

# RAPPORT DE PROJET

## LUDOCIEL

L3 économie appliquée  
Mme. PAIN



Ludociel propose une expérience dynamique pour les passionnés de jeux de société, avec deux bibliothèques distinctes : partagée et personnelle. Les utilisateurs contribuent facilement à ce jeu de données collaboratif en remplissant un formulaire, et des filtres efficaces simplifiant la recherche. Ludociel vise ainsi à créer une plateforme centralisée pour les amateurs de jeux de société.



# SOMMAIRE

<b>1. INTRODUCTION .....</b>	<b>4</b>
<b>2. PAGE D'ACCUEIL.....</b>	<b>5</b>
2.1    QUI SOMMES-NOUS ? .....	5
2.2    INSCRIPTION .....	5
2.3    AJOUTER UN JEU.....	6
<b>3. BIBLIOTHEQUE PARTAGEE .....</b>	<b>7</b>
3.1    LE BANDEAU .....	7
3.2    LISTE DES JEUX .....	7
3.3    FILTRES.....	8
<b>4. BIBLIOTHEQUE PERSONNELLE .....</b>	<b>9</b>
4.1    S'IDENTIFIER .....	9
4.2    BANDEAU.....	9
4.3    ACCES AUX JEUX.....	9
<b>5. DICTIONNAIRE DES OBJETS .....</b>	<b>10</b>
<b>6. JUSTIFICATION DU CODE.....</b>	<b>12</b>
6.1    LES MODULES .....	12
6.1.1 <i>DéclarationVariables</i> .....	12
6.1.2 <i>OuvrirUsf</i> .....	12
6.1.3 <i>AjouterUnJeu</i> .....	13
6.1.4 <i>Filtres</i> .....	13
6.2    LES USERFORMS .....	13
6.2.1 <i>usfInscription</i> .....	14
6.2.2 <i>usfAjouterUnJeu</i> .....	15
6.2.3 <i>usfJeuExistant</i> .....	18
6.2.4 <i>usfConnexion</i> .....	19
<b>7. LES AXES D'AMELIORATION.....</b>	<b>20</b>
<b>8. CONCLUSION .....</b>	<b>21</b>

# 1.

## INTRODUCTION

***Comment accéder aux jeux de sociétés que l'on possède et qui nous ont marqué tout en découvrant des nouveautés ?***

C'est à cette question que nous avons souhaité répondre avec notre application : Ludociel.

Intuitive, dynamique et colorée, cette application a pour objectif d'aider les amateurs de jeux de société à consulter et agrandir leur collection. Pour cela, ils ont accès à deux bibliothèques distinctes.

La première est une bibliothèque partagée avec l'ensemble des utilisateurs. Chaque jeu de société ajouté est enregistré directement dans celle-ci. La deuxième bibliothèque est la bibliothèque personnelle de l'utilisateur.

Pour enrichir la bibliothèque, il suffit de remplir un formulaire et de renseigner quelques informations nécessaires relatives au jeu en question. Enfin, En utilisant les filtres de recherche, retrouver un jeu déjà enregistré est simple et rapide.

En somme, nous envisageons de disposer d'un vaste jeu de données, regroupant les informations de différents utilisateurs ainsi que leurs jeux de société. Les différentes étapes pour accéder à la bibliothèque partagée ou personnelle seront simplement des filtres appliqués à ce jeu de données.

# 2.

# PAGE D'ACCUEIL

En utilisant des boutons cliquables, la page d'accueil de Ludociel offre un accès aux deux bibliothèques de l'application, permettant ainsi de consulter la bibliothèque partagée de l'application ainsi que sa bibliothèque personnelle (les détails de leur fonctionnement seront expliqués dans les prochaines sections). Il est également possible d'ajouter un jeu de société directement depuis la page d'accueil. Pour bénéficier de ces fonctionnalités, l'utilisateur doit être identifiable, ce qui nécessite son inscription s'il ne l'a pas déjà fait. De plus, un bandeau à gauche de l'interface présente le logo de l'application ainsi qu'un bouton cliquable intitulé « Qui sommes-nous ? ».



## 2.1 QUI SOMMES-NOUS ?

Il est possible d'en apprendre davantage sur les créatrices de l'application en cliquant sur « Qui sommes-nous ? ». Cela ouvre un UserForm avec un petit texte expliquant les origines de l'application et son but. On retrouve aussi une brève présentation d'Emma Bougeard et Louan Paranthoen.

La seule action que l'utilisateur peut effectuer sur la page est de cliquer sur le bouton « Fermer ».

## 2.2 INSCRIPTION

Afin d'ajouter un jeu à la bibliothèque, l'utilisateur doit être identifiable. Pour ce faire, il doit s'inscrire s'il ne l'a pas déjà fait. L'inscription implique le choix d'un pseudo, qui sera associé à une sélection de jeux spécifique. Cette action se déroule au moyen d'un UserForm comprenant une TextBox. L'utilisateur doit donc saisir un pseudo. Pour éviter toute confusion. Le pseudo doit être unique.

Si le pseudo choisi est déjà utilisé, une MessageBox s'ouvre au moment de l'enregistrement et demande à l'utilisateur d'en choisir un nouveau. En cliquant sur "OK", la MessageBox se ferme, le pseudo choisi précédemment est effacé de la TextBox, et l'utilisateur peut en choisir un autre.

Une fois inscrit, la personne peut accéder à sa bibliothèque et ajouter des jeux. Seul l'accès à la bibliothèque partagée ne nécessite pas d'identification.

## 2.3 AJOUTER UN JEU

L'utilisateur peut cliquer sur le bouton "Ajouter un nouveau jeu". Une fenêtre s'ouvre (un Userform), semblable à un formulaire, dans lequel il peut saisir plusieurs informations. Certaines sont obligatoires, d'autres facultatives. De plus, des restrictions sont appliquées aux zones de texte pour éviter l'ajout de symboles, par exemple.

