GESTIN

Gaëtan Semaine 26

Rapport de Projet – Projet Arcanor

1. Organisation du projet :

Dès le début nous nous sommes consacré tous les quatre à faire les diagrammes de classe et de séquence pour pouvoir partir sur une bonne base afin de bien commencer la programmation.

Puis nous avons commencé à coder le squelette de nos classes principales grâce au diagramme de classe à 4, du moins le début. Comme Mathias a décidé de ne pas poursuivre en seconde année nous avons constitué un groupe de trois avec Lucas, Gaëtan et Yohan après 1 semaine de programmation du coup on a dû rééquilibrer les tâches afin d’intégrer Lucas dans le groupe.

Ensuite, une fois le groupe formé, nous nous somme consacrés à la programmation des classes principales et d’une versions BETA de Arcanor en version Terminale.

Lors de la 1ere semaine de programmation (semaine 24), on avait convenu un rendez-vous hebdomadaire à l’IUT de 14h - 17h afin de se voir, faire l’état de l’avancement de chacun et aider les autres dans leurs difficultés. Cependant lors de la 2eme semaine et la fin, nous avons accélérer le rythme afin de finir au plus tôt pour améliorer le programme final, donc des rendez-vous de 9h – 18h, ce qui nous as permis de finir en avance et peaufiner le final.

Concernant l’organisation de notre projet elle s’est déroulé comme-cela :

- Cahier des charges et compte rendu d’analyse : Groupe de 4

- Début programmation Version Terminal : ½ Yohan ½ Gaëtan + Lucas

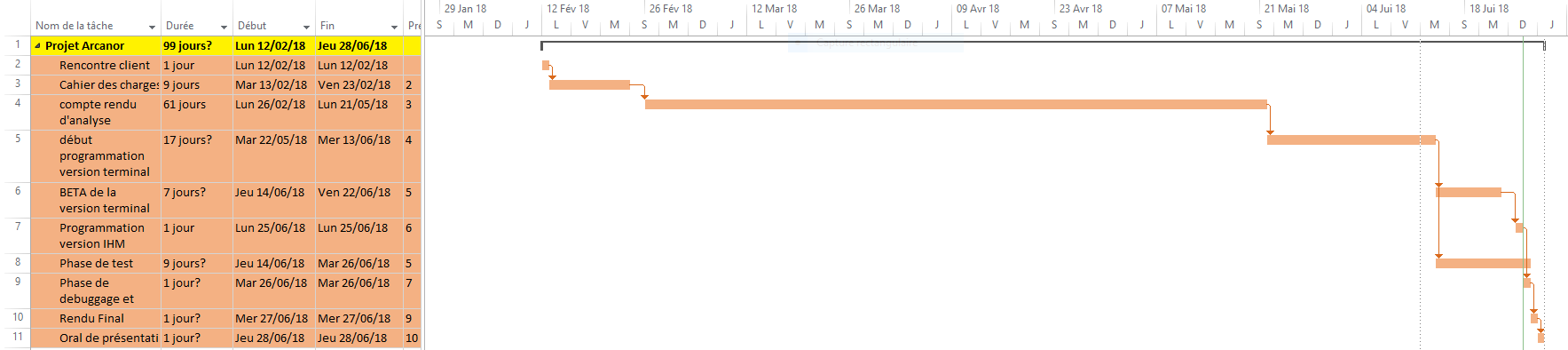
- Beta Version Terminal: Gaëtan

- Programmation Version IHM : ½ Yohan ½ Gaëtan + Lucas

- Phase de Test : Lucas

- Phase de débogage et améliorations : Gaëtan + Yohan

Nous avons respecté les dates de rendu et fixé lors du projet, nous n’avons pas eu de problèmes de temps, pas de retards.



