IHM : Projet

Régis WITZ, Nicolas LUTZ, Stephane CHOLET
– rwitz@unistra.fr, nicolas.lutz@unistra.fr, scholet@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 aussi vous permettre d’expérimenter avec de nouveaux thèmes, de nouvelles techniques et de nouveaux outils, si vous jugez ceux-ci plus appropriés à la tâche énoncée, plus intéressants pour la carrière que vous envisagez, ou simplement plus amusants.

Énoncé

Ce projet vise à proposer une implémentation de **Timebomb**, un jeu de cartes de Yusuke Sato à partir de 4 joueurs. Les règles du jeu sont décrites dans ce document.. Ce document sert de spécification à ce projet.

Vous pouvez vous faire une idée du déroulement d'une partie dans cette vidéo assez exhaustive d'une quinzaine de minutes.

Groupes

Ce projet devra être réalisé en groupes de **4 à 6 étudiants**. Chaque étudiant ne peut être membre que d'un seul groupe, mais la composition des groupes est libre.

Essayez de composer un groupe fonctionnel, où chacun peut accomplir sa part de travail, de manière adaptée à ses envies et ses capacités. Comme vous pourrez le remarquer lors de la lecture de ce sujet, une part importante de votre succès réside dans vos facultés d'analyse, de conception et de rédaction. En conséquence, n'hésitez pas à jouer sur les forces de chacun. Par exemple, si votre groupe comprend quelqu'un de moins doué pour le code mais davantage porté sur le suivi de projet et le *reporting*, il pourra être utile que cette personne se charge avec rigueur de la rédaction des rapports et du suivi des autres membres de l'équipe, quitte à n'écrire aucune ligne de code. Le principal est de maximiser la qualité de votre projet dans sa globalité (code *et* rapports), mais aussi d'en retirer un maximum de choses pour votre carrière future.

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, communiquer, 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.

Pour une communication enseignant/étudiant efficace, nous avons mis en place un slack, qui est une application de chat instantané. Libre à vous d'utiliser ou non l'application. L'idée est qu'en mettant en place ce slack public, les questions et les réponses seront visibles par tous et il sera possible de discuter ensemble de problèmes rencontrés et de solutions potentielles. Si vous avez du mal à trouver un groupe, ou si votre groupe recherche des membres supplémentaires, vous pouvez utiliser le slack, 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 libre, mais il est fortement conseillé d'utiliser Qt/qml, vos enseignants de TP étant plus confortables avec le programme et l'IDE associée. Il vous faudra tenir compte des contraintes suivantes :

- Tout le groupe doit être d'accord sur le bien-fondé des technologies choisies. Votre projet peut évidemment utiliser plusieurs technologies différentes, mais celles-ci doivent communiquer correctement, et être appropriées à leurs rôles respectifs. Vos différents choix devront être motivés dans votre rapport.
- Votre projet doit pouvoir être buildé, être lancé, et évidemment être joué sur les machines en salle de TP (T01, T02 et T03 au moins).

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 la spécification donnée dans l'énoncé. Vous pouvez proposer de nouvelles règles ou aménager les règles existantes, mais doit se faire via une ou plusieurs extensions. Les règles « de base », elles, devront toujours rester accessibles.
- L'implémentation du noyau logiciel devra être indépendante et offrir des interfaces claires avec les autres modules. En d'autres termes, votre implémentation devra respecter le principe de séparation des responsabilités : par exemple, aucune notion d'IHM, d'IA ou de réseau ne devra transpirer dans la conception du noyau.

Vous pouvez utiliser, à votre propre discrétion, les assets graphiques de cette page, contenant toutes les cartes ainsi que l'image de la boîte du jeu de 2016, sauf le verso des cartes câble. Vous pouvez vous en servir pour créer vos cartes.

Interface graphique (Module 1)

Ce projet consiste réellement en une *adaptation* des règles de Timebomb en jeu vidéo. Comme vous le savez, l'expérience d'une jeu avec des cartes en papier autour d'une table de jeu est bien différente d'une partie « sur écran ». Chaque support a ses forces et ses faiblesses. Votre but pour ce projet est de tirer le meilleur parti des forces du support informatique, de sublimer l'expérience de Timebomb par le jeu vidéo. Vous ne devez pas chercher à imiter à tout prix l'expérience originale autour d'une table, telle que celle montrée dans les vidéos données à titre d'illustration dans l'énoncé. Il vous faudra notamment faire preuve d'ingéniosité pour mettre en place un système de communication original, car le jeu est fortement basé sur le bluff, la tromperie, le jugement, et la prise de risque.

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 ergonomiques.
- 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 d'essayer de trouver des solutions techniquement à votre portée et maximisant l'attrait esthétique de votre interface. Concernant l'esthétique extérieure, vous n'êtes absolument pas obligés de reprendre le *look & feel* de l'original. Cependant, souvenez-vous que l'apparence a une influence sur l'ergonomie : le résultat doit former un « tout » cohérent.

Intelligence Artificielle (Module 2, facultatif)

Ce module est facultatif.

Time bomb se joue au minimum à 4 joueurs. Si vous voulez permettre à un joueur unique de jouer à votre application, il vous faut donc proposer des intelligences artificielles qui remplaceront 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.