### PSEUDO (obligatoire)

Le pseudo est sélectionné à partir d'une liste déroulante comportant l'ensemble des pseudos déjà enregistrés. Il suffit de choisir son pseudo individuel.

### NOM (obligatoire)

L'utilisateur renseigne le nom du jeu de société dans une zone de texte à l'aide de la fonction "TextBox".

### CATEGORIE (obligatoire)

La catégorie du jeu est également une caractéristique obligatoire à informer. Les catégories disponibles dans la combobox sont :

- Jeux de plateaux
- Jeux de cartes
- Jeux de rôle
- Jeux de stratégie
- Jeux de coopération
- Jeux de quiz et culture générale
- Jeux d'énigme
- Jeux d'ambiance

### NOMBRE DE JOUEURS (obligatoire)

Il doit également spécifier le nombre de joueurs en utilisant des boutons avec une seule sélection possible (« OptionButton ») avec les options suivantes :

- Jeu solitaire
- Jeu en duo
- De 3 à 6 joueurs
- De 6 à 8 joueurs
- 10 joueurs et plus

### CARACTERISTIQUES FACULTATIVES

En ce qui concerne la durée de la partie et l'âge requis, rien n'est obligatoire. L'utilisateur peut, s'il le souhaite, renseigner ses informations de la même manière que le nombre de joueurs.

Les âges proposés sont : 3, 6, 12 et 16 ans. Quant à elle, la durée de la partie peut être de moins de 15 minutes, de 15 à 30 minutes, de 30 minutes à 1 heure et de plus d'une heure.

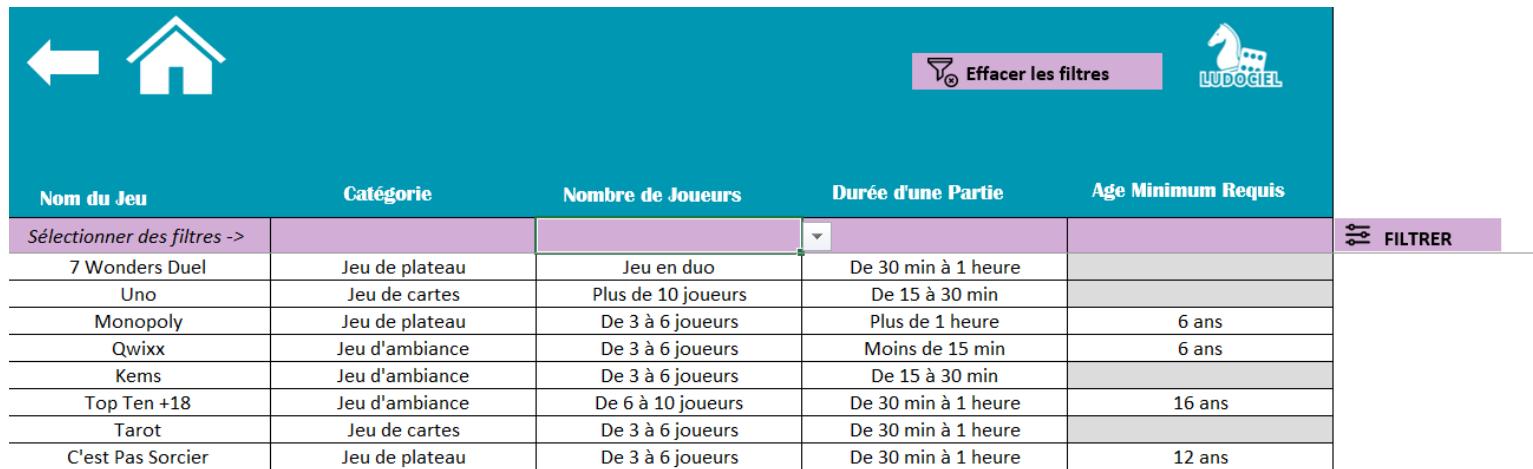
### ENREGISTREMENT DES DONNES

En conclusion, toutes les informations seront enregistrées dans la feuille "Jeu de données" une fois que l'utilisateur a cliqué sur "Enregistrer".

# 3.

## BIBLIOTHEQUE PARATGEE

La bibliothèque partagée est constituée de la liste des jeux de société enregistrés par l'utilisateur actif ou tout autre utilisateur de l'application. Elle rassemble ainsi l'ensemble des bibliothèques personnelles des utilisateurs. Dans cette section, il n'est pas possible d'ajouter un jeu de société. Seule la visualisation des jeux est autorisée, ainsi que le filtrage de ces jeux.



The screenshot shows a user interface for managing a shared library of board games. At the top, there are navigation icons (back, home, search) and a 'LUDOGIEL' logo. Below the header is a search bar with placeholder text 'Rechercher' and a magnifying glass icon. A 'Filtres' button is located to the right of the search bar. The main content area is a table with the following columns: 'Nom du Jeu', 'Catégorie', 'Nombre de Joueurs', 'Durée d'une Partie', and 'Age Minimum Requis'. The table lists various games with their details. A 'Filtrer' button is located at the bottom right of the table.