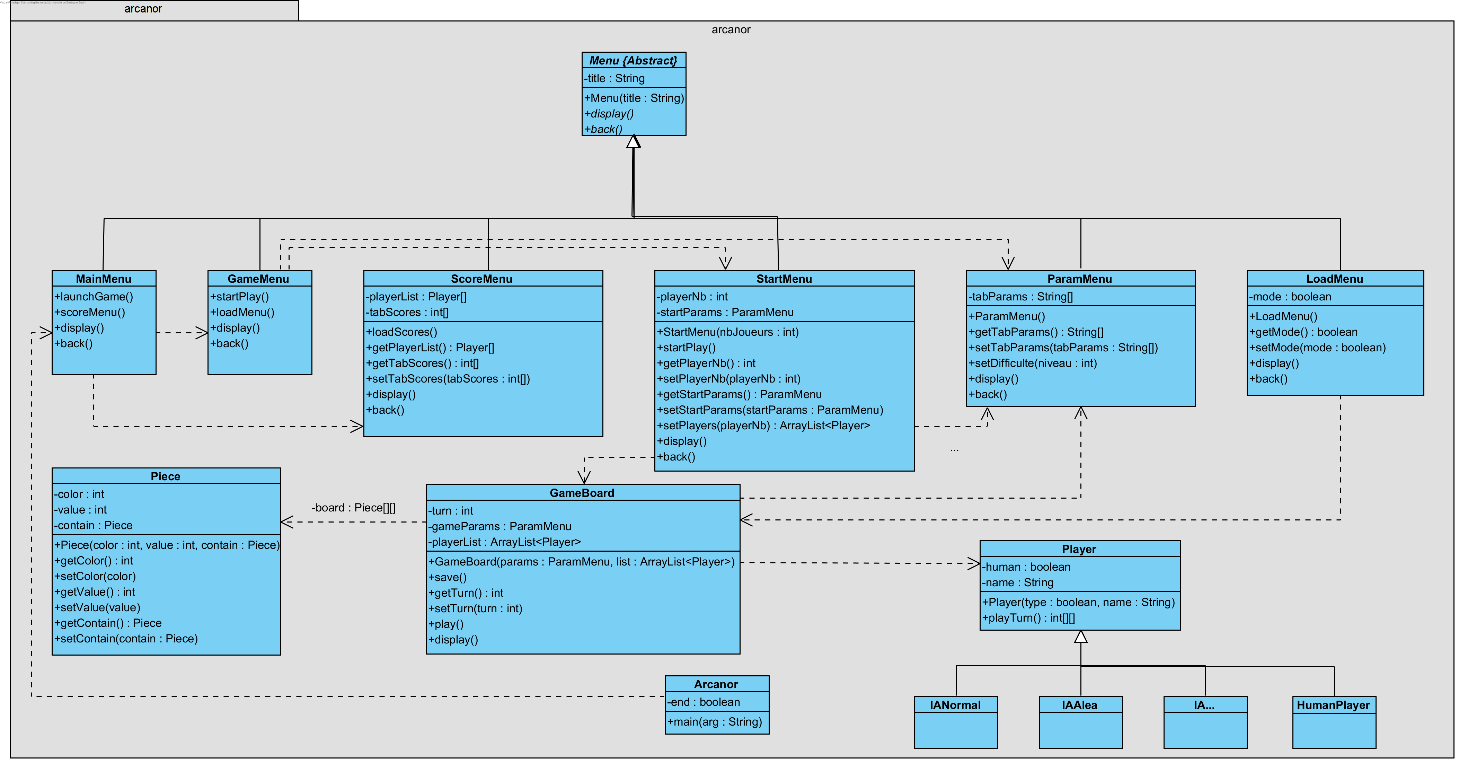
Organiser son projet est un impératif (10/10) car sans organisation un groupe se retrouvera à faire des tâches en double, ou pire en oubliant certaines, bref cela ne serait pas efficace. De plus il faut un environnement bien organisé pour coder et assembler du code provenant de sources différentes sous peine de se retrouver avec des erreurs innombrables lors de la compilation du fait de la programmation à plusieurs. Ainsi le rôle de chacun doit être défini au début du projet, chacun endossant des rôles spécifiques afin de correspondre au mieux des affinités / facilité de chacun.

La planification est elle aussi importante (9/10) car savoir estimer le temps que nous allons consacrer à une tache est très utile pour organiser ses objectifs, optimiser le temps passé sur ceux-ci et rester dans les temps l’est aussi.

