

Projet C : jeu Qwinkle

Le projet sera en binôme avec la répartition du travail ...

Notation :

Validation sur machine

**Un dossier + code commenté de chaque partie par le programmeur
(individuel)**

Structure du dossier

- ☐ Une description de la structure générale du programme (8 pages maximum) contenant les principes et éléments **majeurs** de fonctionnement du programme (structures de données importantes, méthodes de résolution particulières, ...).
- ☐ Une description de chaque module du programme (1/2 page à 5 pages selon l'importance et la complexité du module)
- ☐ Une documentation utilisateur
- ☐ Un listing
- ☐ Une description du déroulement de la coopération au sein du binôme d'étudiants.
- ☐ Les jeux d'essai

Etudiant 1

L'interface (saisies, contrôle de saisie et affichage)

Le paramétrage

La Pile de coup

La machine qui joue (aléatoire en priorité)

Les tests unitaires des parties qu'il n'a pas développées

Etudiant 2

Les bateaux

La vérification des règles et le comptage des points

Les sauvegardes/ restaurations

Les tests unitaires des parties qu'il n'a pas développées

Les deux étudiants

Les tests d'intégration...

Objectif

Jouer à la bataille navale, paramétrer des options du jeu, sauver des parties, des scores de joueur, rejouer des parties.

Les règles de programmation

Respecter une programmation modulaire

Séparer saisies, traitements et affichages.

Respecter les normes (MVC Modèle Vue Contrôleur)

Proposer des fonctions de test et des jeux de tests au moins pour la partie contrôleur.

Exemple d'interface (vue)

Menu de démarrage :

Permet de créer une nouvelle partie, de charger une partie enregistrée dans un fichier, de consulter les meilleurs scores, de consulter les règles du jeu ou de quitter le programme.

Vous pouvez faire des sous menus mais à tout instant on doit pouvoir quitter une partie en l'enregistrant ou pas dans un fichier et en mettant à jour les meilleurs scores.

Créer une nouvelle partie

Permet de lancer une nouvelle partie après avoir :

- paramétré (nombre d'instances de chaque type de bateau, description des bateaux, ...) le jeu à partir de valeurs saisies ou d'un fichier de paramétrage
- Saisie du joueur (nom, prénom)
- initialisé la partie (demander au joueur de placer ses bateaux et placer ceux de la machine).