Nom du Jeu	Catégorie	Nombre de Joueurs	Durée d'une Partie	Age Minimum Requis
Sélectionner des filtres ->				
7 Wonders Duel	Jeu de plateau	Jeu en duo	De 30 min à 1 heure	
Uno	Jeu de cartes	Plus de 10 joueurs	De 15 à 30 min	
Monopoly	Jeu de plateau	De 3 à 6 joueurs	Plus de 1 heure	6 ans
Qwixx	Jeu d'ambiance	De 3 à 6 joueurs	Moins de 15 min	6 ans
Kems	Jeu d'ambiance	De 3 à 6 joueurs	De 15 à 30 min	
Top Ten +18	Jeu d'ambiance	De 6 à 10 joueurs	De 30 min à 1 heure	16 ans
Tarot	Jeu de cartes	De 3 à 6 joueurs	De 30 min à 1 heure	
C'est Pas Sorcier	Jeu de plateau	De 3 à 6 joueurs	De 30 min à 1 heure	12 ans

### 3.1 LE BANDEAU

En haut de la page, un bandeau offre plusieurs options à l'utilisateur. Ce dernier reste affiché à l'écran même lorsque la page évolue. Les volets sont figés. Trois boutons sont présents sur ce bandeau. Le premier, représenté par une icône en forme de flèche, permet de revenir à la page d'accueil. À droite du bandeau, le logo de l'application est également visible. Lorsqu'on clique sur celui-ci, le UserForm « Qui sommes-nous ? » s'affiche.

Juste en dessous de ce bandeau, on retrouve la liste des jeux de la bibliothèque partagée.

### 3.2 LISTE DES JEUX

Cette liste est présentée sous la forme d'un tableau Excel, répertoriant les caractéristiques de chaque jeu :

- Nom
- Catégorie
- Nombre de joueurs
- Durée d'une partie
- Âge requis

Il est possible que certaines informations soient manquantes en fonction des jeux. Cependant, le nom, la catégorie et le nombre de joueurs seront toujours renseignés.

Il est important de noter que chaque jeu, sans que cela soit visible pour l'utilisateur, est également associé aux personnes qui le possèdent dans leur bibliothèque personnelle.

### 3.3 FILTRES

L'objectif de ces filtres est de faciliter la recherche d'un jeu correspondant aux caractéristiques renseignées lors de l'ajout d'un jeu. Il est possible de rechercher un jeu précis ou un jeu comportant certaines propriétés en filtrant la bibliothèque partagée.

Chaque colonne du tableau commence par les filtres. Il est possible de filtrer sa recherche en cliquant sur la première cellule de la colonne. Ainsi, une liste déroulante s'affiche afin de sélectionner les filtres souhaités.

L'utilisateur peut filtrer toutes les colonnes sauf celles contenant le nom des jeux.

Une fois ses choix faits, il doit cliquer sur le bouton cliquable « Filtrer ». Il peut également effacer ses filtres grâce à un autre bouton cliquable : « Effacer les filtres ».





Nom du Jeu	Catégorie	Nombre de Joueurs	Durée d'une Partie	Age Minimum
7 Wonders Duel	Jeu de plateau	Jeu en duo	De 30 min à 1 heure	
Uno	Jeu de cartes	Plus de 10 joueurs	De 15 à 30 min	
Monopoly	Jeu de plateau	De 3 à 6 joueurs	Plus de 1 heure	6 ans
Qwixx	Jeu d'ambiance	De 3 à 6 joueurs	Moins de 15 min	6 ans
Kems	Jeu d'ambiance	De 3 à 6 joueurs	De 15 à 30 min	
Top Ten +18	Jeu d'ambiance	De 6 à 10 joueurs	De 30 min à 1 heure	16 ans

## 4.1 S'IDENTIFIER

Afin d'accéder à sa propre bibliothèque, l'utilisateur choisit son pseudo parmi la liste déroulante. Cette dernière s'ouvre dans un UserForm directement lorsqu'on clique sur « Ma Bibliothèque », depuis la page d'accueil. Une fois la validation faite, la feuille « Ma bibliothèque » s'ouvre.

## 4.2 BANDEAU

Comme sur la bibliothèque partagée, un bandeau en haut de page est fixe et reste affiché même si on descend dans la feuille.

Il est composé du logo de l'application, des titres des colonnes du tableau et d'une flèche cliquable, qui permet de retourner à la page d'accueil.

## 4.3 ACCES AUX JEUX

La bibliothèque prend la forme d'un tableau Excel dont les colonnes présentent les caractéristiques du jeu. Certaines d'entre elles ne sont pas remplies car toutes les informations ne sont pas obligatoires lors de l'enregistrement du jeu.

L'utilisateur visualise l'ensemble des jeux qu'il a lui-même enregistrés ainsi que ceux qu'il a ajouté à sa bibliothèque mais qui existaient déjà dans notre jeu de données.

Les informations sont présentées dans l'ordre suivant : Nom du jeu, Catégorie, Nombre de joueurs, Durée d'un partie et Age minimum.

# 5.

# DICTIONNAIRE DES OBJETS

**usfInscription**  
INSCRIPTION

**lblPseudo**  
Renseignez votre pseudo

**Enregistrer** **cmdEnregistrer**    **Annuler** **cmdAnnuler**

**usfConnexion**  
CONNEXION

**lblPseudo**  
Sélectionner votre Pseudo

**Accéder à ma bibliothèque** **cmdAccederBiblio**

**usfAjouterUnJeu**  
Ajouter un jeu

**lblPseudo**  
Votre pseudo \*

**lblNomDuJeu**  
Nom du jeu \*

**lblCategorie**  
Catégorie du jeu \*

Jeu de plateau  
Jeu de cartes  
Jeu de rôles **lstCategorie**  
Jeu de stratégie  
Jeu de coopération  
Jeu de culture générale

**fraDuree**  
Durée de la partie

Moins de 15 min **optMoinsDe15**  
 De 15 à 30 min **opt1530**  
 De 30 min à 1 heure **opt3060**  
 Plus de 1 heure **optPlusDe60**

**fraNbreDeJoueurs**  
Nombre De Joueurs\*

Jeu en solitaire **optSolitaire**  
 Jeu en duo **optDuo**  
 De 3 à 6 joueurs **opt3a6**  
 De 6 à 10 joueurs **opt6a10**  
 Plus de 10 joueurs **optPlusDe10**

