Py Quentin Groupe 4

Valli Dylan

Lacoste Antoine

Projet Tuteurés

Weissend Eddy

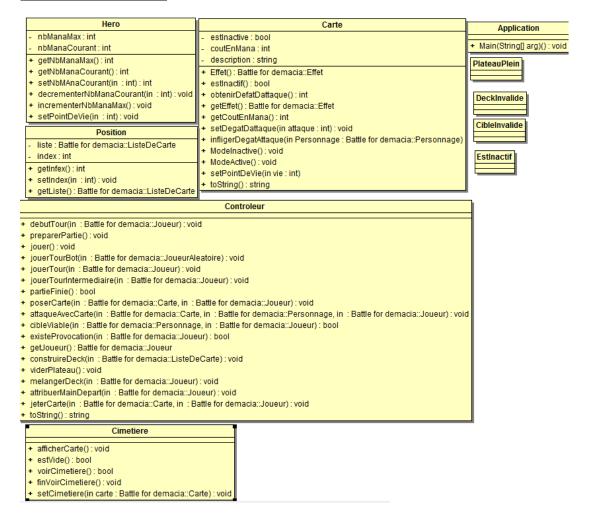
Dichtel Jonathan

Lors de la conception nous avons établi des diagrammes de classes (Voir dans le dossier D2/jalon 2) qui a amené des modifications que nous vous présentons ci-dessous :

Les 4 prochains diagrammes seront ceux avec les tous les détails des attributs et des méthodes et suivront les 2 diagrammes de classes avec les liens d'intégrités fonctionnelles sans le détail pour la partie moteur du projet mais avec en ce qui concerne l'IHM.

-les 3 Premières images sont la représentation des classes de la Partie Moteur :

<u>Première Partie :</u>

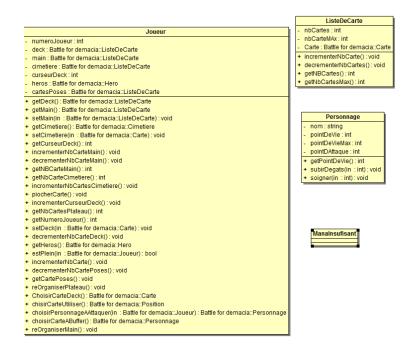


Deuxième Parti:

Jeu joueur: Battle for demacia::Joueur NB CARTES DECK: int = 60 NB MAX CARTES MAIN: int NB MANA MAX: int LISTE CARTE GENERALE: Battle for demacia::ListeDeCarte NB_CARTES_MAX_POSES: int + debutTour(in : Battle for demacia::Joueur) : void + preparationPartie():void + jouer():void + joueurTourBot(in : Battle for demacia::JoueurAleatoire) : void + jouerTour(in : Battle for demacia::Joueur) : void + jouerTourIntermediaire(in : Battle for demacia::Joueur) : void + partieFinie(): nool + finTour(in : Battle for demacia::Joueur) : void + poserCarte(in : Battle for demacia::Carte, in : Battle for demacia::Joueur) : void + attaquerAvecCarte(in : Battle for demacia::Carte, in : Battle for demacia::Personnage, in : Battle for demacia::Joueur) : void + cibleViable(in : Battle for demacia::Personnage, in : Battle for demacia::Joueur) : bool + existeProvocation(in : Battle for demacia::Joueur) : bool + getJoueur(): Battle for demacia::Joueur + contruireDeck(in : Battle for demacia::ListeDeCarte): void + creerListeDeCarteGenerale(): Battle for demacia::ListeDeCarte + viderPlateau(): void + melangerDeck(in : Battle for demacia::Joueur) : void + attribuerMainDeDepart(in : Battle for demacia::Joueur) : void + jeterCarte(in : Battle for demacia::Carte, in : Battle for demacia::Joueur) : void + toString():void + getVue() + trouverIndexCarteDansMain(in : Battle for demacia::Carte, in : Battle for demacia::Joueur) : int

Effet	JoueurAleatoire
activation:string nom:string montantdegat:int nbCible:int getActivation():string getNom():string getNontantDeDegat():int getNbCible():int appliquerEffet(in:Battle for demacia:Joueur):void	+ attribuerDeckAleatoire(): void + choisirCarteDeck(): Battle for demacia::Carte + choisirCarteAUtiliser(): Battle for demacia::Carte + choisirCarteAUtiliser(): Battle for demacia::Carte + peurEncoreJouer(): bool + peutPoserUneCarte(): bool + peutEncoreAttaquer(): bool + choisirPersonnageAAttaquer(in : Battle for demacia::Joueur): Battle for demacia::Personnage + choisirCarteAAttaqueeAleatoire(): Battle for demacia::Carte + choisirCarteABuffer(): Battle for demacia::Carte

Troisième Parti:



Pour la Partie IHM:

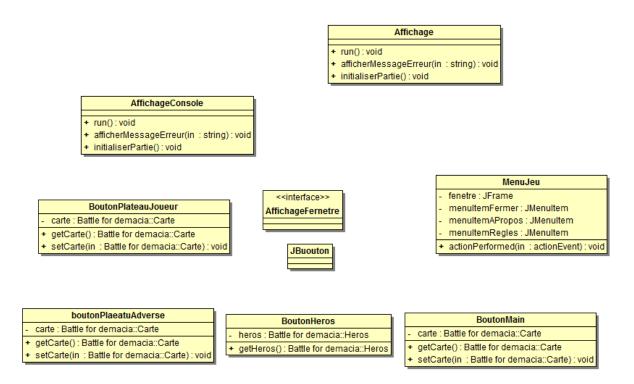
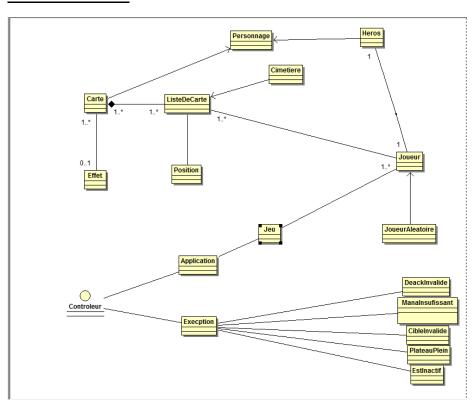
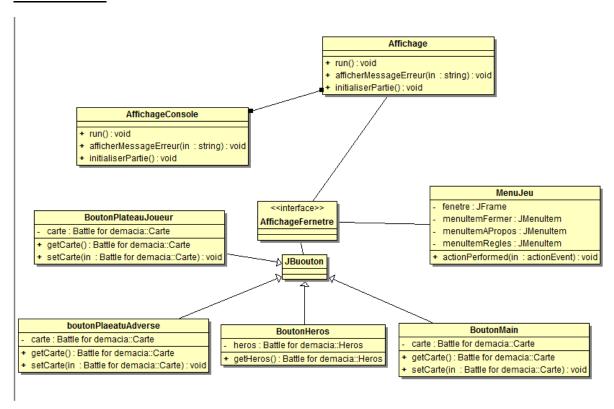


Diagramme avec les liens d'intégrité fonctionnelle :

Côté Moteur:



Côté IHM:



En conclusion nous retiendrons qu'établir un diagramme de classe correct dès le début est très difficile et nous voyons après programmation que beaucoup de chose sont manquantes sur les premiers.