

---

Le projet **AP4B** a pour objectif d’initier les étudiants à la conception orientée objet et au développement d’applications ludiques à l’aide du langage **Java**. Dans ce cadre, il est proposé de réadapter des jeux de société existants sous forme d’applications numériques. Le but n’est pas de réaliser un jeu commercial ou graphiquement abouti, mais de concevoir un **cœur fonctionnel solide**, bien structuré et conforme aux principes de la programmation objet. L’interface graphique servira à illustrer le fonctionnement du jeu sans en être le point central. L’accent est mis sur la **conception UML**, la **structure du code**, et la capacité à modéliser les entités essentielles du système de jeu. Chaque groupe, composé de (2 à 4), choisira un jeu parmi les deux proposés et en proposera une adaptation thématique autour de l’univers de l’UTBM (vie étudiante, campus, cours, examens, etc.). L’objectif est de concevoir un jeu qui garde l’esprit des règles originales tout en y intégrant une touche de créativité liée à l’environnement universitaire.

Les groupes doivent être constitués et communiqués avec le sujet choisi pour le 26 novembre (avant minuit). Une fois un groupe mis en place pour un sujet, il ne peut plus y avoir de modification d’ici la fin du semestre. Les changements/séparation de groupe et les changement de sujet sont interdits.

Avant les vacances de Noël, le rapport sera à rendre et portera principalement sur la partie conception UML. A la rentrée de Janvier, vous devrez faire une présentation de votre projet avec démonstration sous forme de vidéo et nous fournir (1) votre projet sous la forme d’un fichier exécutable (2) vos codes sources (3) un rapport modifié si jamais la partie conception a été changée lors de la phase de programmation. Voici donc le planning prévu:

Avant la 2 <sup>ème</sup> séance	Constitution des groupes et choix du sujet. (Liste des équipes de travail)
Du 12/11 au 24/11	Travail sur les cas d'utilisation. Proposition du premier diagramme de classe et de cas d'utilisation.
Du 28/11 au 10/12	Travail sur la conception Diagrammes UML nécessaires du fonctionnement de l'application.
À partir du 12/12	Finalisation de la conception, rédaction du rapport de conception Début de la programmation.
19/12 (avant minuit)	Rendu du rapport de conception.
09/01 (avant minuit)	Rendu du projet (vidéo, sources, exécutable, rapport modifié si nécessaire)

En rouge les dates limites.

## Jeu 1 – *Love Letter*

Le jeu *Love Letter* est un petit jeu de cartes à la fois stratégique et rapide, qui repose sur la déduction, le bluff et la gestion du risque. Les joueurs incarnent des courtisans cherchant à transmettre une lettre d’amour à la princesse tout en éliminant la concurrence. Le principe du jeu est simple : à chaque tour, un joueur pioche une carte, en choisit une à jouer et applique son effet. Ces effets peuvent révéler des informations sur les autres joueurs, modifier les cartes en jeu ou protéger son propre personnage. L’objectif est de terminer la manche en étant le seul joueur encore en jeu ou en possédant la carte de plus grande valeur lorsque le paquet est épuisé. L’adaptation demandée consiste à concevoir une version numérique du jeu dans un contexte UTBM. Par exemple, la princesse pourrait être remplacée par un responsable d’unité d’enseignement, les courtisans par des étudiants, et les cartes représenteraient différentes actions académiques (rapport, projet, note, rattrapage, etc.). L’application devra gérer l’intégralité de la logique du jeu : distribution, effets des cartes, conditions de victoire et gestion des tours. Une interface simple permettra d’afficher les cartes en main, les actions possibles et le déroulement de la partie.

## Jeu 2 – *Trio*

*Trio* est un jeu de cartes basé sur la mémoire et la déduction. Le principe repose sur la recherche de trois cartes identiques parmi un ensemble de cartes réparties entre les joueurs et le centre de la table. À chaque tour, un joueur tente de retourner deux cartes de même valeur. S’il réussit, il peut essayer d’en trouver une troisième pour former un trio. Chaque trio complété rapporte un point, et le joueur rejoue. La partie se termine lorsque tous les trios ont été trouvés. Dans le cadre du projet, les étudiants devront concevoir une adaptation numérique de ce jeu en Java. L’univers pourra être librement revisité : par exemple, le jeu pourrait devenir une chasse aux “trios de rapports”, “trios de projets” ou “trios d’étudiants d’un même groupe”. L’application devra permettre de manipuler les cartes à l’écran à l’aide d’une interface graphique, d’afficher les révélations successives et de gérer le score des joueurs.

