

	<p align="center">Concepts de programmation avancés</p> <p align="center">Projet de NFP 121</p> <p align="center">2021-2022</p>	<p align="right">21/01/2022</p> <p align="right">Par J. LAFORGUE</p>
--	--	--



NFP 121 – Travaux pratiques

*IPST-
CNAM
Laforgue*

PROJET

Années 2021-2022
TOULOUSE
J. LAFORGUE

1. LE BESOIN

Nous nous proposons de réaliser un programme interactif permettant à plusieurs joueurs de jouer au **Poker Texas Hold'em no-limit** (https://fr.wikipedia.org/wiki/Texas_hold%27em)

Afin de simplifier l'affichage de la table de Poker, nous limitons le nombre maximal de joueur à 4 (nord, sud, ouest, Est). Mais on peut jouer à 2 ou 3 joueurs uniquement.

Le programme pourra être réalisé en 2 grandes étapes :

- une première version, appelée « version locale », dans laquelle, le programme est une seule JVM dans laquelle les IHM de chacun des joueurs, et l'Applicatif de jeu sont des classes liées entre elles.
- une deuxième version, appelée « version distante », dans laquelle le programme est constitué de :
 - Une JVM (appelé « Serveur ») contenant l'Applicatif de Jeu. Cette JVM est lancée en premier.
 - Autant de JVM qu'il existe de joueur. Cette JVM se connecte au serveur afin que le joueur puisse jouer.

La version cible du projet est la version Distante. Vous pouvez décider de faire directement la version Distante.

L'avantage de faire en deux versions est que si vous n'arrivez pas à terminer la version finale vous aurez un projet complet qui fonctionne en local.

Le désavantage est de « perdre du temps » car le passage de la version Locale à la version Distante nécessite de modifier légèrement certains aspects de programmation (ce sera léger si vous avez suivi une conception à base de Designs Patterns).

Dans la version Locale, vous précisez en paramètre de lancement du programme le nombre de joueur autour de la table. Chacun des joueurs s'appellent « JOUEUR 1 », « JOUEUR 2 », ...

Dans la version Distante, le Serveur étant lancé, c'est en lançant plusieurs Client IHM de joueur qui se connecte au Serveur que le Serveur saura le nombre de joueur autour de la table. Le programme Client IHM a en paramètre d'exécution le nom du Joueur.

Chaque joueur distribue à tour de rôle les cartes pour commencer une nouvelle manche.

Tous les joueurs ont le même nombre de jetons. Ce nombre est une constante de l'Applicatif de jeu. Pour simplifier, tous les jetons ont la même valeur.

	Concepts de programmation avancés Projet de NFP 121 2021-2022	21/01/2022 Par J. LAFORGUE
--	--	--------------------------------------

Chaque IHM d'un joueur affiche :

- le nom de chacun des joueurs avec un indicateur qui permet de savoir si c'est à son tour de faire une action.
- les cartes de chacun des joueurs. Seuls les cartes du joueur dans sa propre IHM sont visibles.
- le nombre de jetons que possèdent chacun des joueurs
- le total du nombre de jetons misés par tous les joueurs pendant une manche, appelé le « pot ».
- les cartes du « tableau » (flop, tournant, rivière) visibles au centre de la table
- une zone de saisie permettant de changer la valeur de la blind courante (par défaut la valeur de la blind est de 2 jetons)

Les joueurs sont sensés connaître les règles du jeu. Ainsi, on ne demande pas que le programme détermine quels actions un joueur a le droit de faire à son tour. Cela vous permet de réaliser ce programme sans connaître précisément les règles de ce jeu de Poker. Cependant, le programme détermine quel est le tour du joueur.

A son tour chaque joueur peut faire les actions suivantes :

- miser la simple blind
- miser la double blind
- miser des jetons
- faire « tapis » (all-in)
- passer (checker)
- abandonner
- prendre le « pot ». En cas de « tapis », le joueur prend les jetons que possèdent chacun des joueurs qui ont également fait « tapis » à la hauteur des jetons qu'il possède, en plus de pot qu'il récupère.

