

Formateurs : Evelynne VITTORI & Marie-Laure Nivet

Modélisation/Programmation

"Mon Jeu vidéo préféré"

Dans le cadre de ce travail qui se déroule tout au long de ce semestre à partir du 18 octobre, il vous est demandé de proposer une modélisation UML et programmation du jeu de votre choix dans le langage Java en respectant les principes de la conception et programmation orientée objets étudiés dans le cadre des cours de COO (UML) et POO (Java).

Le travail est à réaliser en binôme selon votre choix.

Un canal *ProjetJeu* a été créé dans TEAMS pour partager les documents relatifs au projet. Dans l'onglet *Fichiers*, nous déposerons les fichiers vous donnant des indications générales sur le déroulement du projet, le planning et les attendus.

Nous créerons également un dossier pour chaque équipe (individuelle ou binôme) dans lequel vous devrez déposer les rendus demandés.

Les dates données dans ce document sont indicatives et sont susceptibles d'évoluer au cours du déroulement du projet. Vous serez naturellement avertis des changements sur le canal *ProjetJeu*. Il est donc de votre responsabilité de le consulter régulièrement.

1. ECHEANCIER PREVISIONNEL

- mardi 26 octobre : constitution des éventuels binômes – Vous devrez avoir complété le fichier Excel *BinomesProjetJeu* présent sur Teams dans le canal *projetJeu*, dans l'onglet *Fichiers* en spécifiant les membres éventuels du binôme et le titre prévisionnel de votre jeu (vous pourrez encore changer par la suite).
- lundi 8 novembre : courte présentation synthétique du Jeu choisi (nom, éditeur, lien vers une description de la logique de jeu), du principe de jeu, des interactions et classes envisagées : 1 à 2 transparents
- mardi 16 novembre : rendu fichier de présentation initial, ébauche diagramme de classe, description textuelle du déroulement du jeu
- lundi 22 novembre : rendu ébauche diagramme de cas d'utilisation
- lundi 6 décembre : rendu ébauche documentation cas d'utilisation
- Point hebdomadaire sur l'avancée du projet dans les cours UML et Java sur les diagrammes et sur le code
- Soutenance intermédiaire : lundi 13 décembre après-midi, présentation de l'avancée du travail réalisé à l'oral
- Rendu final (début janvier)

2. EVALUATION

L'évaluation du projet comportera trois notes:

- Note Suivi Projet (/10)
 - Sérieux et implication dans le projet pendant tout son déroulé

- Régularité du travail au cours du semestre
- Ponctualité rendus
- Qualité et soin des présentations
- Soutenance intermédiaire
- Note partie Java
- Note Partie UML

Le détail des critères d'évaluation des parties java et uml sont décrites dans la section 4 de ce document.

Le projet servira de base pour vos notes de TP en java (UE POO) et en UML (UE COO).

Les notes de TP auront en commun la note de suivi de projet.

Elles seront calculées selon les formules suivantes:

Note TP java= Note suivi projet + note partie java

Note TP UML= Note suivi projet + note partie UML

3. MODALITES DES RENDUS

Présentations du jeu

Les présentations (diaporamas) seront à déposer dans la partie fichier de votre équipe sous Teams.

Fichiers	Format de fichier	Date de dépôt
Diaporama de présentation du projet	pdf	lundi 8 novembre
Description textuelle du déroulement du jeu	pdf	mardi 16 novembre
Diaporama de la soutenance	pdf	lundi 13 décembre

Code Java

Vous devrez créer un dépôt Git et nous communiquer le lien associé (**lien à compléter dans le fichier Excel partagé "Binomes Projet Jeu"**). Vous veillerez :

- si vous travaillez en binôme à ce que les deux membres du binômes soient contributeurs sur le dépôt ;
- à associer à votre dépôt un fichier README.md décrivant votre projet (Nous vous invitons pour cela à aller consulter les “bonnes pratiques” de la communauté GitHub ou GitLab sur le sujet) ;
- à pousser **régulièrement** votre code sur le dépôt afin de nous permettre de juger de son avancée au fil du temps ;
- à documenter, et en particulier à positionner les commentaires JavaDoc permettant de spécifier les auteurs du code ainsi que les « modes d’emploi » des classes et méthodes.

Partie UML

- A la fin du projet (**début janvier**), vous devrez déposer dans la partie fichier de votre équipe sous Teams, le **dossier UML final** intégrant l'ensemble des documents demandés avec notamment les diagrammes sous forme d’images.
- Tout au long du semestre, vous devrez également déposer dans la partie fichier de votre équipe sous Teams **plusieurs fichiers intermédiaires de travail**. Ces fichiers seront des ébauches lors du premier dépôt et ils devront évoluer au fur et à mesure de l'avancée du projet.

Voici la liste des fichiers de travail à déposer dans votre dossier avec les dates des premiers dépôts. Chaque fichier déposé devra intégrer les noms et prénoms du ou des concepteurs du jeu.

