
- Les soutenances auront lieu lors des crénaux de TP, à partir du **mercredi 2 mai**.

Alternatives

Ce sujet ne constitue qu'une proposition par défaut. Si vous avez une idée alternative qui pourrait constituer un sujet de projet d'IHM plus adapté à vos envies ou à votre future carrière, vous en avez la possibilité.

Dans ce cas envoyez-moi, avant le **vendredi 16 mars** inclus, un document d'une ou plusieurs pages détaillant votre proposition de sujet, la liste des étudiants composant votre groupe, votre motivation pour le faire, et de quelle manière il s'inscrirait dans cet enseignement d'Interaction Homme Machine.
Je vous donnerai mon *go/no go* dans les meilleurs délais, au pire après en avoir rediscuté avec vous durant votre créneau de TP suivant la remise de votre document. Gardez à l'esprit que plus vite nous en parlerons ensemble, plus de temps vous aurez pour travailler sur votre projet.

La majorité du présent sujet reste applicable pour votre sujet de projet alternatif. En particulier, tout projet alternatif aura, dans la mesure du possible, les mêmes critères de notation que le sujet par défaut : qualité de l'application livrée, du rapport ainsi que de la soutenance, et pertinence de l'organisation de l'équipe ainsi que du code source