## Adaptation UTBM et travail demandé

Les étudiants doivent adapter le jeu choisi à un contexte propre à la vie étudiante de l’UTBM. L’univers du campus, les enseignants, les cours, les laboratoires ou encore les activités étudiantes peuvent devenir la base du thème de l’application. Cette adaptation thématique doit être visible à travers les noms des entités, les cartes ou les éléments visuels du jeu.

Sur le plan technique, l’objectif est de modéliser le jeu en respectant les principes de la programmation orientée objet. Le développement doit être découpé en trois grandes étapes, correspondant aux trois séances du projet. À chaque séance, une progression logique est attendue, allant de la conception abstraite à la concrétisation du jeu fonctionnel.

# Déroulement du projet

## Séance 1 – Conception et modélisation UML

Cette première séance marque le point de départ du projet. Elle est entièrement consacrée à la création des groupes, au choix du jeu, à la compréhension du jeu choisi et à son adaptation à l'univers de l'UTBM, ainsi qu'à la mise en place d'une architecture logique claire à l'aide du langage UML. Les étudiants doivent commencer par lire attentivement les règles du jeu (Love Letter ou Trio) afin d'en dégager la structure globale, les entités principales et les interactions qui les relient.

L'objectif est de passer d'une description ludique à une représentation conceptuelle : quelle adaptation thématique du jeu à l'UTBM allez-vous proposer ? Quelles sont les classes nécessaires pour faire fonctionner le jeu ? Quelles relations existent entre elles ? Quels sont les attributs et les comportements attendus pour chaque entité ?

À partir de ces réflexions, chaque groupe devra produire :

- un **diagramme de cas d'utilisation** permettant d'identifier les acteurs et les interactions avec le système ;
- un **diagramme de classes UML préliminaire** représentant les principales entités du jeu et leurs relations (associations, héritages, agrégations, etc.) ;
- une description textuelle des principaux scénarios de jeu (début de partie, tour de joueur, fin de partie).

En fin de séance, le squelette du projet Java devra être créé, comprenant les packages et les classes principales, même si celles-ci ne contiennent encore aucune implémentation. L'objectif est que chaque groupe dispose d'une structure de code prête à accueillir la logique du jeu.

## Séance 2 – Implémentation du cœur logique en Java

La deuxième séance est consacrée à la traduction du modèle UML en une version exécutable du jeu, sans interface graphique. L'objectif est de donner vie au cœur logique du projet à travers le code Java.

Les étudiants commenceront par implémenter les classes conçues lors de la première séance : création des objets représentant les cartes, les joueurs, le plateau ou la partie, selon le jeu choisi. Chaque méthode doit être programmée en respectant la conception UML initiale, tout en permettant d'ajuster celle-ci si nécessaire au fur et à mesure de la mise en œuvre.

Le jeu devra être testable en mode **console** afin de vérifier le bon fonctionnement des règles : tirage aléatoire des cartes, gestion des tours, effets spécifiques (dans Love Letter) ou vérification des trios (dans Trio). L'affichage en console servira uniquement de support pour valider la logique du jeu.

Pendant cette séance, les étudiants apprendront également à gérer les transitions d'état : début de partie, tour actif, vérification des conditions de victoire et fin de partie. Une

fois la logique validée, ils mettront à jour leur diagramme UML pour refléter les éventuelles évolutions par rapport au modèle initial.

À la fin de la séance, chaque groupe doit disposer d'une version jouable du jeu en mode texte, permettant de dérouler une partie complète avec des entrées simples (ou même des actions automatiques pour les tests).

## Séance 3 – Interface graphique et finalisation du projet

La troisième et dernière séance vise à donner une forme visuelle et interactive au projet. Après avoir validé la logique interne, les étudiants devront intégrer une interface graphique.

L'objectif de cette phase est de représenter les entités du jeu (cartes, joueurs, scores, messages) à l'écran et de permettre à l'utilisateur d'interagir avec l'application : lancer une partie, effectuer un tour, visualiser les résultats, etc. L'interface n'a pas besoin d'être esthétiquement complexe, mais elle doit être fonctionnelle et refléter le déroulement du jeu.

## Rendu de projet

Le rapport de TP et le code sont à rendre en groupe de 2 à 4 personnes pour le **09 Janvier 2026** avant minuit. Vous déposez le rapport et le code dans la section "devoir" dédiée à ce TP. Seuls les codes en JAVA sont acceptés et les rapports en format pdf. Donc vous avez un total de trois fichiers à télécharger pour le projet :

1. **Rapport\_AP4B\_NomEquipe.pdf** : Explication détaillée du travail effectué + version final de la conception UML.
2. **Code\_AP4B\_NomEquipe.zip** : Code en JAVA (+ fichier exécutable .jar).
3. **Video\_AP4B\_NomEquipe.mp4** : Présentation vidéo du jeu d'environ 5 minutes.