**fraAge**  
Age minimal pour jouer

3 ans **opt3ans**  
 6 ans **opt6**  
 12 ans **opt12**  
 16 ans **opt16**

**ENREGISTRER** **cmdEnregistrer**    **QUITTER** **cmdQuitter**

**lblChampsObligatoire**  
\*Champs obligatoire à renseigner

**usfJeuExistant**  
AJOUTER UN JEU A SA PROPRE BIBLIOTHEQUE X

**IblChoixPseudo**  
Veuillez choisir votre pseudo

**IblSelectionnerJeu**  
Veuillez sélectionner des jeux

7 Wonders  
Chaoullite  
Monopoly  
Fvq  
Ùlikjn,fdxsolllllllx  
Tjhndg  
Hcnfc  
Mopo  
Dsgmlhfbo  
Ezrgezthz  
Ergea **IstSelectionJeu**  
Egz  
Egr  
Aerta

**IblJeuAjoute**  
Jeux ajoutés à votre bibliothèque

**IstJeuChoisi**

**Ajouter les jeux à la liste** **cmdAjouterJeu** **Supprimer** **cmdSupprimer**

**Enregistrer** **cmdEnregistrer** **Annuler** **cmdAnnuler**

**usfQuiSommesNous**  
Qui sommes-nous ? X

**IblNaissance**  
La naissance de Ludociel

**imaLouan**   
**IblLouan** Louan

**imaEmma**   
**IblEmma** Emma

Emma et Louan ont crée Ludociel en 2023. Toutes les deux étudiantes en économie, elles sont passionnées depuis petites par les jeux de sociétés et les activités ludiques. En se retrouvant régulièrement pour partager leur collection, elles eurent l'idée d'une application permettant à chacun de découvrir des jeux de société mais aussi de répertorier et agrandir sa collection.

**IblTexte**

# 6.

# JUSTIFICATION DU CODE

*A noter que dans chacun des modules et chacun des UserForm, « Option Explicit » est toujours présent permettant d'obliger la déclaration de variables. Cela nous a permis de voir des erreurs plus facilement.*

## 6.1 LES MODULES

### 6.1.1 DéclarationVariables

```
Option Explicit

'Compte le nb de lignes pour chacune des feuilles
Public LigneJeuDonne As Integer
Public LigneBiblioPartage As Integer
Public LignePseudo As Integer
Public LigneMaBiblio As Integer

' Définir la feuille
Public wsJeuDeDonne As Worksheet
Public wsPseudo As Worksheet
Public wsMaBiblio As Worksheet
Public wsBiblioPartage As Worksheet

Public Sub CompteLigne()

    Set wsJeuDeDonne = ThisWorkbook.Worksheets("Jeu de données")
    Set wsPseudo = ThisWorkbook.Worksheets("Pseudo")
    Set wsMaBiblio = ThisWorkbook.Worksheets("Ma Bibliothèque")
    Set wsBiblioPartage = ThisWorkbook.Worksheets("Bibliothèque partagée")

    LigneJeuDonne = ThisWorkbook.Worksheets("Jeu de données").Cells(wsJeuDeDonne.Rows.Count, 1).End(xlUp).Row
    LigneBiblioPartage = ThisWorkbook.Worksheets("Bibliothèque partagée").Cells(wsBiblioPartage.Rows.Count, 1).End(xlUp).Row
    LignePseudo = ThisWorkbook.Worksheets("Pseudo").Cells(wsPseudo.Rows.Count, 1).End(xlUp).Row
    LigneMaBiblio = ThisWorkbook.Worksheets("Ma Bibliothèque").Cells(wsMaBiblio.Rows.Count, 1).End(xlUp).Row

End Sub

'   ' fait appel à une procédure qui compte les lignes
'   Call CompteLigne
```

Nous avons fait le choix de déclarer les variables en public. Cette approche implique que la valeur attribuée à chaque variable reste en mémoire pour chaque utilisation ultérieure. Nous avons appliqué cette méthode aux noms des feuilles pour éviter de devoir constamment réécrire la feuille utilisée dans le code futur. De plus, nous avons adopté cette approche lors du comptage de lignes. En d'autres termes, chaque fois que la variable « ligne » est utilisée, elle reprend la valeur mémorisée précédemment. De manière similaire, les variables « ligne » prennent la valeur du comptage en commençant par le bas et en remontant jusqu'à ce qu'une cellule non vide soit rencontrée. En conclusion, nous invoquons cette procédure avec la commande « Call CompteLigne » (qui est le nom de la procédure). En déclarant cette procédure comme publique, nous permettons ainsi à d'autres modules ou UserForms de réutiliser ces valeurs et de les mettre à jour.

La ligne de commentaire sous cette procédure sera la ligne utilisée lorsque la procédure sera appelée.

### 6.1.2 OuvrirUsf

Ce module a pour fonction d'ouvrir tous les UserForms et d'accéder aux différentes feuilles. Nous avons veillé à ce que lorsqu'une feuille est affichée, les autres soient masquées, et dès que l'on accède à une autre feuille, le processus se répète.

De plus, à partir de ce module, nous avons pris des dispositions pour figer les volets de la première ligne de la feuille "Bibliothèque partagée".

### 6.1.3 AjouterUnJeu

Ce module a pour fonction d'ouvrir des boîtes de dialogue (MessageBox) afin d'orienter l'utilisateur lorsqu'il clique sur "Ajouter un jeu".