Réseau (Module 3, facultatif)

Ce module est facultatif.

Timebomb se joue au minimum 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 autres fonctionnalités de 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, elles 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 lors de la conception de l'application.

Afin de démontrer l'efficacité de votre moteur, vous implémenterez une ou plusieurs extensions en explicitant leur but : nouveaux modes de jeu, nouvelles cartes (par exemple l'espion), jeu à 3/2 joueurs, plusieurs bombes…

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.
À noter que cette méthode vaudra des points supplémentaires *aux deux groupes* : celui qui a développé l'extension, ainsi que celui qui a supporté l'extension. Le décompte exacte des points de chaque groupe dépendra du travail fourni (si deux groupes conçoivent ensemble une ou plusieurs extension, aucun problème, ils seront récompensés de la même manière).

Planning

- La composition des groupes doit nous être communiquée par mail le **dimanche 6 octobre** au plus tard.
  Nous composeront les groupes restants en répartissant aléatoirement tous les étudiants ne faisant pas partie d'un groupe passé cette date.
- Votre premier rapport doit nous être rendu le **dimanche 13 octobre** au plus tard.
- Votre second rapport doit nous être rendu le **dimanche 3 novembre** au plus tard.
- Votre code source et votre troisième rapport doivent nous être rendus le **dimanche 8 décembre** au plus tard.
- Les soutenances auront lieu lors des crénaux de TP, à partir de la **semaine du 9 décembre**.

Rapports

Les différents rapports à rendre devront, entre autres, détailler et justifier les choix ergonomiques et techniques que vous aurez fait. Leur nombre (un seul document, plusieurs, …), leur formalisme (descriptif, *story-driven*, …) et leur maquette sont libres. Leur format est lui aussi libre : *HTML*, *.pdf*, *.odt*, *LaTeX*, … votre seule contrainte est que j'arrive à le lire. En particulier, je n'ai pas de licence Microsoft™ © ® Word™ © ®, donc évitez les *.docx*.

Premier rapport

Ce premier rapport devra décrire de manière claire et exhaustive la maquette de votre futur projet IHM sous forme de *wireframes*. Il devra inclure des explications textuelles pertinentes sur la manière d'accéder aux fonctionnalités de l'application, la motivation de tel ou tel choix ergonomique, et ainsi de suite.

Il est à remettre le **dimanche 13 octobre** au plus tard.

Second rapport

Ce second rapport devra décrire de manière claire et exhaustive les interfaces entre les différents modules que votre groupe a choisi d'implémenter. Il devra inclure des explications pertinentes sur la manière dont chaque module communique réellement avec les autres : services/fonctions appelés, format des données échangés, et ainsi de suite. Inutile de trop entrer dans les détails d'implémentation, cependant : seules les *interfaces* entre les modules m'intéressent, pas les détails d'implémentation propres à chaque module.

Si vous n'avez choisi de n'implémenter que le noyau logiciel et l'interface graphique, vous n'avez à priori qu'une seule interface à décrire dans ce rapport. Cependant, si vous partez sur une implémentation un peu plus recherchée (MVC, MVP, *Client/Serveur*, *Peer-to-peer*, …), il peut être utile de décrire la situation plus en profondeur.

Il est à remettre le **dimanche 3 novembre** au plus tard.

Troisième rapport

Ce troisième rapport devra décrire la manière dont vous vous êtes organisés en tant qu'équipe de développement : répartition du travail, planning et jalons, et ainsi de suite. Il sera utile aussi que vous décriviez ce que vous avez retiré du projet : réussites, difficultés, leçons apprises, experience acquise, … Et cela, tant au niveau technique qu'humain.

Si votre logiciel final diffère de ce que vous avez planifié dans votre premier rapport et/ou votre second rapport, il faudra aussi que vous discutiez de ces changements (nature, raisons, coût, …) dans votre troisième rapport.

Il est à remettre en même temps que les sources du projet, le **dimanche 8 décembre** au plus tard.

Livrable logiciel

Le **dimanche 8 décembre** au plus tard, cchaque groupe devra le remettre 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). Ces informations devront être expliquées dans un README, et au mieux être exécutées par un script de build.
- 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.

Soutenance

Chaque groupe présentera son travail lors d'une **soutenance**. Celle-ci aura lieu à partir de la **semaine du 9 décembre**.

La 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 une vingtaine de minutes, et que chaque groupe devra être présent au complet.

Là encore, essayez de voir votre soutenance comme une occasion de progresser, de vous exprimer en public, de développer vos capacités de synthèse, entre autres !

Barême

Votre logiciel devra être fonctionnel et respecter un maximum de principes ergonomiques.

À cet impératifs de base se rajoutent tout un tas de domaines dans lesquels vous pouvez engranger des points supplémentaires et obtenir une note supérieure à la moyenne. 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 et motivez vos choix dans la soutenance et le rapport.
- 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, coopératif, compétitif, 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.
- Soigner votre livraison.
- 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).
- Faire une soutenance qui claque !

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.

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 **dimanche 6 octobre** inclus, un document d'une ou deux pages détaillant brièvement 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.

Last updated 2019-09-31