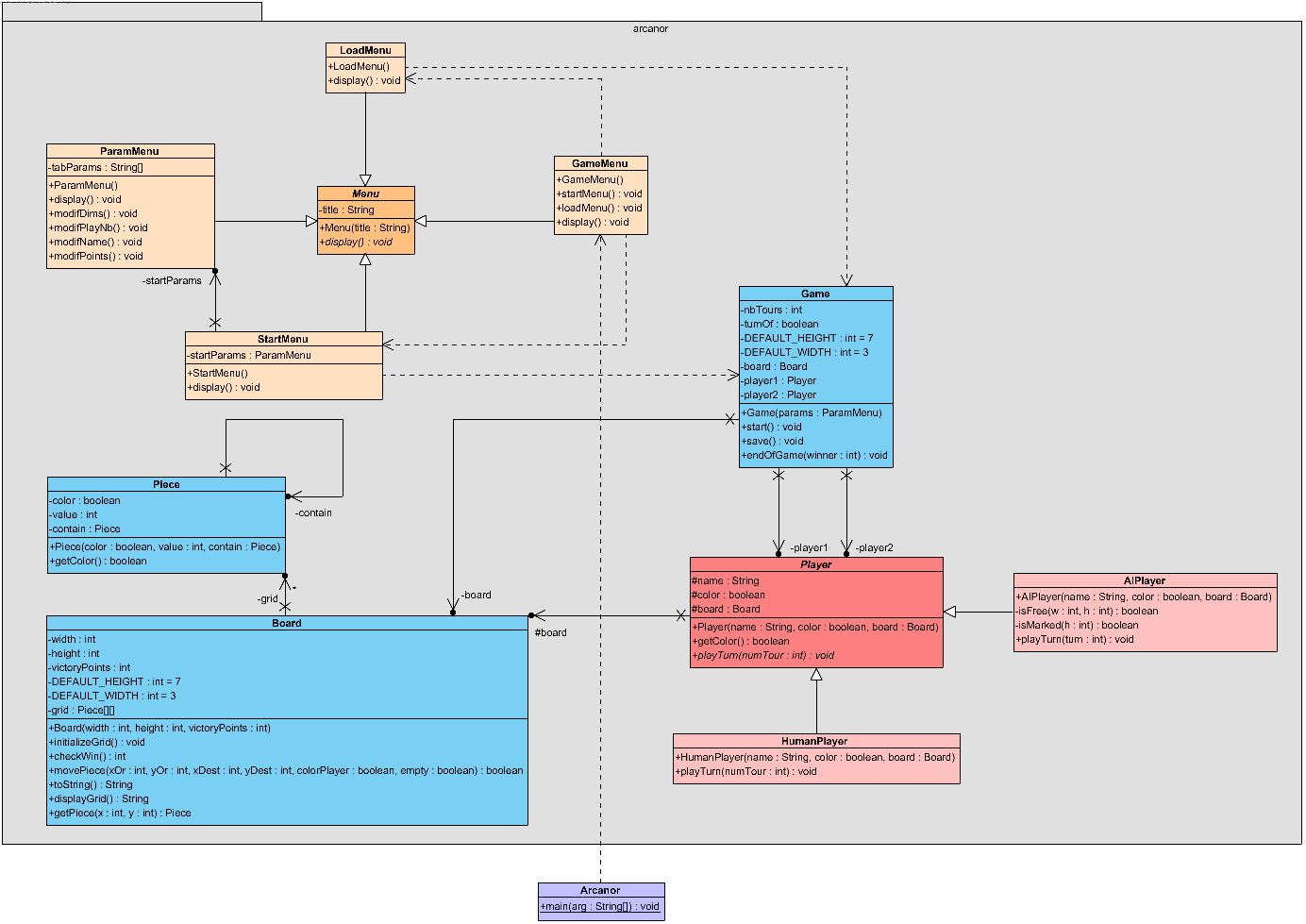
Nous pensons que le principal conseil serait de choisir un binôme avec lequel on a une bonne synergie et de ne pas prendre du retard lors des débuts du projet, pour être bien organisés lors de la 2ème partie du projet.