### 6.1.4 Filtres

Ce module offre la possibilité d'appliquer des filtres dans la feuille "Bibliothèque Partagée". Une première procédure est dédiée à la réinitialisation des filtres. Ensuite, une seconde procédure (celle ci-dessous), activée lorsque le bouton « Filtrer » est cliqué, permet de filtrer le jeu de données en fonction des filtres choisis.

```
Sub FiltrerLesDonnées()

    Dim PlageDonne As Range
    Dim FiltreCategorie As String, FiltreNbJoueur As String, FiltreAge As String, FiltreDuree As String
    Dim i As Integer

    Set PlageDonne = ThisWorkbook.Worksheets("Bibliothèque partagée").Range("B3:B1000")

    ' réinitialise les lignes masqués
    PlageDonne.EntireRow.Hidden = False

    ' fait appel à une procédure qui compte les lignes
    Call CompteLigne

    ' récupérer les critères de filtre
    FiltreCategorie = wsBiblioPartage.Range("B2").Value
    FiltreNbJoueur = wsBiblioPartage.Range("C2").Value
    FiltreDuree = wsBiblioPartage.Range("D2").Value
    FiltreAge = wsBiblioPartage.Range("E2").Value

    ' parcourir les lignes et masquer celles qui ne correspondent pas aux filtres
    For i = 3 To LigneBiblioPartage + 1

        If Not (wsBiblioPartage.Cells(i, 2).Value = FiltreCategorie Or FiltreCategorie = "") Then
            wsBiblioPartage.Rows(i).Hidden = True
        ElseIf Not (wsBiblioPartage.Cells(i, 3).Value = FiltreNbJoueur Or FiltreNbJoueur = "") Then
            wsBiblioPartage.Rows(i).Hidden = True
        ElseIf Not (wsBiblioPartage.Cells(i, 4).Value = FiltreDuree Or FiltreDuree = "") Then
            wsBiblioPartage.Rows(i).Hidden = True
        ElseIf Not (wsBiblioPartage.Cells(i, 5).Value = FiltreAge Or FiltreAge = "") Then
            wsBiblioPartage.Rows(i).Hidden = True
        End If

    Next i

End Sub
```

Le début du code offre la possibilité à l'utilisateur d'ajouter un filtre tout en permettant la réinitialisation et la réapplication des filtres précédemment choisis. Cela évite à l'utilisateur d'effacer tous les filtres pour les sélectionner à nouveau. Il est important de noter que l'utilisateur n'est pas contraint de sélectionner tous les filtres ; il peut choisir ceux qu'il souhaite.

La structure de boucle "For" parcourt toutes les cellules de la ligne 3 jusqu'à la dernière cellule non vide + 1. À l'intérieur de cette boucle, une structure conditionnelle est intégrée pour vérifier si la valeur dans la cellule des filtres est égale à la valeur de la ligne. Ce processus est répété pour chaque ligne et chaque colonne. Si cette condition n'est pas satisfaite, la ligne est masquée. De plus, la structure conditionnelle vérifie si la cellule de filtre est vide et si la ligne i est égale à la cellule de filtre. Si l'une des deux conditions n'est pas vérifiée, la ligne est masquée. Cette approche permet d'éviter le filtrage des données lorsque la cellule de filtre est vide.

## 6.2 LES USERFORMS

## 6.2.1 usfInscription

---

Nous avons opté pour que chaque utilisateur dispose d'un pseudo unique, évitant ainsi toute confusion lorsqu'il ajoutera des jeux à sa bibliothèque ou voudra la consulter.

```
Private Sub CmdEnregistrer_Click()
    ' déclaration des variables
    Dim Pseudo As String
    Dim i As Integer
    Dim PlagePseudo As Range

    ' fait appel à une procédure qui compte les lignes
    Call CompteLigne

    'prend la valeur de la textbox
    Pseudo = txtPseudo

    Set PlagePseudo = ThisWorkbook.Worksheets("Pseudo").Range("A2:A" & LignePseudo)

    '***** On vérifie si le pseudo choisi existe déjà ou non *****
    For i = 2 To (LignePseudo - 1)
        If WorksheetFunction.XLookup(Pseudo, PlagePseudo.Value, PlagePseudo.Value, False) = False Then
            Exit For 'sort de la boucle car le pseudo à déjà été ajouté
        Else
            MsgBox "Le pseudo choisi existe déjà." & " Veuillez choisir un autre pseudo."
            txtPseudo = ""
            Exit Sub
        End If
    Next i

    ThisWorkbook.Worksheets("Pseudo").Cells(LignePseudo + 1, 1).Value = Pseudo 'on écrit le pseudo dans la ligne (dernière non vide)
    Unload Me 'ferme le usf
    MsgBox "Merci de votre inscription"
End Sub
```

Une boucle a été implémentée pour vérifier l'existence du nom du jeu, parcourant les lignes de la 2e ligne jusqu'à la dernière ligne comptée non vide (-1). Dans une structure conditionnelle, une vérification est réalisée en utilisant la fonction Excel "RechercheX". Le code suit la même logique que sur Excel lors de l'utilisation de cette fonction. Si la fonction renvoie "false", cela indique qu'aucune valeur correspondante n'a été trouvée, ce qui signifie que le pseudo n'existe pas encore. Dans ce cas, la boucle "for" est interrompue, et la procédure peut se poursuivre.

Cependant, si le pseudo existe déjà, une MessageBox s'affiche pour informer l'utilisateur que le pseudo est déjà pris. Il lui est alors impossible de sélectionner ce pseudo, et lors de la validation de la MessageBox, le champ de texte du pseudo se réinitialise. Cela permet à l'utilisateur de réessayer en choisissant un autre pseudo.