Fichiers	Format de fichier	Date de premier dépôt (ébauche)
Diagramme de classe	pdf (ou image) mais pas de .mdj	mardi 16 novembre
Diagramme de cas d'utilisation	pdf (ou image) mais pas de .mdj	lundi 22 novembre
Documentation d'un cas d'utilisation	pdf	lundi 29 novembre
Modèle d'analyse	pdf	lundi 13 décembre
Dossier UML final	pdf	début janvier

4. DESCRIPTION DES ATTENDUS

4.1. Partie UML

Au niveau de la conception UML, vous devez concevoir votre jeu indépendamment des contraintes de codage. Vous devez imaginer le jeu tel que vous souhaiteriez qu'il soit dans sa version finale sans vous limiter à ce qui est réalisable dans les temps imposés en termes d'implémentation.

Le travail demandé comporte 3 parties qui correspondent à des étapes de la conception. Pour chacune des parties, des diagrammes et des documents vous sont demandés. L'ensemble de ces documents devront être regroupés dans **un seul fichier pdf (intégrant les diagrammes sous forme d'images)**.

• Modèle du domaine

Dessinez le diagramme de classe du domaine permettant de représenter toutes les informations nécessaires au fonctionnement de votre logiciel de jeu dans sa version finale.

Document à rendre : Diagramme de classe du domaine

• Modèle des cas d'utilisation

Définissez le modèle des cas d'utilisation de votre jeu. Ce modèle doit être complet au niveau du diagramme des cas d'utilisation et de leurs descriptions succinctes mais partiel au niveau de la documentation (Un seul cas d'utilisation à documenter).

Documents à rendre :

- les descriptions succinctes de chaque acteur et de chaque cas d'utilisation (1 ou 2 phrases)
- un ou plusieurs **Diagramme(s) de cas d'utilisation** (plusieurs dans le cas de plusieurs packages)
- Documentation complète d'un cas d'utilisation que vous choisirez parmi ceux que vous allez programmer en Java :
 - o documentation textuelle
 - o DSS du **scénario nominal**
 - o maquette fonctionnelle simplifiée de l'interface utilisateur faisant apparaître les champs de saisie : il s'agit juste d'une ébauche minimaliste de l'interface, vous pouvez très bien la réaliser sur word ou powerPoint par exemple.

- **Modèle d'analyse**

Définissez le modèle d'analyse partiel de votre jeu en vous limitant à la réalisation du cas d'utilisation que vous avez choisi de documenter dans le modèle des cas d'utilisation.

Documents à rendre :

- Le diagramme des classes participantes de votre cas d'utilisation (diagramme de classe d'analyse restreint aux classes impliquées dans la réalisation de votre cas d'utilisation)
- Le diagramme de séquence d'analyse du **scénario nominal** du cas d'utilisation.

Les points soumis à évaluation coté UML

Ce travail sera évalué tout au long du semestre en fonction des avancées du cours. Les points suivants seront considérés :

- Qualité et cohérence des diagrammes
- Maîtrise des concepts manipulés
- Qualité de la documentation des cas d'utilisation
- Soins des rendus

4.2. Partie JAVA

Les classes programmées doivent permettre d'illustrer et de mettre en œuvre la logique du jeu décrite dans un ou plusieurs cas d'utilisation identifié(s) dans la partie UML. Le choix des cas d'utilisation effectivement développés devra avoir été validé par les enseignants.

- Entre 3 et 5 classes liées les unes aux autres par des relations d'association, d'aggrégation et/ou de composition sont attendues.
- Elles doivent permettre de montrer votre compréhension et votre maîtrise des concepts de la POO niveau programmation en Java.
- Les classes à terme¹ doivent comporter :
 - des attributs de classe et d'instance ayant différents niveaux d'accessibilité,
 - des constantes de classe et/ou d'instance,
 - des méthodes de classe et d'instance permettant d'interagir avec les classes et les objets et de développer une forme d'interaction basique entre vos classes,
 - au moins un exemple de méthode surchargée,
 - des constructeurs au nombre d'au moins deux par classe,
 - les méthodes héritées de la classe `Objet` vous paraissant nécessaires,
 - au moins une relation d'héritage dans laquelle au moins une méthode sera redéfinie,
 - au moins une classe abstraite,
 - une mise en œuvre d'un clonage,
 - au moins une interface,
 - au moins une classe `Enum`,
 - de la gestion d'exception,
 - l'utilisation de diverses collections d'objets,
 - une organisation modulaire/package du code,
 - un main illustrant de façon basique et textuel (aucune interface graphique n'est demandée) une partie de la logique de jeu (simplifiée) mise en œuvre.

¹ C'est-à-dire à la fin du semestre, une fois tous les concepts vus en cours

Les points soumis à évaluation coté Java

Ce travail sera évalué tout au long du semestre en fonction des avancées du cours. Les points suivants seront considérés :

- originalité du code et de la programmation, tout "sourçage" (en ligne ou sur un camarade) non assumé et dument signalé sera porté à votre discrédit ;
- respect des conventions de nommages ;
- respect de design guidelines java (vous trouverez sur le site du cours des références en ce sens) ;
- respect des principes d'encapsulation ;
- lisibilité des codes (présentation, commentaires) ;
- logique de jeu développée : niveau d'interaction avec les objets et des objets entre eux ;
- traces d'un travail au long court, échelonné tout au long du semestre (push réguliers sur le dépôt Git).

Documents à rendre :

- Codes source des classes/interfaces Java, mis à disposition sur un dépôt Git dont vous nous donnerez l'adresse.