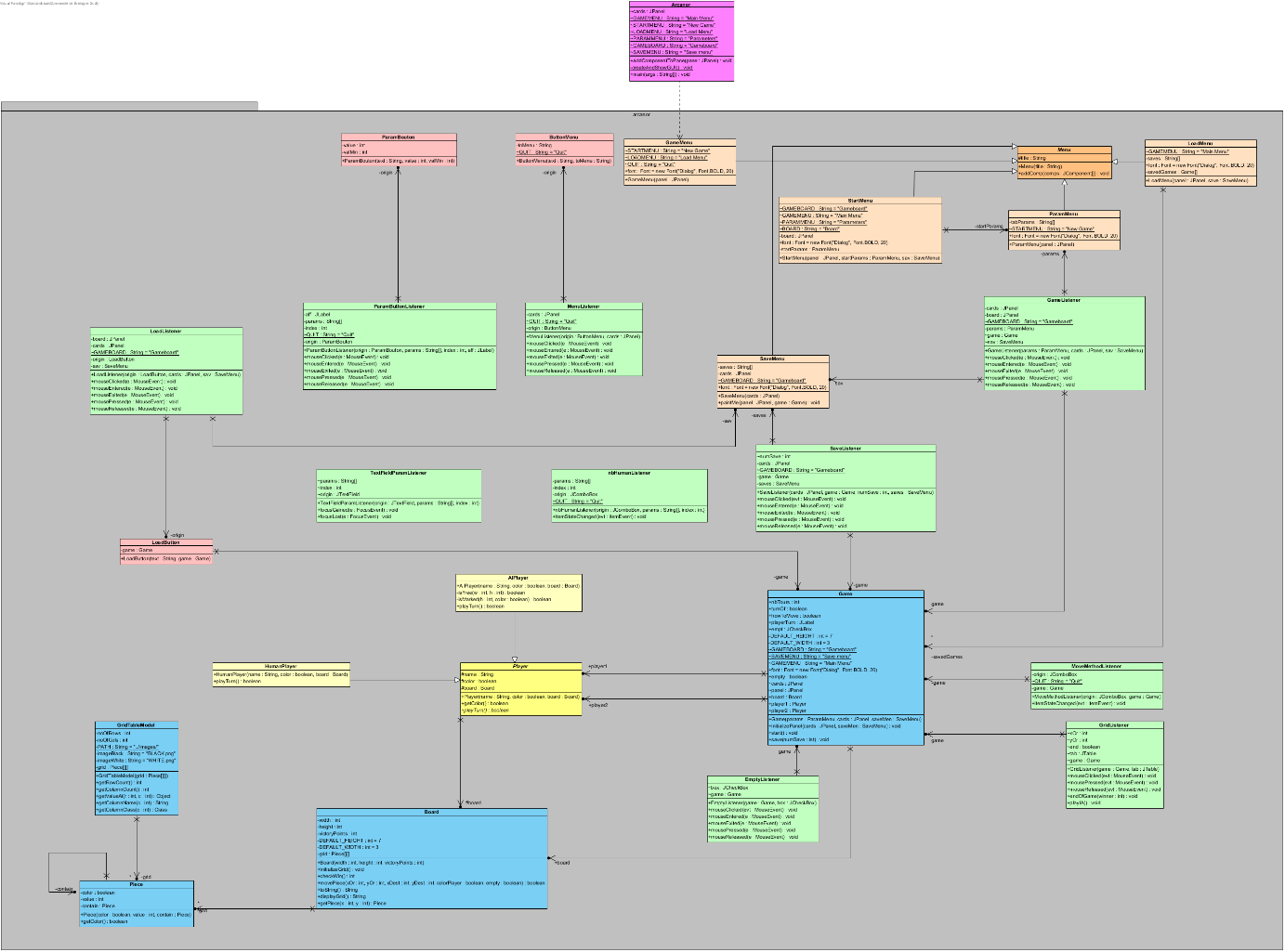
1. Diagrammes :

1er diagramme de classes :