- **Obliger la saisie dans les champs obligatoire**

Nous avons pris la décision d'obliger les utilisateurs à renseigner 4 champs qui seront donc obligatoires. Pour ce faire, une structure conditionnelle a été mise en place afin de vérifier la saisie de ces champs requis.

```
***** Obliger la saisie dans le champs est obligatoire *****
```

```
optSelect = (optSolitaire.Value = False) And (optDuo.Value = False) And (opt3a6.Value = False) And (opt6a10.Value = False)
' si un des champs n'est pas renseigné alors
If txtNomDuJeu.Text = "" Or cboPseudo.Text = "" Or ((optSolitaire.Value = False) And (optDuo.Value = False) And (opt3a6.Value = False) And (opt6a10.Value = False))
    ' Afficher un message d'erreur
    MsgBox "Vous n'avez pas complété tous les champs obligatoires.", vbExclamation, "Erreur de saisie"
    If txtNomDuJeu.Text = "" Then
        ' Mettre en évidence la TextBox
        lblNomDuJeu.ForeColor = RGB(0, 0, 0)
    End If

    If cboPseudo.Text = "" Then
        ' Mettre en évidence la TextBox
        lblPseudo.ForeColor = RGB(0, 0, 0)
    End If

    If optSelect Then
        ' Mettre en évidence la TextBox
        fraNbreDeJoueurs.ForeColor = RGB(0, 0, 0)
    End If

    If lstCategorie.ListIndex = -1 Then
        ' Mettre en évidence la TextBox
        lblCategorie.ForeColor = RGB(0, 0, 0)
    End If

    ' empêcher la poursuite de l'enregistrement
    Exit Sub
End If
```

Dans un premier temps, la structure examine si l'un des champs n'est pas renseigné. Si tel est le cas, elle affiche une MessageBox. De plus, elle vérifie ensuite chaque champ non complété, si tel est le cas, la couleur du label change afin que l'utilisateur puisse visualiser où il n'a pas complété le champ. Ensuite, la procédure se termine. Dans une autre procédure, lorsque l'utilisateur remplit le champ, la couleur revient à sa valeur initiale.

- **Vérifie si le nom du jeu existe déjà**

Nous avons opté pour la mise en place de ces procédures de vérification afin de garantir qu'un seul jeu portant le même nom apparaisse dans le jeu de données. Cette approche a contribué à simplifier notre application et à rationaliser la gestion des erreurs. Ainsi, dans la bibliothèque partagée, où l'on visualise tous les noms de jeux existants dans l'application, il est impossible d'avoir deux occurrences du même nom.

```
***** Vérifier si le nom existe déjà *****

' la variable prend la valeur de la textbox
NomJeuUnique = txtNomDuJeu.Value

'Affecter la feuille "jeu de donnée" à la variable
Set PlageNomJeu = ThisWorkbook.Worksheets("Bibliothèque partagée").Range("A2:A" & LigneBiblioPartage)

' Boucle pour vérifier si le nom existe
For u = 2 To (LigneBiblioPartage - 1)

    If WorksheetFunction.XLookup(NomJeuUnique, PlageNomJeu.Value, PlageNomJeu.Value, False) = False Then
        Exit For 'Le nom n'existe pas, on peut continuer le code

    Else
        If MsgBox("Le nom du jeu choisi existe déjà." & Chr(30) & "Voulez-vous l'ajouter à votre bibliothèque ?", vbYesNo) = vbYes Then
            Unload Me
            usfJeuExistant.Show
            Exit Sub
        Else
            txtNomDuJeu = "" 'effacer nom du jeu
            ' Boucle à travers tous les éléments de la lstSelectionJeu
            For i = 0 To lstCategorie.ListCount - 1 'compte le nombre d'élément dans la lstbox
                lstCategorie.Selected(i) = False ' Désélectionne chaque élément
            Next i
            Exit Sub
        End If
    End If
Next u
```

Ce code suit le même principe que la vérification de l'existence du pseudo.

Une boucle a été implémentée pour vérifier l'existence du nom du jeu, parcourant les lignes de la 2e ligne jusqu'à la dernière ligne comptée non vide (-1). Dans une structure conditionnelle, une vérification est réalisée en utilisant la fonction Excel "RechercheX". Le code suit la même logique que sur Excel lors de l'utilisation de cette fonction. Si la fonction renvoie "false", cela indique qu'aucune valeur correspondante n'a été trouvée, ce qui signifie que le nom du jeu n'existe pas encore. Dans ce cas, la boucle "for" est interrompue, et la procédure peut se poursuivre.

En revanche, si le nom du jeu existe déjà, l'utilisateur a la possibilité de l'ajouter à sa propre bibliothèque. Une MessageBox s'affiche pour lui laisser le choix. S'il répond par l'affirmative, cela ouvre le UserForm prévu à cet effet, et la procédure se termine. Si l'utilisateur ne souhaite pas ajouter le jeu à sa bibliothèque, les champs remplis sont remis à zéro, lui permettant ainsi de saisir les détails d'un autre jeu non encore existant.

- **Copier le jeu de donnée dans 2 feuille différente**

Ce code a pour objectif de copier la feuille « Jeu de données » vers deux feuilles distinctes, à savoir « Ma Bibliothèque » et « Bibliothèque Partagée ».

Nous avons opté pour cette approche pour deux raisons :

- **Copie vers la Bibliothèque Partagée** : Le choix de ne pas inclure le pseudo lors de la copie vers la bibliothèque partagée vise à garantir l'anonymat des utilisateurs, conformément à l'objectif de l'application. L'idée est que les utilisateurs puissent uniquement visualiser les jeux sans savoir qui les a ajoutés. De plus, cette méthode évite d'avoir à spécifier à chaque ligne de code que, en plus de l'enregistrement dans la feuille « Jeu de données », il faut également l'enregistrer dans la feuille « Bibliothèque Partagée ».
  - **Copie vers la Bibliothèque Personnelle** : La copie du jeu de données vers la bibliothèque personnelle s'explique simplement par le fait que la feuille « Bibliothèque personnelle » consolidera l'ensemble des données, offrant une vue complète du jeu de données, y compris les jeux déjà possédés par les utilisateurs. La justification du code dans le « usfJeuExistant » complétera cette approche.

```

" Copier le jeu de donnée directement dans biblio partagée et ma biblio "
" Déclaration des variables "
Dim PlageSource As Range
Dim PlageSourceMaBiblio As Range

Dim PlageDestination As Range
Dim PlageMaBiblio As Range

" Définir quelles valeurs les variables prennent "
' on choisi à quoi correspond les plages

Set PlageSource = wsJeuDeDonne.Range("B" & LigneJeuDonne + 1 & ":F" & LigneJeuDonne + 1)
Set PlageDestination = wsBiblioPartage.Range(wsBiblioPartage.Cells(LigneBiblioPartage + 1, 1), wsBiblioPartage.Cells(LigneBiblioPartage + 1, 5))

" Copier les valeurs dans bibliopartagée "
PlageSource.Copy Destination:=PlageDestination

" copier les valeur dans biblioperso "
Set PlageSourceMaBiblio = wsJeuDeDonne.Range("A" & LigneJeuDonne + 1 & ":F" & LigneJeuDonne + 1)
Set PlageMaBiblio = wsMaBiblio.Range(wsMaBiblio.Cells(LigneMaBiblio + 1, 1), wsMaBiblio.Cells(LigneMaBiblio + 1, 6))

PlageSourceMaBiblio.Copy Destination:=PlageMaBiblio

'Ferme le usf
Unload Me
MsgBox "J'ajout de jeu a bien été enregistré"