Seul le donneur peut faire les actions suivantes :

- distribuer le flop
- distribuer le turn
- distribuer le river
- recommencer une nouvelle partie.
 - Si c'est la première fois qu'un joueur fait cette action, le jeu de cartes est mélangé, il est le donneur, et les cartes sont distribuées.
 - Sinon cette action termine la partie courante, et le joueur à la gauche du donneur devient le nouveau donneur. Le jeu est coupé et les cartes sont distribués pour commencer une nouvelle partie.

Le gagnant est celui qui a éliminé tous ses adversaires.

Le joueur qui n'a plus de jetons quitte la table. Mais il peut continuer à regarder la partie se dérouler (l'IHM du joueur reste)

Un joueur ne pas miser plus de jetons qu'il n'a.

	Concepts de programmation avancés Projet de NFP 121 2021-2022	21/01/2022 Par J. LAFORGUE
--	--	--------------------------------------

2. L'ENONCE

2.1. *Le dossier du projet*

Faire le dossier du projet (fichier word) suivant le plan suivant :

1/ Identification

Vous précisez dans ce § :

- Le numéro de votre groupe de projet et le nom de guerre de la Team (un seul mot)
- les noms et prénoms des membres du projet
- les rôles de chacun (si nécessaire)

2/ Conception préliminaire

Une reformulation du besoin et, éventuellement, un complément du besoin dans une démarche de conception permettant de s'assurer de la bonne compréhension du besoin et de présenter une certaine conception technique de l'ensemble du projet.

Un diagramme de communication UML peut être réalisé afin d'identifier les DP et composants, et de décrire la communication entre ces composants.

1/ Identifier les cas d'utilisation

Dans un premier temps, identifiez les cas d'utilisation nominaux.

2/ Définir l'aspect de l'IHM

Un dessin sous PPT qui décrit les différents éléments de l'IHM

Précisez pour chacun des éléments quels actions sont réalisables par l'utilisateur.

3/ Faire le diagramme de classe

Faire le(s) diagramme(s) de classe de l'ensemble du projet.

Ce diagramme est une première ébauche. Il sera enrichi en fonction du développement réalisé.

4/ Identifier les différentes étapes de développement

Il est conseillé d'identifier 3 étapes de développement (1 par semaine).

Ce dossier est à remettre au plus tard le Mercredi 26 Janvier à 21h00.

Vous devez déposer ce dossier sur la plate-forme **lecnam.net**, dans la section PROJET dans **Dépot du "dossier du projet" (version intermédiaire)**.

2.2. *Faire le développement*

Si cela est nécessaire, vous devez mettre à jour le dossier du projet, en fonction du développement que vous réalisez.

Vous devez remettre un fichier zip contenant :

- l'ensemble des sources
- la documentation Java si vous avez décidé d'en faire une
- le dossier du projet à jour.

	<p style="text-align: center;">Concepts de programmation avancés</p> <p style="text-align: center;">Projet de NFP 121</p> <p style="text-align: center;">2021-2022</p>	<p style="text-align: right;">21/01/2022</p> <p style="text-align: right;">Par J. LAFORGUE</p>
--	---	--

NB : J'ai fait évoluer l'outil de Formulaire afin d'ajouter la méthode **setImage** afin de pouvoir changer l'image d'une image qui a été créée avec la méthode **addImage**.



3. BESIONS COMPLEMENTAIRES

Ces besoins complémentaires ne sont pas obligatoires. Vous choisissez ce que vous voulez faire en fonction du temps qu'il vous reste.

Vous ne faites ces besoins complémentaires qu'une fois le projet correspondant au besoin du §1 terminé.

La réalisation de ces besoins complémentaires, vous fait gagner des points.

Vous devez préciser dans le dossier du projet quels sont, de ces besoins complémentaires, ceux que vous avez réalisés.

3.1. **Déterminer les actions possibles d'un joueur**

Il s'agit d'inhiber et d'activer les boutons d'un joueur en fonction des règles du jeu.

3.2. **Fil de discussion des joueurs**

Il s'agit de permettre aux joueurs de discuter entre eux en mode full. N'importe quel joueur peut taper un texte qui sera vu par tous les joueurs. Chaque message est précédé par le nom de l'émetteur dans le fil de discussion.

