

Ile Interdite M2104 Projet

Projet par groupe de 4 étudiants



Les règles du jeu vous sont données dans un document disponible sur Chamilo.

Chaque groupe désigne un chef de projet responsable de la planification et de la coordination des différentes tâches à réaliser, du suivi de l'avancement, du respect des délais.

Projet

L'objectif est de débiter l'informatisation du jeu « l'Île Interdite » que vous avez pratiqué en groupe, en démontrant vos capacités à concevoir et réaliser une application.

L'informatisation du **jeu complet** fera l'objet du Projet Tutoré.

Livrables

Dossier de conception

Un dossier de conception comportant au minimum :

- Un diagramme des cas d'utilisation
- Un diagramme de classes

Pensez à optimiser votre conception en appliquant le principe de polymorphisme et donc en factorisant les opérations de même sémantique.

- Les diagrammes de séquences pour les scénarios de cas d'utilisation demandés.

Réalisation

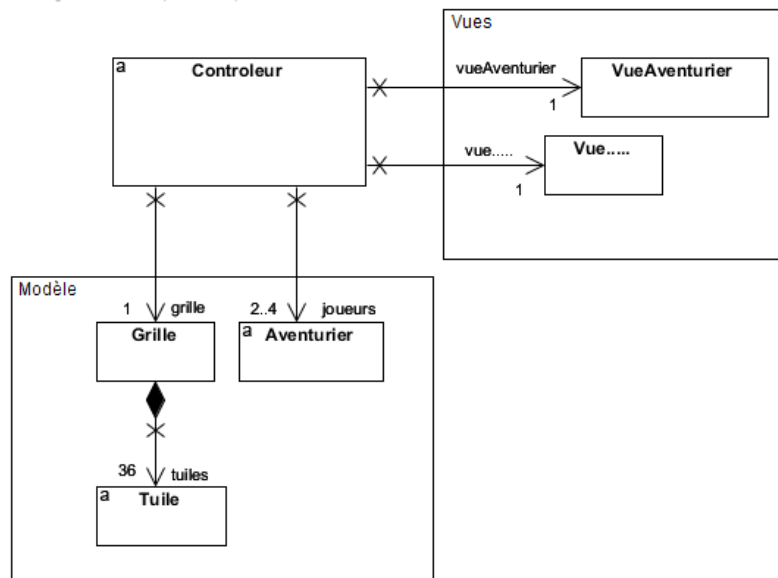
Vous devez aussi livrer sur Chamilo l'archive du code de la partie demandée du jeu en java, code évidemment conforme aux spécifications et à votre dossier de conception, basé sur une architecture MVC.

Ce dossier et le code doivent être déposés sur Chamilo au plus tard le 09/06/2017 à minuit.

Diagramme de classes proposé (à compléter)

Nous vous proposons une ébauche de diagramme de classes, que vous devrez compléter.

Visual Paradigm Standard Edition (UT2 - UPMF)



null	null			null	null
null					null
null					null
null	null			null	null

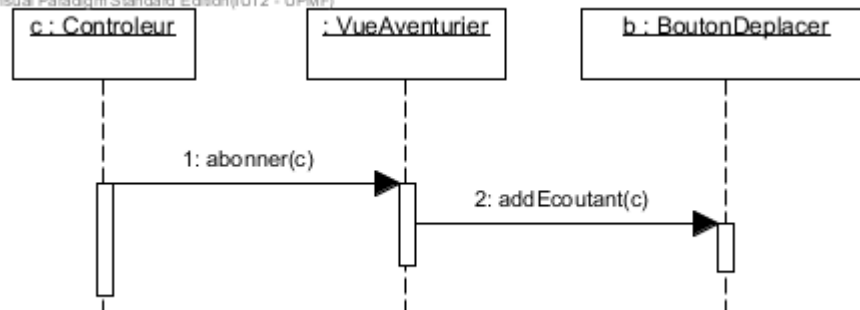
La classe **Grille** permet de représenter la grille du jeu. Elle comporte un tableau bidimensionnel (6x6) représentant les tuiles du jeu, dont les éléments en (0,0), (0,1), (0,4), (0,5) ... valent null (il n'y a pas de tuile). Ceci facilite le calcul des déplacements : un objet en position (l,c) [pour ligne, colonne !] peut aller sur les tuiles (si elles sont non null et non coulées) en position (l-1,c), (l, c-1), (l, c+1), (l+1, c) et éventuellement (pour les diagonales) en (l-1, c-1), (l-1, c+1), (l+1, c-1) et (l+1, c+1).