```

L'emploi de la variable "LigneJeuDonne + 1" s'explique par le fait que cette opération est déclenchée lorsque l'utilisateur clique sur "Enregistrer". À ce stade, le décompte des lignes n'a pas encore été mis à jour pour refléter l'ajout récent du jeu, d'où le +1. Ainsi, ce code copie exclusivement la ligne récemment ajoutée au moment où l'utilisateur appuie sur "Enregistrer".

### 6.2.3 usfJeuExistant

---

Le « usfJeuExistant » permet d'enregistrer des jeux de société déjà présents dans l'application, mais dans la propre bibliothèque de l'utilisateur. Ces enregistrements sont effectués dans la feuille « Ma Bibliothèque ». À noter que cette feuille regroupera non seulement les jeux de société ajoutés à partir du « usfAjouterUnJeu », mais également l'ensemble des jeux possédés par chaque utilisateur. En d'autres termes, « Ma Bibliothèque » constitue une compilation complète du jeu de données, incluant tous les jeux détenus par chaque personne inscrite dans l'application.

```
' fait appel à une procédure qui compte les lignes
Call CompteLigne

For Each NomSelectionne In usfJeuExistant.lstJeuChoisi.List

    For i = 1 To LigneBiblioPartage

        If wsBiblioPartage.Cells(i, 1).Value = NomSelectionne Then

            ' enregistre la sélection du pseudo dans la variable pseudo
            Pseudo = cboPseudo.Value

            ' enregistre le pseudo à la dernière ligne dans la colonne 1
            wsMaBiblio.Cells(LigneMaBiblio + 1, 1).Value = Pseudo

            ' copie les données de la colonne A jusqu'à E de la ligne correspondante aux jeux sélectionnés
            For j = 1 To 5 ' Copie jusqu'à la 5è colonne

                wsMaBiblio.Cells(LigneMaBiblio + 1, j + 1).Value = wsBiblioPartage.Cells(i, j).Value
                ' mettre les bordures à chaque nouvelles lignes
                wsMaBiblio.Cells(LigneMaBiblio + 1, j + 1).Borders.Weight = xlThin

                If wsMaBiblio.Cells(LigneMaBiblio + 1, j + 1).Value = "" Then
                    wsMaBiblio.Cells(LigneMaBiblio + 1, j + 1).Interior.Color = RGB(220, 220, 220)

                End If

            Next j

            LigneMaBiblio = LigneMaBiblio + 1 ' passe à la prochaine ligne pour coller

        End If

    Next i

Next NomSelectionne
```

L'objectif global de cette procédure est de transférer des données de la feuille « Bibliothèque Partagée » vers la feuille « Ma bibliothèque », en fonction des jeux sélectionnés dans la liste déroulante (lstJeuChoisi), en ajoutant également le pseudo spécifié dans ladite liste déroulante. Autrement dit, l'objectif est de repérer le nom du jeu sélectionné dans la « Bibliothèque Partagée ». Une fois trouvé, la procédure vise à copier l'intégralité de la ligne correspondante et à la coller dans la feuille « Ma bibliothèque », avec l'ajout du pseudo en première colonne.

En ce qui concerne plus précisément le code, la variable NomSelectionne stocke chaque élément de la lstJeuChoisi. La boucle « For Each » permet de parcourir chaque élément de lstJeuChoisi. À l'intérieur de cette boucle, une autre boucle « For » est imbriquée, elle permet de parcourir chaque ligne de la feuille « Bibliothèque Partagée » jusqu'à la dernière ligne non vide. Une structure conditionnelle est également imbriquée, vérifiant si la valeur de la cellule de la colonne 1 à la ligne i correspond à l'élément sélectionné dans la listebox. Si une correspondance est trouvée, l'objectif principal de la procédure est déclenché.

## 6.2.4 usfConnexion

Ce code permet à l'utilisateur d'accéder à l'ensemble de ses jeux personnels. La recherche de ses propres jeux est réalisée en utilisant le pseudo qui se trouve dans la première colonne, bien que cette information soit masquée.

```
Private Sub cmdAccederBiblio_Click()
    Dim FiltrePseudo As String 'la valeur de la cbo qui choisi le filtre
    Dim i As Integer

    ' fait appel à une procédure qui compte les lignes
    Call CompteLigne

    '***** Filtrer en fonction du pseudo sélectionné *****
    FiltrePseudo = cboPseudo.Value

    For i = 2 To LigneMaBiblio
        'réinitialise le masquage de ligne
        ThisWorkbook.Worksheets("Ma Bibliothèque").Rows(i).Hidden = False

        ' vérifie si la valeur dans la cellule correspond à la valeur à la combobox complété
        If Not (ThisWorkbook.Worksheets("Ma Bibliothèque").Cells(i, 1).Value = FiltrePseudo) Then
            ' si la valeur ne correspond pas, masque la ligne entière
            ThisWorkbook.Worksheets("Ma Bibliothèque").Rows(i).Hidden = True
        Else
            ThisWorkbook.Worksheets("Ma Bibliothèque").Rows(i).Hidden = False
        End If
    Next i