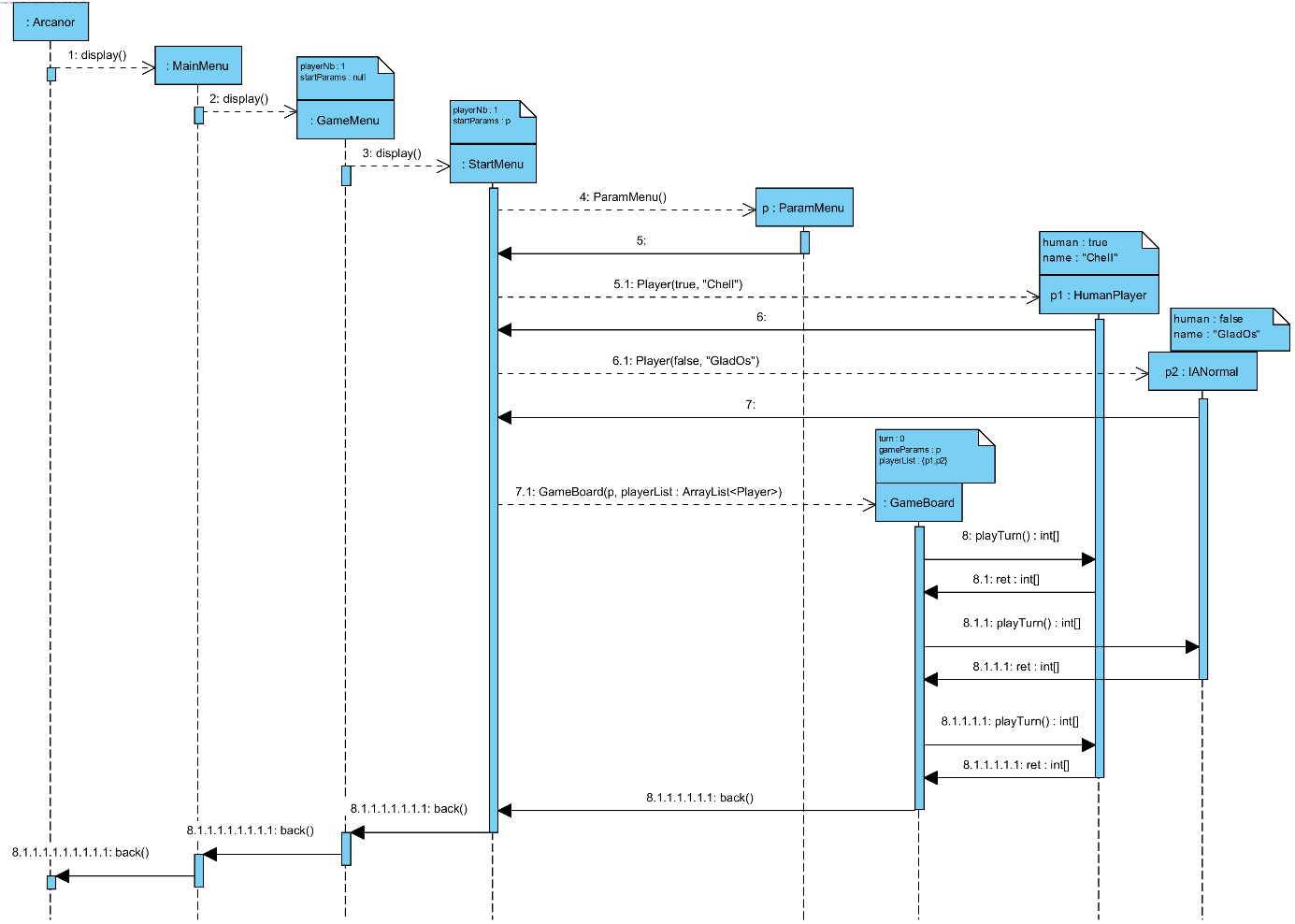


2e version du diagramme de classes (Version définitive Terminal) :

