v1.0 CardYard

FILIERE préING1 • 2024-2025
AUTEURS E.ANSERMIN – R. GRIGNON
E-MAILS eva.ansermin@cyu.fr – romuald.grignon@cyu.fr

DESCRIPTION DU PROJET

- Ce projet est un jeu de cartes dans lequel les différents joueurs ont tous la même vue des différents tas de cartes. Les cartes comportent des valeurs numériques et le but est d'avoir le moins de points à la fin de la partie.
- En début de partie, chaque joueur dispose d'un nombre de **cartes personnelles**, toutes étalées faces cachées devant lui, ainsi qu'un tas de cartes faces visibles servant de **défausse** (initialement vide). Il y a également une **pioche centrale** avec le reste des cartes, faces cachées.
- A tour de rôle, les joueurs vont effectuer les actions suivantes :
  - piocher une carte depuis la pioche centrale **ou** depuis n'importe quelle défausse
  - échanger cette carte avec l'une des cartes étalées devant le joueur (qu'elle soit visible ou non)
  - la carte échangée est placée face visible sur la défausse du joueur
- Le jeu se termine lorsqu'un joueur a rendu visible toutes ses cartes étalées : à ce moment là, toutes les cartes étalées des joueurs sont retournées, chaque joueur fait sa somme et le plus petit score gagne la partie.
- Par défaut, le nombre de cartes du jeu est déterminé à l'avance :
  - Les valeurs vont de **-2** à **+12**
  - Il n'y a que **5 cartes** pour la **valeur -2**
  - Il y a **10 cartes** pour la **valeur -1**
  - Il y a **15 cartes** pour la **valeur 0**
  - Il y a **10 cartes** de chaque **valeur positive**

INFORMATIONS GENERALES

- **Taille de l'équipe**
  Ce projet est un travail d'équipe. Il est autorisé de se réunir en groupe de 3 au maximum. Si le nombre total d'étudiants n'est pas un multiple de 3 et/ou si des étudiants n'arrivent pas à constituer des groupes, c'est au chargé de TD de statuer sur la formation des groupes. Pensez-donc à anticiper la constitution de vos groupes pour éviter des décisions malheureuses.

- **Démarrage du projet**
  Vous obtiendrez de plus amples informations quant aux dates précises de rendu, de soutenance, les critères d'évaluation, le contenu du livrable, ..., quand le projet démarrera officiellement.
- **Dépôt de code**
  Vous devrez déposer la totalité des fichiers de votre projet sur un dépôt central Git. Il en existe plusieurs disponibles gratuitement sur des sites web comme ***github.com*** ou ***gitlab.com***.
- **Rapport du projet**
  Un rapport écrit est requis, contenant une brève description de l'équipe et du sujet. Il décrira les différents problèmes rencontrés, les solutions apportées et les résultats. L'idée principale est de montrer comment l'équipe s'est organisée, et quel était le flux de travail appliqué pour atteindre les objectifs du cahier des charges. Le rapport du projet peut être rédigé en français.
- **Démonstration**
  Le jour de la présentation de votre projet, votre code sera exécuté sur la machine de votre chargé(e) de TD. La version utilisée sera la **dernière** fournie sur le dépôt Git **avant** la date de rendu. Même si vous avez une nouvelle version qui corrige des erreurs ou implémente de nouvelles fonctionnalités le jour de la démonstration, c'est bien la version du rendu qui sera utilisée.
- **Variantes**
  Lors de votre choix de projet, votre chargé(e) de TD vous donnera la ou les variantes que vous devrez implémenter.
- **Organisation**
  Votre projet complet devrait **(dans l'idéal)** être stocké sur un **dépôt git** (ou un outil similaire) tout au long du projet pour au moins trois raisons : éviter de perdre du travail tout au long du développement de votre application, être capable de travailler efficacement en équipe, et partager vos progrès de développement facilement avec votre chargé de projet. De plus il est **recommandé** de mettre en place un **environnement** de travail en **équipe** en utilisant divers outils pour cela (Slack, Trello, Discord, ...)