```

Ce code utilise une boucle qui parcourt de la ligne 2 jusqu'à la dernière ligne non vide afin de masquer les lignes qui ne comportent pas le pseudo. La colonne où les pseudos sont inscrits est réduite au maximum pour que l'utilisateur ne la voie pas. À l'intérieur de cette boucle, une structure conditionnelle est présente pour vérifier si la valeur dans la ComboBox est égale à la valeur de la cellule i. Si les deux valeurs ne sont pas égales, la ligne est masquée. Sinon, elle reste visible.

# 7.

## LES AXES D'AMELIORATION

Pour conclure ce rapport, nous allons présenter les améliorations envisageables concernant le fonctionnement et le code de notre application.

Bien que nous soyons satisfaites du travail accompli et de l'application que nous présentons, nous sommes conscientes qu'elle peut encore être perfectionnée. C'est pourquoi nous nous sommes penchées sur des modifications susceptibles d'améliorer son efficacité en optimisant notre code.

En effet, certaines améliorations auraient pu être apportées pour faciliter l'expérience de l'utilisateur de notre application. Tout d'abord, le tri des combobox, comme celui du pseudo, par ordre alphabétique aurait permis de retrouver plus facilement son pseudo. Une autre solution aurait été de laisser l'utilisateur écrire son pseudo avec une liste déroulante en dessous permettant une recherche intelligente, simplifiant ainsi la recherche parmi une grande liste de pseudos. Le tri alphabétique aurait également pu être appliqué à la listebox des jeux à ajouter à sa bibliothèque (dans le "usfJeuExistant"), facilitant ainsi la recherche des jeux. De plus, lorsqu'un utilisateur ajoute un jeu à sa bibliothèque, il se peut qu'il ajoute un titre qu'il possède déjà. Il aurait été nécessaire de trouver une solution à ce problème.

En second lieu, toujours dans ce même UserForm, il aurait été utile que lorsque celui-ci s'ouvre après que l'utilisateur a rempli les champs et que le nom existe déjà, les champs tels que le pseudo et le nom de jeu soient sélectionnés automatiquement dans ce UserForm.

Enfin, un détail pratique aurait été d'offrir à l'utilisateur la possibilité de désélectionner des boutons d'option dans les champs facultatifs du « usfAjouterUnJeu », au cas où il change d'avis et ne souhaite plus renseigner ces champs. De plus, il aurait été judicieux d'empêcher les utilisateurs d'écrire dans toutes les cellules afin de prévenir toute modification non autorisée de nos données collectées.

En ce qui concerne le fonctionnement de l'application, une possibilité que nous aurions pu envisager était de permettre aux utilisateurs de laisser des commentaires, des notes ou des avis sur les jeux de société. Cela aurait contribué à affiner de manière plus précise la recherche de jeux. Bien que nous soyons restées fidèles aux fondamentaux de notre application, l'ajout de cette fonctionnalité aurait pu s'avérer intéressant à implémenter.

# 8.

## CONCLUSION

Ce projet s'est avéré extrêmement enrichissant à plusieurs égards. Tout d'abord, nous avons saisi l'opportunité de mettre en pratique nos connaissances sur Excel acquises au cours de ces années à l'université. En plus d'utiliser les outils informatiques étudiés en cours de VBA cette année, nous avons également réutilisé des compétences acquises lors de nos première et deuxième années, telles que la validation des données, le gel des volets, l'utilisation de la recherche X, etc. Cependant, c'est dans la programmation que nous avons le plus progressé, car cet exercice était nouveau pour toutes les deux. Pour nous guider dans la rédaction de notre code, nous avons utilisé les connaissances acquises lors de nos cours, ainsi que sollicité l'assistance de ChatGPT.

Deuxièmement, comme de nombreux travaux cette année, ce projet nous a offert l'opportunité de coopérer et de réfléchir collectivement autour d'un objectif commun. Nous sommes conscientes que cette compétence sera primordiale dans nos expériences futures, en particulier sur le plan professionnel.

Enfin, travailler sur le code et les algorithmes a été une expérience très enrichissante. Il a fallu faire preuve de rigueur et être attentives à chaque étape de notre travail. Nous nous sommes rapidement prises au jeu de la programmation, trouvant une grande satisfaction dans la réussite d'une manipulation et dans le bon fonctionnement du code. L'idée que l'on peut constamment améliorer notre code est stimulante, bien que parfois frustrante lorsque l'on doit s'arrêter et ne pas continuer. La découverte de VBA a été particulièrement positive pour nous.

This project has been highly rewarding on several fronts. Firstly, we seized the opportunity to apply the Excel skills we've acquired over the years at university. In addition to using the computer tools covered in this year's VBA course, we revisited our knowledge from the first and second years, such as data validation, freezing panes, utilizing the XLOOKUP function, and more. However, it is in coding that we have seen the most improvement, as this exercise was a novel experience for both of us. To assist us in our code, we relied on the lessons we had received, and we also sought help from ChatGPT.

Secondly, like many assignments this year, this project provided an opportunity to collaborate and think collectively toward a common goal. We are aware that this skill will be crucial in our future experiences, especially in a professional context.

Finally, working on code and algorithms has been very enriching. It required a high level of precision and attention to detail in every aspect of our work. We quickly became immersed in the world of coding, finding great satisfaction in successfully executing a task and witnessing the code function properly. The idea that we can continually enhance our code is motivating, although it can be frustrating when we have to pause and not proceed. The discovery of VBA has been a particularly positive experience for us.