La classe **Tuile** représente une tuile du jeu et son état courant.

La classe **Aventurier** représente évidemment un joueur dans son rôle attribué aléatoirement et les capacités liées. La classe **VueAventurier** est la classe interactive d'un aventurier.

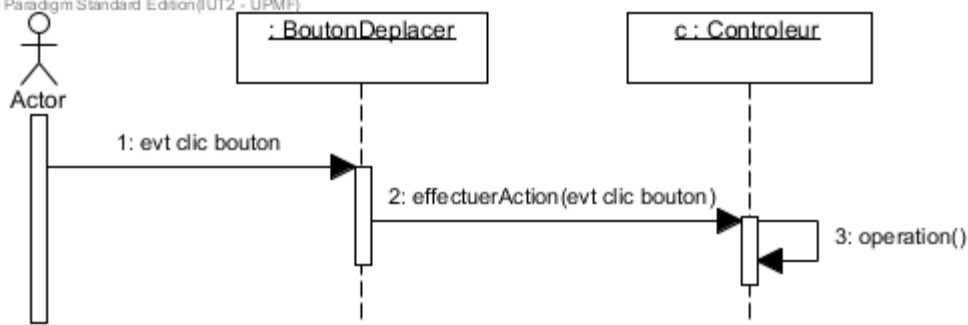
La classe **Contrôleur** gère la logique applicative. Elle est à l'écoute des événements émis par les vues.

Visual Paradigm Standard Edition (UT2 - UPMF)



Exemple

Inscription du contrôleur comme écoutant d'un clic sur un bouton d'une vue Aventurier



Exemple

Exécution par le contrôleur d'une opération suite à un clic sur le bouton Déplacer d'une vue Aventurier

Matériel fourni

Vous trouverez dans une archive zip sur Chamilo un projet NetBeans contenant quelques classes déjà codées :

- util.Utills, qui définit toutes les énumérations ainsi que quelques méthodes bien pratiques ;
- util.Parameters, qui définit les paramètres de fonctionnement de l'application.
- une classe view.VueAventurier simpliste et le bout de code du contrôleur qui réagit à un clic de bouton.

Ces classes vous sont fournies pour vous aider à démarrer.

Semaine 1 Conception

1. Comprendre le rôle du contrôleur et l'interaction Contrôleur-Modèle-Vues.
2. Compléter la partie modèle du diagramme de classes avec les classes que vous pensez utiles pour le déroulement du jeu. Ce diagramme n'est bien sûr qu'une ébauche qui sera complétée ultérieurement et éventuellement revue.
3. Faire le diagramme de séquences pour un tour de jeu : chaque joueur effectue jusqu'à 3 actions.
4. Réaliser vos modèles sous Visual_Paradigm (utiliser le TeamWork Server)
5. Imprimer vos diagrammes.

Semaine 2 Conception

1. Faire le diagramme de séquences pour le déplacement de chacun des types d'aventurier (sauf le navigateur que nous laissons de côté).
2. En parallèle, vous pouvez également faire les diagrammes de séquences pour assécher une tuile.
3. Compléter votre modélisation avec les nouveaux diagrammes de séquences.

Semaine 2 Programmation

1. Générer le squelette des classes Java à partir du diagramme de classes.
2. Coder l'initialisation d'une partie avec ses cases, les aventuriers. Inscrire les joueurs avec leur pion.
3. Coder la boucle du jeu qui fait jouer les joueurs les uns après les autres.

Semaine 3 Fin Conception - Programmation

Semaine 4 Programmation et démonstration en séance