CRITERES GENERAUX

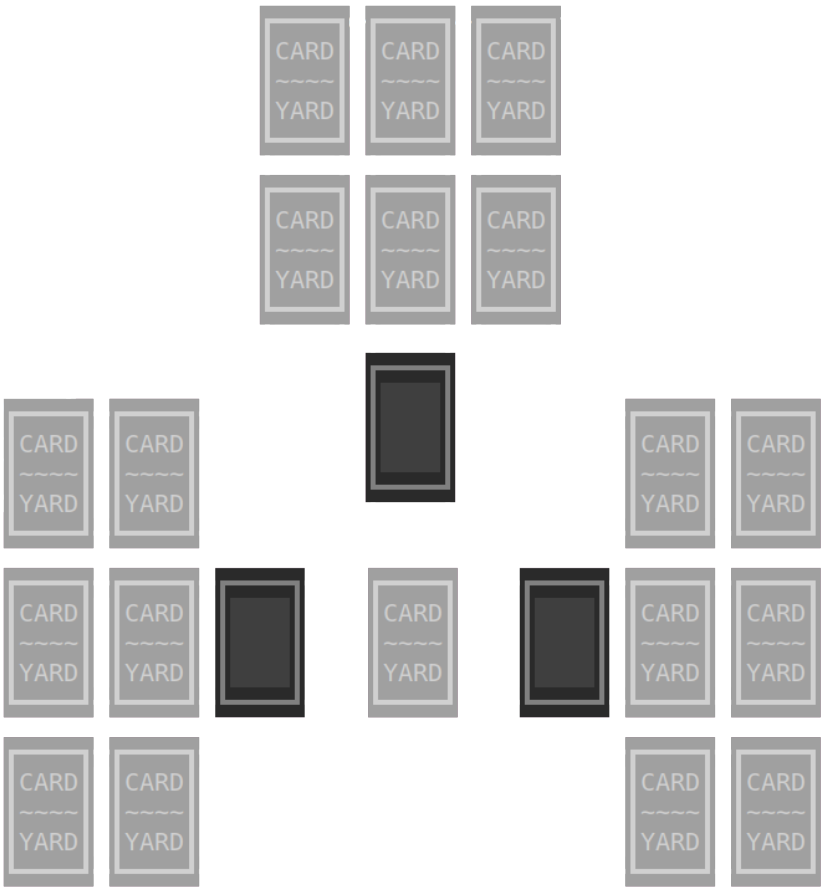
- Le **but principal** du projet est de fournir une **application fonctionnelle** pour l'utilisateur. Le programme doit correspondre à la description en début de document et implémenter toutes les fonctionnalités listées.
- Votre code sera généreusement **commenté**.
- Tous les éléments de **votre code** (variables, fonctions, commentaires) seront écrits dans la **même langue**. Langue anglaise conseillée mais pas obligatoire.
- Votre application ne doit jamais s'interrompre de manière intempestive (crash), ou tourner en boucle indéfiniment, quelle que soit la raison. Toutes les erreurs doivent être gérées correctement. Il est préférable de d'avoir une application stable avec moins de fonctionnalités plutôt qu'une application contenant toutes les exigences du cahier des charges mais qui plante trop souvent. Une application qui se stoppe de manière imprévue à cause d'une erreur

de segmentation ou d'une exception, par exemple, sera un événement très pénalisant.

- Votre application devra être **modulée** afin de ne pas avoir l'ensemble du code dans un seul et même fichier par exemple. Apportez du soin à la conception de votre projet avant de vous lancer dans le code.
- Le livrable fourni à votre chargé(e) de TD sera simplement l'**URL** de votre **dépôt Git** accessible **publiquement**. Même si vous n'avez pas utilisé ce dépôt régulièrement au cours du projet, le code final sera livré dessus.

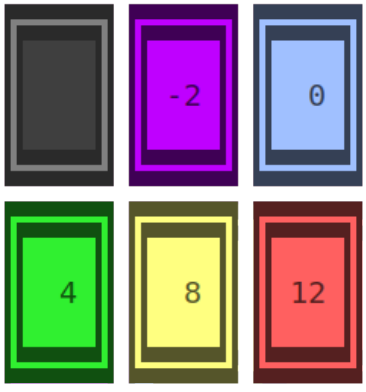
FONCTIONNALITES DU PROJET

- Le jeu se joue forcément à plusieurs : à vous de voir quelles sont les limites possibles : un intervalle conseillé serait [2-8] joueurs. C'est en début de partie que l'on va sélectionner ce nombre.
- Le jeu va charger en mémoire les différentes zones de jeu (pioche centrale, défausses de chaque joueur, zone des cartes de chaque joueur) et initialiser correctement.
  Ci-dessous un exemple d'affichage possible pour 3 joueurs avec chaque joueur ayant 6 cartes. On voit la pioche centrale face cachée, les 6 cartes de chaque joueur faces cachées, et les 3 défausses (1 pour chaque joueur) vides.