3.3. **Déterminer quel joueur a la meilleure main**

Quand il reste au moins 2 joueurs et une fois que le tour de mise est terminé suite à la Rivière, le programme détermine quel joueur a la meilleure main. Le programme donne automatiquement le pot au joueur gagnant en traitant le cas des « tapis ». En cas d'égalité entre deux joueurs, il partage le pot en deux.

3.4. **Remplacer un joueur par un robot**

Il s'agit de remplacer un joueur par un robot. Il ne s'agit pas de faire en sorte que le robot soit entièrement autonome.

A ce titre, on peut aussi simplifier, les règles du jeu :

- pas de blind : un joueur mise ou abandonne la partie (un joueur ne peut pas passer)
 - pas de « tapis »
 - le joueur humain est toujours le donneur
 - le premier qui mise est le donneur
 - ne se joue que à 2 joueurs (le robot et un joueur)
 - ~~le robot mise toujours le même nombre de jetons à chaque tour de mise~~
 - le robot mise si il a une bonne main sinon il abandonne la partie
 - il mise le même nombre de jeton que le joueur humain a misé
 - une bonne main du robot est d'avoir, dans au moins une des configurations :
 - ses 2 cartes
 - ses 2 cartes + flop
 - 6 configurations de 5 cartes parmi : cartes[1] + cartes[2] + flop[1] + flop[2] + flop[3] + turn
 - 15 configurations de 5 cartes parmi : cartes[1] + cartes[2] + flop[1] + flop[2] + flop[3] + turn + river,
- une de ces mains :
- quinte flush (suite + couleur)
 - couleur
 - carré
 - full (brelan+ paire)
 - suite
 - brelan
 - double paire
 - paire
 - ses 2 cartes sont de la même couleur

	<p style="text-align: center;">Concepts de programmation avancés</p> <p style="text-align: center;">Projet de NFP 121</p> <p style="text-align: center;">2021-2022</p>	<p style="text-align: right;">21/01/2022</p> <p style="text-align: right;">Par J. LAFORGUE</p>
--	---	--

- ses 2 cartes contiennent au moins 1 As ou 1 Roi.

3.5. Autres améliorations

Toutes améliorations, expliquées et justifiées.

	Concepts de programmation avancés Projet de NFP 121 2021-2022	21/01/2022 Par J. LAFORGUE
--	--	--------------------------------------

4. QUELQUES PRECISIONS

Les règles de codage que vous devez suivre sont :

- sauf pour les constantes, les attributs de toutes les classes doivent être privés
- toutes les classes commencent par une majuscule
- tous les attributs, méthodes, paramètres et variables commencent par une minuscule
- les noms de tous les répertoires de package sont en minuscule
- les packages créés commencent tous par "fr.cnam"
- une indentation stricte est respectée dans tout le code
- un minimum de commentaire est nécessaire afin de comprendre ce que fait le code. Surtout quand ce que fait le code n'est pas décrit précisément dans le sujet.

Conseils :

- Ecrivez peu de code à chaque fois afin d'avoir toujours un code qui se compile correctement
- Après chaque étape ou sous-étape, le programme s'exécutant correctement, faites une sauvegarde des sources. Cela permet de pouvoir revenir en arrière si besoin et permet aussi de pouvoir donner, si besoin, à l'enseignant une version du programme qui s'exécute correctement et qui remplit correctement une partie des fonctionnalités demandées dans le sujet.
- Dans un premier temps ne faites pas plus que le sujet ne le demande. Une fois que le programme respecte le sujet, vous faites une sauvegarde et vous pouvez apporter des améliorations qui sont toujours appréciées par le correcteur, à condition qu'elles soient bien commentées.

5. PLANNING

Vous avez 3 séances de TP pour réaliser ce projet

Dans la section « PROJET » de la plate-forme **lecnam.net**, j'ai créé deux dépôts intermédiaire, **ETAPE 1** et **ETAPE 2**, vous permettant de déposer une version intermédiaire de votre projet.

Dans la section « PROJET », le dépôt **ETAPE DEFINITIVE** est le dépôt de votre projet définitif qui sera noté.

A REMETTRE avant le Mercredi 16 Février 2021 Minuit.

Vous avez donc **1 mois** pour réaliser l'ensemble du projet.

Je rappelle que, sauf situation exceptionnelle, il est obligatoire d'être présent durant les séances de TP.