Uses cases

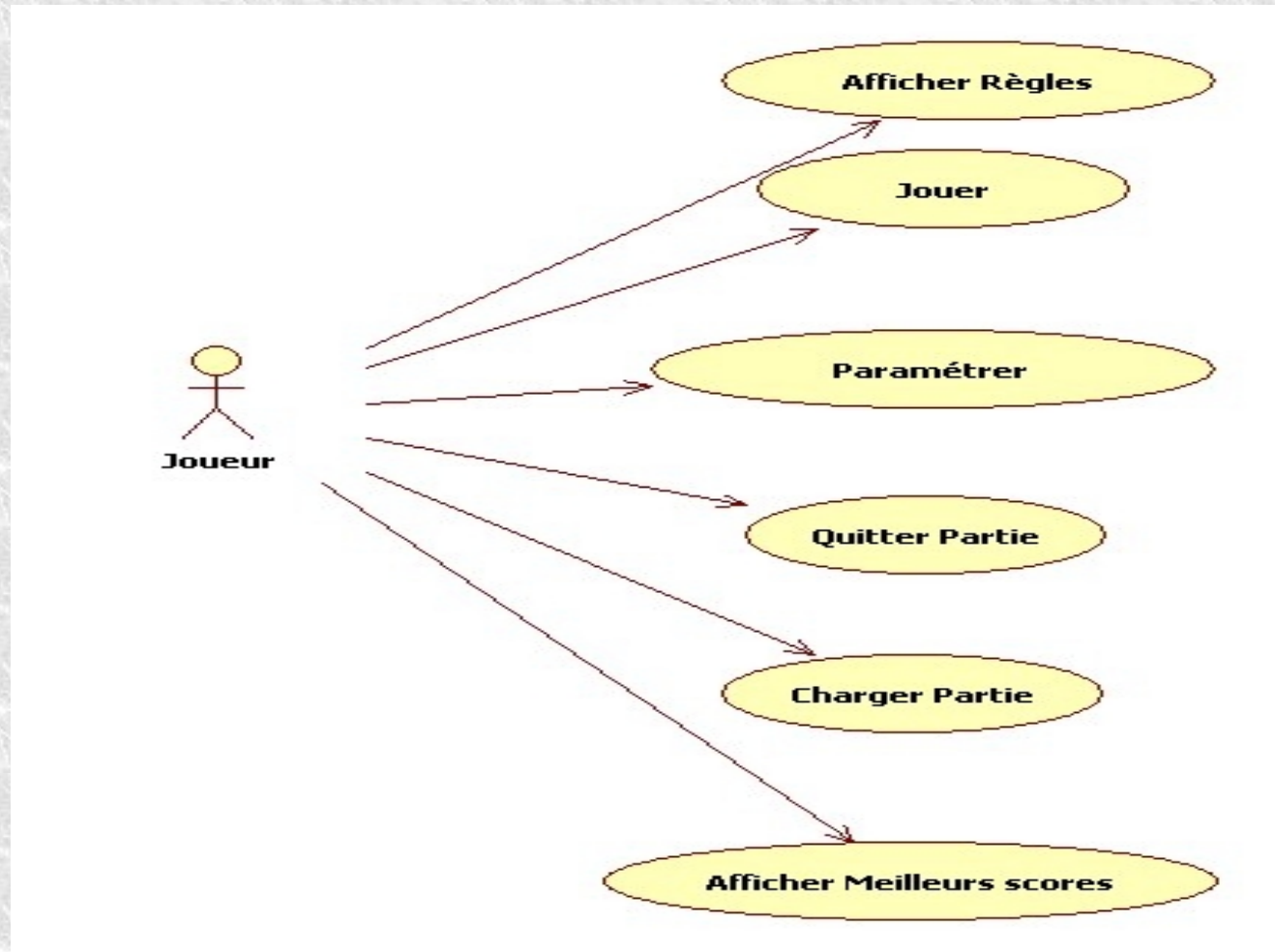


Diagramme de