



Le PERUDO est un jeu de société dont le but consiste à être le dernier joueur à posséder un ou plusieurs dés.

En début de partie, chaque joueur (au nombre maximum de 6) prend un gobelet et les 5 dés correspondant à sa couleur. Le premier joueur à jouer est défini aléatoirement et la partie se poursuit dans l'ordre des aiguilles d'une montre.

La partie se déroule de la manière suivante :

- les joueurs mélangent leurs dés, ils sont alors les seuls à connaître la valeur des dés mélangés ;
- le premier joueur doit faire une proposition : il doit annoncer un nombre de dés ainsi qu'une valeur de dés ;
- le joueur suivant dispose de trois choix :
 - Faire une **nouvelle proposition** qui va **surenchérir** la précédente (soit en augmentant le nombre de dés, soit en augmentant la valeur ou bien les deux) ;
 - Déclarer que le joueur précédent **ment**. Ici, tous les joueurs vont dévoiler leurs dés. Si le nombre annoncé par le joueur précédent est inférieur à ceux sur la table alors il perd un dé. Sinon c'est le joueur qui l'a accusé qui perd un dé ;
 - Déclarer le "tout-PILE". Ici, tous les joueurs vont dévoiler leur dés. Si le nombre de dés annoncé vaut exactement ceux sur la table alors le joueur gagne un dé, sinon il en perd un.
- Si une nouvelle proposition a été annoncée, le tour continue. Si un joueur a gagné ou perdu un dé, un nouveau tour commence avec ce dit-joueur qui va annoncer une nouvelle proposition (il n'est pas obligé de surenchérir la dernière proposition).

Les dés peuvent prendre comme valeur : PERUDO, 2, 3, 4, 5 et 6. La valeur PERUDO est une valeur de dé qui prend en valeur celle du dé annoncé (il s'agit donc d'une sorte de joker).

L'objectif de ce projet est de modéliser le jeu sous la forme d'une interface logiciel codée en Java. Le logiciel pourra *héberger* ou *rejoindre* une ou plusieurs parties (cf le détail des versions).

Exemple



La partie débute avec 3 joueurs : Ana, Bernard et Jo.
Chacun d'entre eux dispose de 5 dés.

- Jo débute la partie, il annonce **cinq 3** (**cinq** dés ayant pour valeur **3**) ;
- C'est au tour d'Ana, elle annonce **cinq 4**.
- C'est au tour de Bernard, il annonce **menteuse**.
- Tous les joueurs dévoilent leurs dés : il y a un PERUDO, trois 2, cinq 3, trois 4, un 5 et deux 6.
On compte : un PERUDO + trois 4 = quatre 4. Ana a menti, elle perd un dé.
- Un nouveau tour débute, les dés sont mélangés. C'est à Ana de commencer puisqu'elle a perdu.
Ana annonce **six 5**.
- C'est au tour de Jo, il annonce **tout-pile**.
- Tous les joueurs dévoilent leurs dés : il y a quatre PERUDO, un 2, cinq 3, un 4, deux 5 et un 6.
On compte : quatre PERUDO + deux 5 = six 5. Le tout-pile est réussi, Jo récupère un dé. Bravo Jo.

Travail demandé

V1 : 1 Partie et on y va

Dans cette première version, une seule partie est organisée à la fois. 1 serveur central est chargé d'héberger et de synchroniser le jeu des clients.

Les clients disposant de leur côté d'une application permettant d'y jouer.

V2 : X Parties et ça y va

Maintenant, le serveur doit être capable d'héberger plusieurs parties

Les applications des clients doivent permettre de sélectionner une partie et d'y jouer.

V3 : Peer Parties et c'est bon

Chaque client peut dorénavant héberger 1 partie. Le serveur central dispose de la connaissance des parties (en cours ou à venir).

V4 : Messagerie inline

Dans cette étape, un utilisateur doit pouvoir dialoguer en privé avec un autre utilisateur du système. Un utilisateur peut ouvrir une « fenêtre de dialogue » pour discuter avec un autre contact. Ces conversations privées ne devront pas transiter par le serveur.

Organisation du projet et travail attendu

Ce projet est à rendre de manière incrémentale. Chaque livrable sera évalué et doit comporter deux parties :

Partie I : Conception de l'application

Concevez et rédigez les interfaces de l'application permettant de répondre aux spécifications précédemment énoncées. Constituez un dossier qui comportera :

- Les diagrammes UML adéquats exprimant le résultat de l'analyse de cette application.
- Les différentes catégories d'entités logicielles en identifiant leur rôle et leur positionnement envisagé sur l'environnement technologique cible.
- Les interactions pouvant survenir entre ces types d'entités.

Partie II : Développement d'une maquette de l'application

Une maquette de la solution envisagée doit être développée. Celle-ci devra donner une image fidèle de la solution réelle ensuite déployée. On s'attachera en priorité (au-delà des aspects IHM, traitement, persistance...) à rendre opérationnelles les interactions entre entités logicielles.

Évaluation

A chaque échéance, chaque trinôme devra présenter son projet et faire une démonstration. Ce même jour, vous devez rendre le rapport lié à la version livrée. Votre projet sera évalué en fonction des choix techniques que vous effectuerez, de la qualité technique du code produit, de la qualité de la documentation, de son respect du sujet, du respect des échéances, de l'ergonomie de votre application et de son allure générale.

Tout retard dans la fourniture des livrables sera sanctionné dans l'évaluation.

Vos livrables pourront être soumis à évaluation d'outils détectant différentes formes de plagiat.

Rappel sur le plagiat :

Il existe de nombreuses bibliothèques et applications qui réalisent tout ou partie du travail demandé. Vous pouvez les regarder et vous en inspirer à la condition expresse de citer vos sources. Mais, pour la réalisation de ce projet, il vous est interdit :

- de reprendre du code de ces bibliothèques ;
- de s'inspirer, de plagier le code ou la conception réalisée par les autres étudiants.

Consignes diverses :

- Les fonctionnalités supplémentaires que vous mettrez en œuvre seront aussi comptabilisées à la condition que tous les éléments du projet aient été correctement traités au préalable ET sont soumises à l'accord préalable de l'équipe enseignante.

ÉCHEANCES :

- Date de remise de la version 1 : 01/02/2017
- Date de remise de la version 2 : 15/02/2017
- Date de remise de la version 3 : 8/03/2016