

Projet commun de L2 informatique

--- JEU DE STRATEGIE COMBINATOIRE ---

C++ Stratego

Le jeu

Nous avons choisi Le Stratego, un jeu de stratégie relativement proche des échecs à la grande différence que les pièces sont disposées en début de partie selon la stratégie du joueur. De plus, le nombre de pièces par joueurs s'élève à 40 et s'opposent sans connaître la disposition des pièces adverses. L'idée vient de moi-même car étant donné que j'aime les échecs mais que nous n'avions pas le droit de prendre ce jeu, j'ai souhaité choisir un jeu de stratégie de mon enfance qui avait selon mon souvenir une bonne ressemblance avec les échecs. L'idée a été approuvée à l'unanimité à ma grande satisfaction.

Le langage

Nous avons choisi le C++ car je trouvais que l'opportunité d'un projet libre semblait idéale à l'essai d'un langage que nous ne connaissions pas, mais qui attirait mon intérêt. Le C++ s'est donc imposé comme un défi lancé à notre groupe, l'objectif étant de le découvrir et de pouvoir le comparer au C et Java, dont nous sommes déjà initiés.

Graphique ou console

Mon esprit d'initiative et mon côté téméraire m'ont fait proposer au groupe de travailler sur Qt dès le début. En effet j'ai proposé au groupe de travailler sur Qt et de s'entraîner à coder des mini-jeux à l'aide de cette GUI afin de se familiariser avec la bibliothèque et le C++ lui-même qui rappelons-le est totalement inconnu pour nous tous. C'était une erreur de stratégie de commencer par la partie « compliquée » tandis que nous n'avions même pas les bases du C++. J'assume la responsabilité de cette erreur de stratégie car au final nous avons manqué de temps avec les partiels et autres projets ce qui nous a valu de devoir repasser à un jeu console par manque de temps.

Programmation Orienté Objet

Nous avons estimé que la programmation objet faisait bon mélange entre C++ et le Stratego, car les classes étaient un bon moyen de structurer la gestion du plateau et des pièces. Nous nous sommes donc réparti des classes à implémenter, une chacun.

Cependant encore une fois par manque de temps nous n'avons pas pu tirer profit de l'héritage, notion fondamentale de la POO qui nous a fait défaut par manque d'expérience sur l'utilisation du C++.

Intelligence Artificielle

Présente parmi notre cahier des charges, nous voulions (moi tout particulièrement) implémenter une intelligence artificielle pour notre jeu afin de pouvoir jouer de manière autonome. Je me suis donc beaucoup documenté sur les différents algorithmes d'évaluations de positions, de mouvements et de probabilités. Notamment sur des algos comme minimax qui n'est pas spécialement judicieux dans le cas du Stratego, et encore une fois j'ai été trop ambitieux en me laissant envouter dans toutes les nouvelles connaissances sur les IA et algorithmes de probabilités, finalement encore une fois nous avons manqué de temps pour le code du jeu.

Répartition des tâches et difficultés

Nous avons tout d'abord mené chacun de notre côté des recherches sur Qt et les manière d'appréhender le C++. S'en ai suivi la période d'examens combinées à la réalisation d'autres projets. Nous sommes ainsi répartis les tâches à la hâte à l'approche de la date de rendu :

- Marcus : la classe Plateau + la classe Case
- Amel : la classe Piece
- Naïm : la classe Coordonnées + tentative d'appréhender Sfml à la place de Qt
- Fatou : la classe Jeu/Main
- Moi (Fabio) : la classe Joueur + tentative d'appréhender l'IA

Etant pris de cours à cause de mes erreurs de stratégies et après s'être réparti les tâches de la sorte, nous n'avons que trop peu communiqué et nous sommes retrouvé à mettre en commun nos classes 2 jours avant la date de rendu.

(d'ailleurs nous pensions avoir jusqu'au 06 je le précise non pas pour nous chercher des excuses mais pour dire qu'étant donné notre investissement sur le tard nous aurions sûrement pu rendre quelque chose de bien plus présentable en 2 jours pleins de travaux : nous avons même fait une visio-conférence de 15h de long lors de la mise en commun)

Ainsi le manque de temps, le manque de communication et surtout mon manque de stratégie sont les principales causes de notre échec sur ce projet, car en effet il a beau compilé il n'est pas fonctionnel... Ce qui est d'autant plus frustrant qu'avec toutes les recherches menées et les leçons tirées, je suis convaincu que nous pourrions faire vraiment beaucoup mieux et avec bien plus d'efficacité.

Rôle de coordinateur

Je pense qu'on peut considérer que je jouais le rôle de coordinateur car je donnais les lignes directrices sur lesquelles se pencher, répondait aux questions des membres que ce soit sur les choix d'implémentation ou bien sur des détails techniques de

compilations... Malgré ça je suis aussi celui qui a proposé de mauvaises stratégies et induit à la mauvaise gestion du temps. Je sens que j'ai énormément appris sur la gestion du temps d'une d'équipe, en plus des nouvelles connaissances assimilées.

Si c'était à refaire (et ça le sera !)

J'ai pris une décision fondamentale pour mes futurs projets de programmation : commencer par l'implémentation la plus simple et basique pour l'enrichir au fur et à mesure plutôt que partir sur un grand projet avec pleins d'idées et tenter de tout faire en même temps pour finalement échouer. De plus je pense qu'il est important de commencer cette première implémentation le plus tôt possible pour ensuite faire des recherches moins vastes et plus centrée sur les besoins du projet. Pour ce qui est de la gestion du travail d'équipe j'ai trouvé que, même si c'était sur le tard, le fait d faire des visio-conférences avec des partages d'écrans pour qu'on puisse avancer tous ensemble tout en communiquant et s'entraidant était vraiment efficace, surtout qu'une fois tout le monde sur la même longueur d'ondes grâce aux réunions de simples échanges brefs par messages deviennent suffisant pour se comprendre mutuellement.

Je compte refaire ce projet pendant les vacances avec la ferme intention de tirer partie de mes échecs et ainsi pouvoir être fier du projet réalisé. Car même si beaucoup d'enseignements sont à tirer de ce projet je ne suis évidemment pas fier la réalisation rendu, et me sens même coupable auprès du groupe.

Résolution de l'un des problèmes

Le plus gros problème technique est surtout que notre code n'est pas fonctionnel, afin de palier celui-ci, je pense qu'il est important de reprendre tout depuis le début en testant régulièrement le code à l'aide d'un main. De plus il me semble important que chaque membre du groupe ait la pleine compréhension de l'ensemble des méthodes et qu'ils puissent vérifier que chacune d'entre elles soient implémentées correctement, afin qu'il puisse les utiliser dans sa propre partie et par la même occasion exécuter un main pour tester son programme.

Retour sur l'expérience

Une expérience très enrichissante mais frustrante qui me donne envie de prendre ma revanche, je compte donc prendre en main la GUI SfmL qui m'a semblé agréable au développement de petits projets graphiques, approfondir largement le domaine des algorithmes liées à l'IA et la théorie des graphes (je pense m'être découvert une passion pour ce domaine) et enfin j'aimerais également creuser sur la mise en connexion d'un jeu avec différents protocoles car au final je n'ai pas eu le temps de me pencher de ce côté-là au cours du projet.