classe : packages

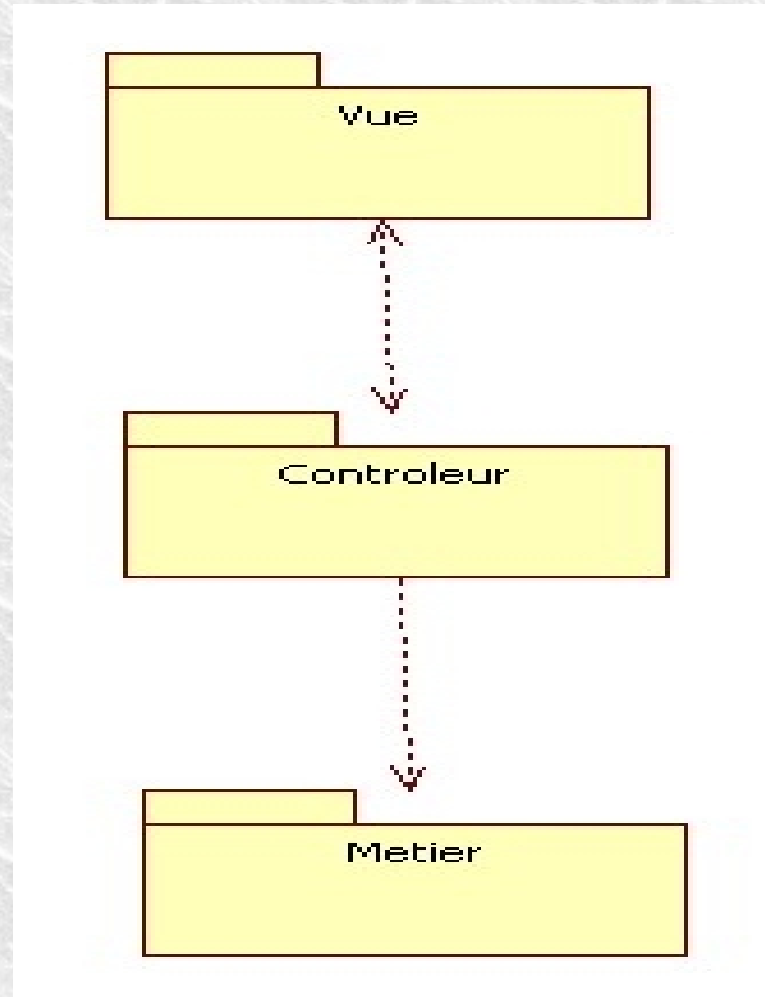
