**RAPPORT PROJET POKER**

* **Présentation du projet**

Pour notre projet, nous avons décidé de créer un jeu de poker. Après avoir comparé nos différentes idées et leurs avantages et difficultés, nous avons finalement décidé de faire un jeu de poker entre un joueur humain et 5 « joueurs ordinateurs », avec des niveaux de difficulté différents.

* **Description de l’algorithme**

Pour jouer au poker, il faut principalement des joueurs, un paquet de cartes, et une table. Ainsi, les classes qui composent notre code sont :

• **Carte :** chacune ayant sa valeur, sa couleur, un icone (son image), et un boolean pour savoir si la carte est retournée ou pas.

• **Paquet :** une LinkedList avec les 52 cartes classiques d’un jeu de poker

• **Hand :** tient compte des 2 cartes du joueur et des cartes sur la table pour calculer le score maximal du joueur.

• **Joueur :** possède comme attributs caractéristiques le nom du joueur (String), sa quantité d’argent (int), et sa hand (de type Hand). Mais aussi cette classe a d’autres attributs, notamment des booleans, qui nous permettent de connaître l’état du joueur et d’organiser le jeu.

• **Ordinateur :** hérite de Joueur et possède un niveau (int) comme unique attribut.

• **Intelligence :** cette classe permet de prendre les décisions des « joueurs ordinateurs » selon leur niveau.

• **Jeu :** c’est la classe qui permet la gestion, l’organisation et le développement du jeu.

• **FenetreJeu :** permet de représenter le jeu sous forme d’interface graphique.

• **Ecouteur :** rend les boutons fonctionnels, pour la prise de décision du joueur humain {Raise, Call, Check) et pour passer à la tournée suivante.

• **Node :** pour éviter qu’il y ait des joueurs répétés.

• **LinkedListCirculaire :** cette classe nous a beaucoup facilité le fait de créer un jeu fluide, en optimisant le passage entre tournées.

Grâce à une véritable réflexion pendant l’étape de préconception, nous sommes bien partis dès le début. En effet, parmi les classes citées ci-dessus, seulement les classes Node et LinkedListCirculaire n'étaient pas prévues. En outre, pendant le développement du code, nous avons compris certaines des classes envisagées au début n’étaient pas utiles : c’est le cas d’une class Table, qui allait avoir les cartes du milieu, et une class Banque, qui allait être responsable de gérer l’argent des joueurs, les paris, et le pot.

* **Diagramme UML**

Ainsi, le diagramme UML de notre code final est le suivant.

* **Performance du groupe**
* **Eléments récupérés**

Au cours du développement de l’algorithme, nous nous sommes parfois inspirés de codes trouvés sur internet. Effectivement, pour réussir à coder notre jeu de poker, il nous a fallu repousser nos limites et acquérir de nouvelles connaissances. Mais aussi, à certains moments, nous avons eu besoin d’une fonction spécifique, que nous avons récupéré sur internet. Ainsi, les éléments récupérés tous faits sont :

* **Principaux problèmes résolus**

Sans doute, nous avons rencontré de diverses difficultés pendant la réalisation d’un algorithme si complexe. Les problèmes les plus intéressants que nous avons résolus sont : (algorithme non trivial ou structure de donnée intéressante)

* **Organisation**

Même avant le début du confinement, nous avons mis en place un système de travail collaboratif grâce à l’utilisation de la plateforme Github. En effet, deux d’entre nous connaissaient déjà ce mode de travail et donc le confinement n’a pas vraiment affecté notre progrès, même si le fait de ne pas discuter en personne est moins avantageux. Avec le cours du temps, les autres deux membres du groupe se sont habitués au fonctionnement de Github. Également, même en dehors des heures de cours, nous nous communiquions de manière régulière, principalement avec des messages à travers l’application WhatsApp, mais aussi avec des appels grâce à Discord quand il nous fallait faire le point. En moyenne, on discutait du projet chaque 2 ou 3 jours : nous étions tous les quatre passionnés et engagés pour créer le meilleur programme possible.

* **Semainier**
* **Forces et faiblesses**
* Sans doute, le principal atout de notre projet c’est le fait que le jeu de poker est totalement fonctionnel. Nous avons bien progressé chaque semaine, ce qui nous a permis d’améliorer la performance du code et d’éliminer les bugs rencontrés à chaque étape. ETC
* Nous aurons voulu travailler plus l’interface graphique. En effet, notre version est la plus efficace et non pas la plus attractive visuellement. ETC
* Notre chef de projet a très bien organisé et coordonné le travail à réaliser. Il nous a incité et motivé à surmonter les obstacles rencontrés. En plus d’être un excellent programmeur, il a été un excellent leader.
* Taux d’implication : BARROS Matheus (%) ; BERTOLDI Gustavo (%) ; CADER Matias (%); FRANCO Carlo (%).
* Ainsi, par tout ce qui précède, et surtout conscients des aspects qui pourraient être améliorés, nous considérons que nous méritons une note de 20000/20.
* **Bibliographie**