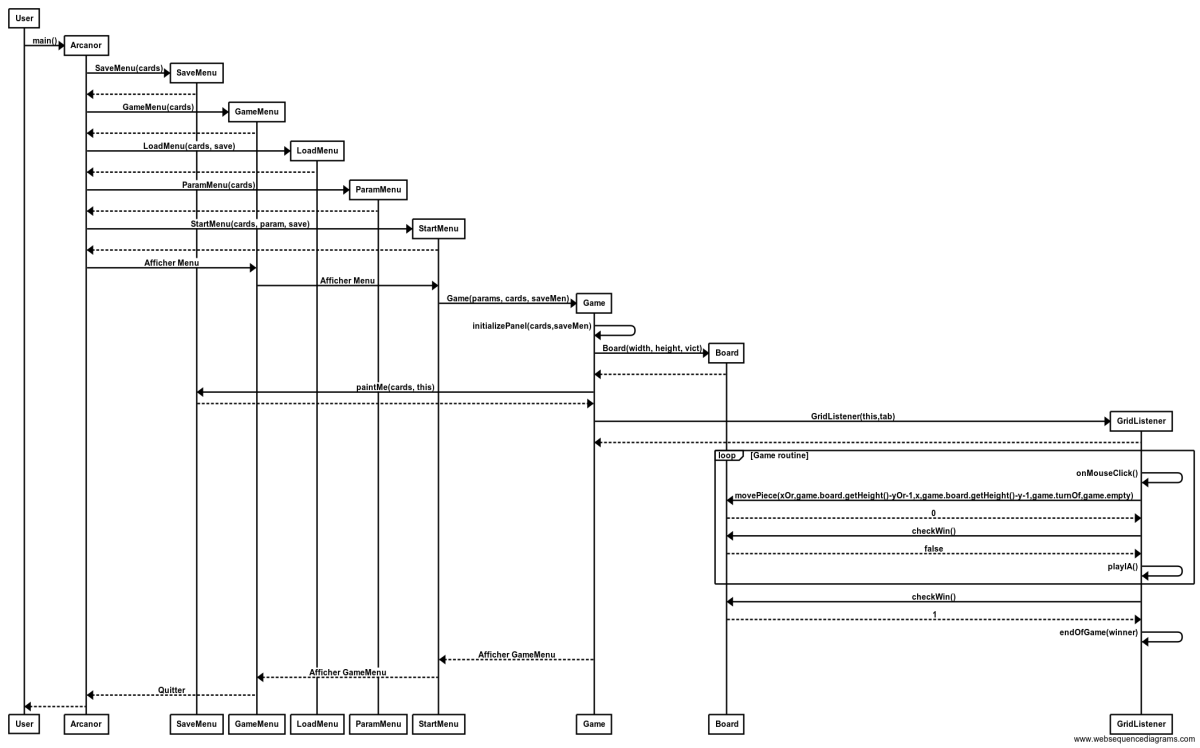


3e Version du diagramme de classes (Version définitive de l’IHM) :

1er diagramme de séquence :



2e diagramme de séquence :