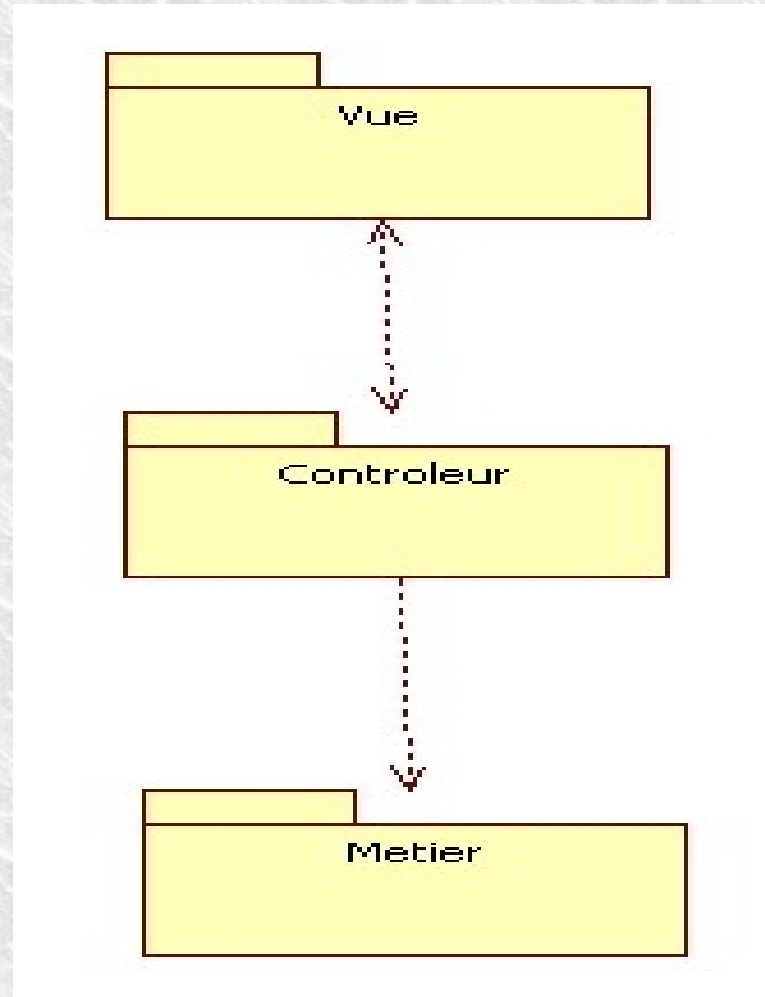
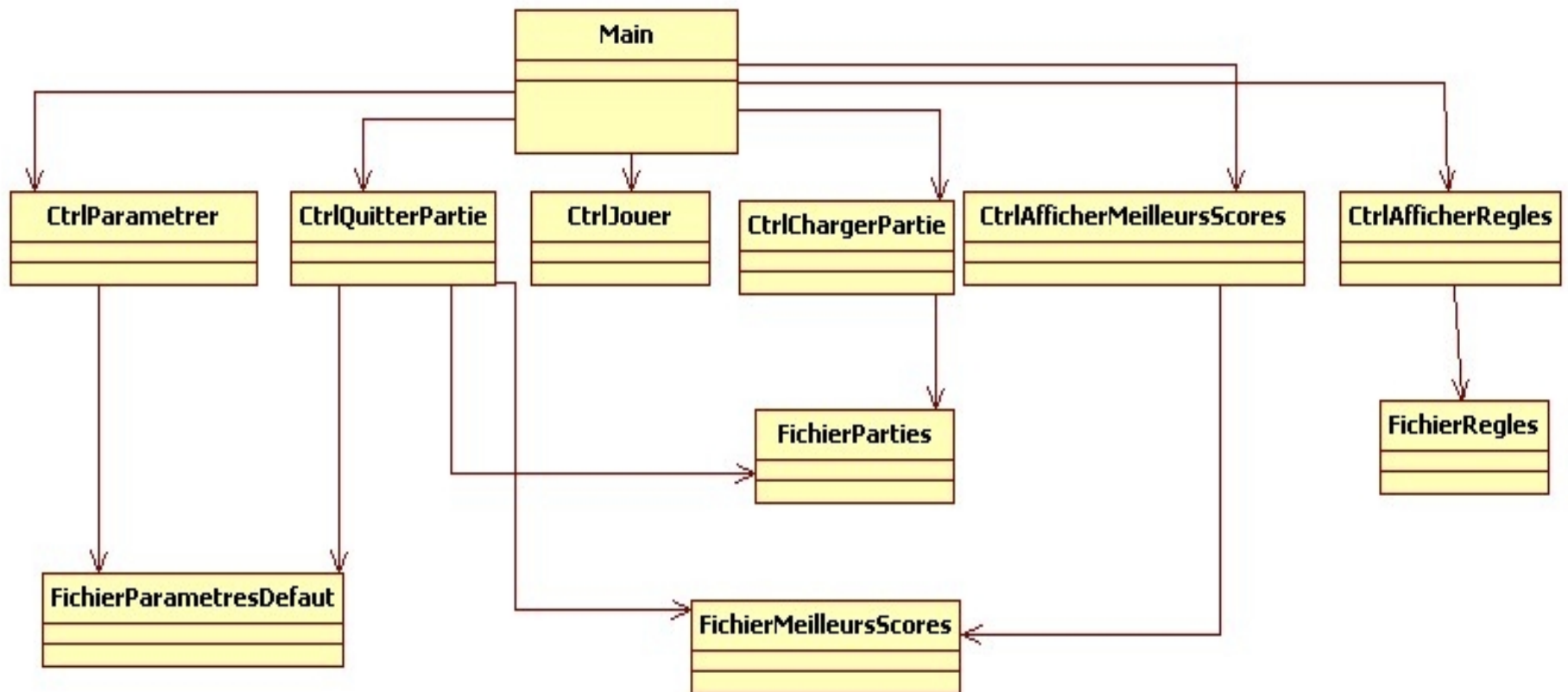