La représentation de chaque carte ci-dessus est réalisée avec des caractères du terminal (bordures) et un jeu de couleurs entre le texte et le fond (seul le placement ici n'est pas réellement tiré d'un terminal).

- Le programme va demander à chaque joueur à tour de rôle de :
  - piocher une carte depuis la **pioche centrale**…
    le joueur doit alors choisir quelle carte personnelle doit être échangée avec cette carte : cela peut etre une carte déjà visible ou une autre. La carte personnelle sera positionnée sur la défausse du joueur, face visible, et la carte piochée sera mise en remplacement face visible également.
    Si le joueur ne veut pas échanger la carte de la pioche centrale, il peut choisir de la mettre sur sa pile de défausse.
  - … ou prendre une carte visible depuis l'une des **défausses** si il y en a. Cette carte doit être échangée avec une des cartes personnelles du joueur comme expliqué ci-dessus
- Le jeu passe de joueur en joueur jusqu'à ce qu'un joueur ait rendu visible toutes ses cartes personnelles. A ce moment précis, les autres joueurs doivent encore jouer 1 fois chacun (peu importe si le joueur actuel, qui a terminé, n'etait pas le premier au tout début du jeu).
  Une fois la partie terminée, les cartes personnelles sont toutes retournées et la somme des valeurs de ces cartes donne le score du joueur. Ces scores sont triés par ordre croissant, le plus petit score étant le gagnant.
  Les noms des joueurs et leurs scores, dans l'ordre croissant sont affichés. La partie est terminée, le programme peut repasser au menu de départ pour choisir le nombre de joueurs d'une partie et en relancer une autre.
  Pour le déroulement du jeu, il faudra veiller à ce que les pioches dans lesquelles vous voulez prendre une carte notamment celles de la défausse ne soient pas vides. La pioche centrale serait moins propice à cette situation mais il vous appartient de le vérifier quand même dans le programme.
- **Visuel** : chaque valeur de carte possède une couleur différente des autres pour pouvoir distinguer plus facilement les cartes qui coutent cher et celles qu'il faut conserver pour gagner. Essayez également d'afficher les cartes de manière stylisée autant que possible (cadres, fonds) même si le terminal reste limité.
  Dans le cas des pioches, il vous appartient d'afficher la bonne valeur de carte posée sur le dessus, et de pouvoir afficher une pile vide si l'on reprend toutes les cartes du dessus.



- **Sauvegarde** : à tout moment d'une partie, on peut décider de stopper la partie et de la sauvegarder. A ce moment là, une fois la sauvegarde effectuée correctement, le programme peut s'arrêter.
  En relançant le programme, sur le menu de démarrage, au lieu de démarrer une nouvelle partie, on peut choisir de restaurer une partie précédente sauvegardée dans un fichier.
  Le format exact des sauvegardes dans les fichiers est totalement libre. Vous pouvez décider de le faire au format ASCII ou binaire, dans un seul ou plusieurs fichiers.

VARIANTES

- **Valeurs des cartes**:
  - variante **VALUE_FILE** : les valeurs et quantités de chaque carte sont fixées dans un fichier (de fait on peut modifier le jeu pour avoir un carte +20 au lieu du +12 par exemple)
  - variante **VALUE_USER** : les valeurs des différentes cartes sont demandées au démarrage du programme (soit une plage de valeurs [-5, 15] ou chaque valeur de carte, une par une.
- **Nombre de cartes personnelles**
  - variante **CARD_USER** : le nombre de cartes personnelles de chaque joueur est demandé au démarrage de la partie.
  - variante **CARD_RAND** : le nombre de cartes personnelles est fixé au hasard.
  - attention à l'affichage, vous pouvez restreindre les valeurs possibles pour simplifier un peu l'affichage

RESSOURCES UTILES

- **Github**
  https://www.github.com
  https://docs.github.com/en/get-started/quickstart/hello-world
- **Couleurs dans le terminal**
  http://sdz.tdct.org/sdz/des-couleurs-dans-la-console-linux.html
- **Emojis dans le terminal**
  https://unicode.org/emoji/charts/full-emoji-list.html
- **Symboles de bordures dans le terminal**
  https://www.w3.org/TR/xml-entity-names/025.html