1. Choix techniques et algorithmiques

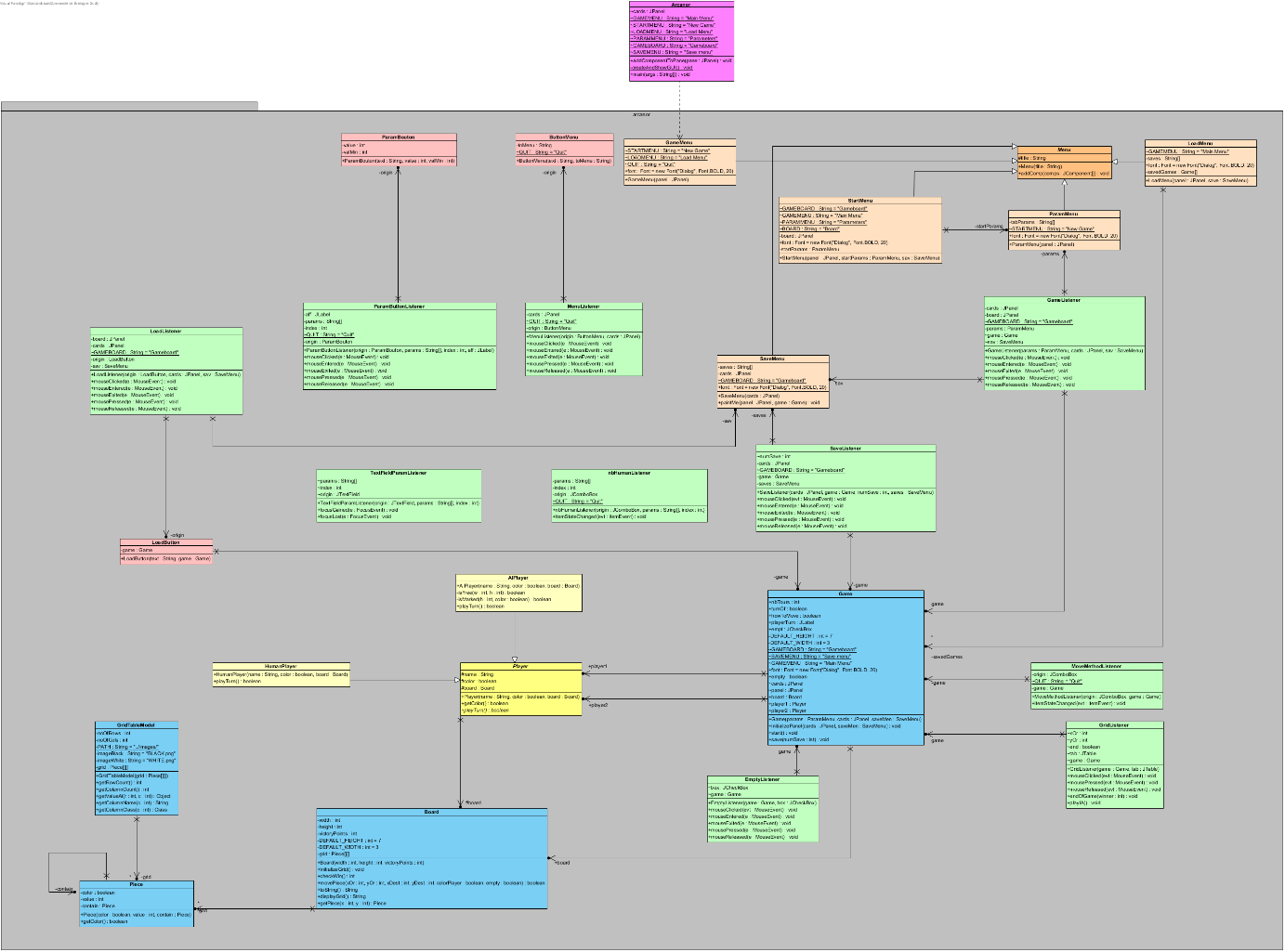
Langage de programmation : Java 8

Sauvegarde : fichier binaire (cf. PPFormatFichier.pdf)

Méthode d'affichage : un Jpanel par Menu, réuni dans un Jpanel possédant un CardLayout, permettant de passer d'un menu à l'autre sans changer de frame et simplement.

Routine de jeu : Tout se passe dans le Listener du plateau, réagissant aux actions de la souris sur celui-ci. Il n'y a pas de boucle de jeu à strictement parler, chaque clic déclanchant possiblement une action si celle-ci est valide (tour du joueur, jeu non fini,...)

1. Diagramme de classes d'implémentation



1. Description de la campagne de tests effectuée

La réalisation de la campagne de test fut longue et très difficile pour Lucas car Junit était nouveau pour nous. Lucas était assigné aux tests des classes de notre projet. Il a donc passé beaucoup de temps à essayer de faire des tests Junit de manière conventionnelle.

Ensuite Lucas a réussi via Maven à créer un environnement viable et automatisé pour faire ses tests.

1. État d'avancement réel

Version Terminal fini le Vendredi 22/06/18 comme prévu

Version IHM fini le Lundi 25/06/18 1 jour avant la date prévu

1. Synthèse des difficultés rencontrées et des solutions apportées

Problème : Réponse :

Junit Maven

Dû au caractère nouveau de Junit pour nous Lucas a eu beaucoup de mal pour compiler puis exécuter ses tests Junit. En effet le classpath fut compliqué à actualiser pour permettre la compilation. De plus lorsqu’il eut réussi à modifier le classpath comme il fallait pour la compilation ; la commande fournie pour exécuter ne fonctionnait pas avec l’organisation que Lucas avait créée. Ainsi Lucas c’est tourné vers Maven pour automatiser ses tests et avoir un environnement viable pour coder ses tests.

1. Bilan personnel du travail réalisé

Etant donné que nous étions un groupe de 3, nous avons dispatcher les tâches par affinité afin que chacun travaille le plus aisément possible avec le plus de rapidité (pour moi, un employé qui aime travailler, travailleras plus), j’ai pris le lead du groupe en ce qui concerne l’organisation et planification, mais je me suis aussi chargé de l’implémentation IA pour la Version Terminale et la Version IHM, je me suis aussi chargé de la javaDoc en générale avec l’aide de Lucas, mais aussi de la direction artistique du projet, c’est-à-dire la musique, le fond de l’application, les couleurs, les icones (qui est d’ailleurs plutôt une décision du groupe).