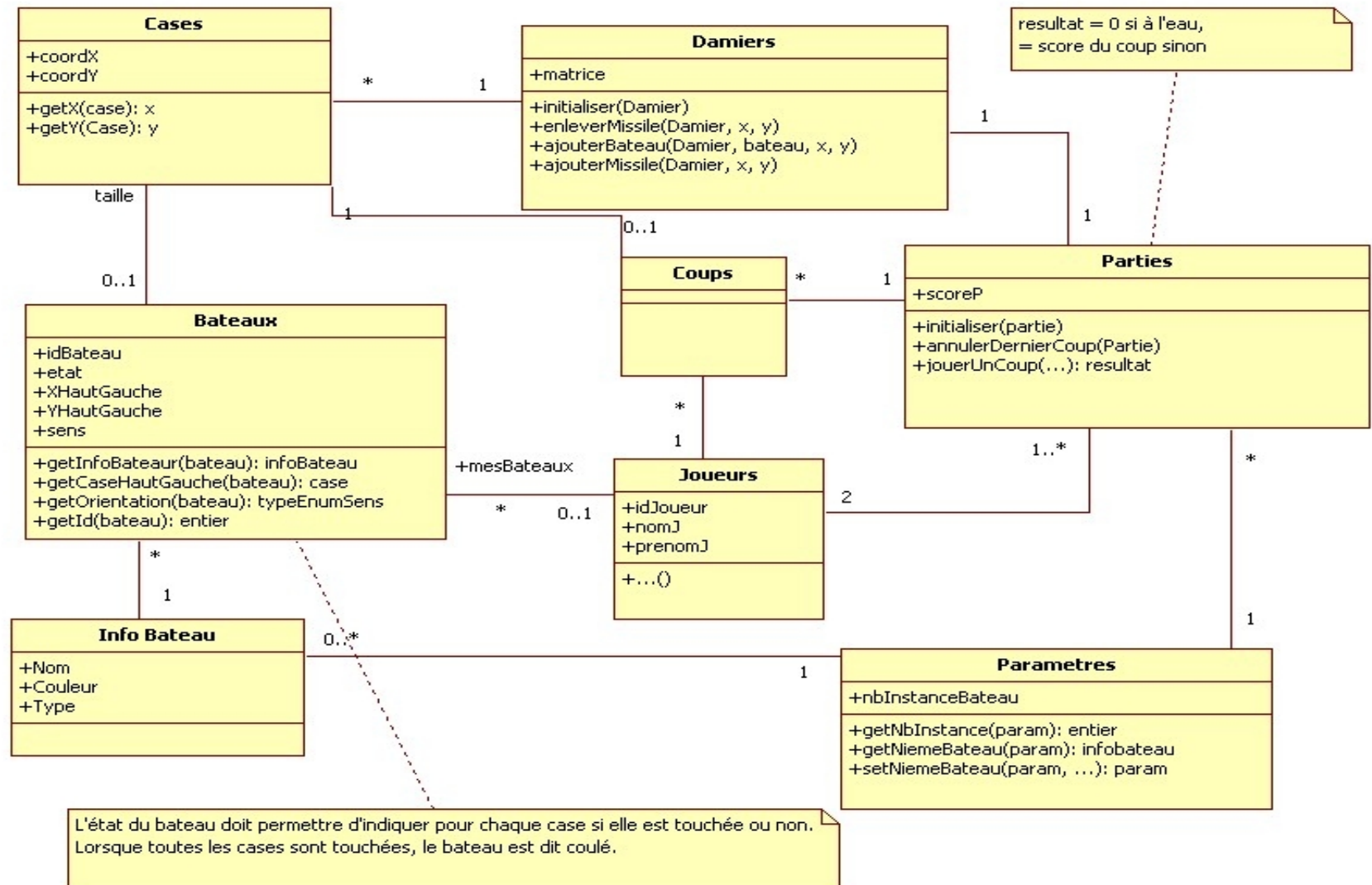
Diagramme de classe : packages



Contrôleur



Métier (incomplet)



Conception

Le travail demandé est découpé en parties.

Le début est guidé et l'objectif est de vous laisser de plus en plus d'autonomie.

L'interface peut être en mode console. Un projet mode console est disponible sous moodle pour vous permettre de démarrer.

Semaine 1-2

Objectif : Objectif définir et tester toutes les fonctions des modules parametre et vueParam.

Chaque programmeur développe un module et teste le module développé par l'autre programmeur.

Vous devez donc développer également des modules testParametre et testVue.

Faire : Récupérer le projet console sous moddle. Lancer le projet en cliquant sur le fichier .cbp (codeBlock projet). Vous avez un projet avec les fichiers .c et .h nécessaires pour débiter.

Semaine 1-2

Implantation des types de bateaux (EType)
Module parametre.h

```
typedef enum { remorqueur=2, porteAvion,  
sousMarin, cargot} EType;
```

Et après ...

- Semaine 3-4 :
 - Jouer un coup, des coups, annuler, tests et jeux d'essais
- Semaine 5 :
 - Calcul du score
 - Affichage du gagnant
 - Sauvegarde de la partie, des meilleurs scores
- Semaine 6: les derniers tests de jeux d'essais
- Semaine 7 : Le dossier
- Semaine 8 : Validation