

Projet Île Interdite

Projet par groupe de 4 étudiant·e·s.

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, et du respect des délais.

Les règles du jeu vous sont données dans un document disponible sur Chamilo¹.

Deux présentations vidéos du jeu sont disponibles :

- <https://youtu.be/n7LF550SIuk>
- https://youtu.be/dpoVfkWq_Pc

1 Projet

Ce projet a pour objectif de fédérer les connaissances acquises dans les modules « Bases de la programmation orientée objets » (M2103), « Bases de la conception orientées objet » (M2104), et « Introduction aux IHM » (M2105).

L'objectif est de finaliser l'informatisation du jeu « l'Île Interdite » que vous avez commencé dans le module M2104, en démontrant vos capacités à concevoir et réaliser une application complète.

2 Recommandations

- Le travail à fournir est *conséquent*.
- Votre temps est précieux, ne le perdez pas.
- Pensez à demander de l'aide à vos enseignants, que ce soit par messagerie instantanée ou par échange vocal pendant les créneaux Discord, ou bien par courriel en dehors de ces créneaux.
- Soignez la conception de votre application en listant avec attention tous les cas possibles.
- Établissez un planning prévisionnel de toutes les tâches à réaliser (à l'aide d'un diagramme de Gantt par exemple) et surveillez votre avancement.

3 Livrables

3.1 Dossier de modélisation (analyse et conception)

Vous complétez le dossier de modélisation réalisé dans le cadre du module M2104. En particulier, vous ajouterez les éléments d'intérêt pour la conception à l'instar de :

- Diagrammes de séquences détaillés pour les cas d'utilisation que vous jugerez pertinent. Par exemple :
 - tirage d'une carte inondation ;
 - tirage d'une carte trésor « Montée des Eaux » ;
 - déplacement d'un pion sur une tuile accessible ;

1. https://chamilo.iut2.univ-grenoble-alpes.fr/courses/INFOM2104/document/Projet/Ile_Interdite-regles.pdf

- récupération du trésor d’une tuile ;
- assèchement d’une tuile ;
- gestion d’un tour ;
- etc.
- une maquette de l’IHM

Votre dossier contiendra une section documentant l’état d’avancement de votre implémentation : quelles fonctionnalités sont terminées, en cours d’implémentation, non implémentées. Vous mentionnerez dans cette section quelles sont les défauts (bogues) connus ainsi que les limitations de votre application.

Votre dossier contiendra aussi une section mentionnant le contribution de chacun-e.

Votre dossier de modélisation sera livré au format **.pdf**. Vous réaliserez vos modèles avec le logiciel « Visual Paradigm ». Ce dossier doit comporter un sommaire, une introduction ainsi que tous les modèles lisibles et commentés. Vous utiliserez le mode paysage pour que les diagrammes soient lisibles.

3.2 Réalisation

Vous devez livrer l’ensemble du jeu implémenté en Java. Le code que vous fournirez sera évidemment conforme aux spécifications, et cohérent avec votre dossier de modélisation. L’architecture sera basé sur le patron de conception MVC (Modèle, Vue, Contrôleur). Votre application sera dotée d’une interface graphique respectant les préconisation ergonomiques vues dans le module M2105.

3.3 Démonstration

Vous réaliserez une courte vidéo (~2 min) qui présentera les fonctionnalités implémentées. Pour rendre votre présentation attractive, vous utiliserez un scénario qui permet de dérouler une partie particulière. À cet effet, vous pouvez éliminer tous les aspects aléatoires du jeu : mélange des défausses, attribution des explorateurs aux joueurs, etc.

Votre démonstration se fera sur le code que vous aurez livré.

3.4 Rendus

Votre dossier de modélisation (fichier **.vpp** réalisé avec Visual Paradigm, et dossier **.pdf**) et votre code seront enregistrés dans votre dépôt **git**. La version évaluée sera la dernière version enregistrée dans votre dépôt **git** avant la date limite.

Date de rendu La date limite de rendu est fixée au 22 juin 2020 à minuit.

4 Simplification des règles

Dans ce projet, le travail du Contrôleur est complexe. Il doit gérer le joueur courant qui mène ses actions. Le tour du joueur courant peut être « interrompu » par d’autres actions : utilisation d’une carte action spéciale, évacuation d’un joueur d’une tuile coulée, défausse d’une carte de la main d’un joueur suite à un don qui l’amène à avoir plus de cinq cartes, etc.

Pour diminuer la complexité de ce travail, vous pouvez proposer une ou plusieurs simplifications des règles du jeu. Ces simplification devront être documentées dans les limitations de votre implémentation.

Voici quelques exemples de simplification :

1. le pouvoir du Navigateur (déplacer d'autres joueurs) est remplacé par la possibilité de jouer quatre actions pendant son tour ;
2. un joueur ne peut utiliser une carte action spéciale pendant son tout uniquement ;
3. un joueur ne se défausse des cartes surnuméraires de sa main qu'au début de son tour (le système vérifie au début de chaque tour si le joueur courant a plus de cinq cartes dans sa main, et lui demande de se défausser du bon nombre de cartes si nécessaire) ;
4. etc.

5 Matériel fourni

Vous continuerez à utiliser l'espace de travail sur la plateforme GitLab de GRICAD² qui vous a été attribué pour le module M2104.

Vous trouverez dans les documents Chamilo du module M2107 une archive³ au format **.zip** contenant les éléments graphiques (*sprites*) du jeu. Ces éléments graphiques vont servir si vous souhaitez vous lancer dans un IHM avec les images.

Nous vous recommandons néanmoins vivement de commencer par une interaction sans image !

Le squelette de code qui vous a été fourni contient plusieurs classes utiles. En particulier :

util.Parameters définit les paramètres de fonctionnement de l'application.

util.Utils définit quelques méthodes pratiques : *toRGB* pour convertir une Color en valeur RGB, *afficherInformation* pour afficher une information, *poserQuestion* pour afficher une fenêtre modale avec une question totale (oui/non) ; ainsi que toutes les énumérations.

Les énumérations définies dans util.Utils sont :

Commandes utile dans les messages interprétables par le contrôleur (à compléter).

EtatTuile asséchée, inondée ou coulée.

Tresor la collection des quatre trésors. Chaque trésor est associé à un libellé, et à deux couleurs permettant de le représenter avec une couleur de texte et une couleur de fond.

Pion la collection des six pions. Chaque pion est associé à son nom et à quatre couleurs. Ces couleurs permettent de colorer la vue VueAventurier pour représenter les états actifs et inactifs, de colorer une tuile cliquable asséchée (pour y aller) ou inondée (pour l'assécher ou y aller).

2. <https://gricad-gitlab.univ-grenoble-alpes.fr>

3. <https://chamilo.iut2.univ-grenoble-alpes.fr/courses/INFOM2107/document/images.zip>