

Questions triviales

Rapport de projet de Programmation Orientée Objet en Java

Professeur

Michel LANDSCHOOT

Membres

Mathilde PAYSANT

Emmie KIEFFER

Antoine AOUAT

Thomas GENDRON

Groupe E

2021/2022



Sommaire

Sommaire	2
Introduction	3
Présentation fonctionnelle	3
Fonctionnalités essentielles	3
Fonctionnalités bonus	4
Présentation technique	4
Package Joueur	4
Package Question	4
Package Theme	5
Déroulement de la partie	5
Phase 1	5
Phase 2	6
Phase 3	6
Présentation des résultats	7
Package Joueur	7
Package Question	7
Package Theme	8
Main	9
Conclusion	11

Introduction

Pour mettre en commun toutes les connaissances apprises lors de nos cours de Programmation Orientée Objet, nous avons été amenés à développer un jeu de Questions/Réponses avec le langage Java et en utilisant le mode console.

Les connaissances que nous avons dû réunir dans ce projet sont les concepts généraux de Java, les packages, les interfaces, les collections, les inputs/outputs, mais également des notions pas vues en cours, qui nous ont été utiles et que nous avons dû apprendre à utiliser par nous-même.

Le but du jeu est simple : avoir le plus de bonnes réponses pour gagner et ne pas être éliminé. Durant le jeu, il y a 3 phases où une personne est éliminée à chaque fin de phase. Pour avoir un seul gagnant à la fin, il faut ainsi que 4 joueurs soient sélectionnés au début pour commencer la partie. La personne qui est éliminée correspond à la personne qui a récolté le moins de points. Chaque réponse juste permet de récolter des points. Une réponse fausse n'enlève aucun point. Concernant les questions, il y a 10 thèmes différents, les questions peuvent être de trois types (Question Choix Multiple, Vrai/Faux, Réponse Courte) et de trois niveaux (Facile, Moyen et Difficile).

Présentation fonctionnelle

L'objectif du projet est de permettre à 4 joueurs de lancer une partie d'un jeu qui les mettra en concurrence à travers des questions de plusieurs types sur des thèmes variés dont ils saisiront les réponses. Le gagnant sera le dernier joueur restant à l'issue des différentes phases, que l'on décrira dans les paragraphes qui suivront, et qui permettront de rendre le jeu moins monotone.

Fonctionnalités essentielles

Le jeu peut se dérouler entièrement, à commencer par le choix du nom des joueurs ainsi que la possibilité d'en supprimer certains avant de lancer la partie.

La partie débute alors par la première phase.

Comme nous l'avons vu, les questions peuvent attendre trois types de réponses différentes. Les questions Vrai / Faux admettent ainsi une seule réponse correcte sur deux proposés ("Vrai" et "Faux"), tout comme les questions QCM (Question Choix Multiple) qui proposent, elles, au moins 3 propositions de réponses. Concernant les questions à Réponse Courte, elles permettent au joueur dont c'est le tour de rentrer sa réponse au clavier, qui sera généralement un mot ou groupe de mot, mais nous avons admis une certaine flexibilité afin que la casse ne soit pas prise en compte. Le code inclut, dans chaque cas, un affichage pour indiquer si la réponse saisie par le joueur est correcte ou non.

Les points sont incrémentés après chaque bonne réponse et un récapitulatif du score de chaque joueur est affiché à la fin de chaque phase. Le joueur au plus faible total de point est éliminé puis la phase suivante débute avec les joueurs encore en lice.

Dans la phase qui suit, la difficulté des questions augmente d'un niveau parmi : Facile, Moyen et Difficile atteint dans la troisième phase du jeu qui oppose les deux derniers joueurs.

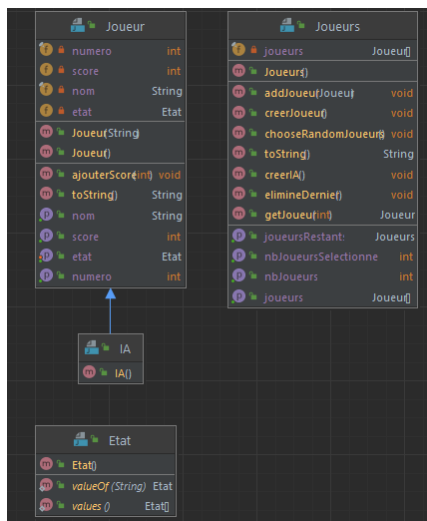
Fonctionnalités bonus

Le jeu étant fonctionnel, nous avons ajouté la possibilité, avant de lancer la partie, d'ajouter une ou plusieurs IA (Intelligence Artificielle) qui remplacent un joueur réel et répondent automatiquement et aléatoirement à chaque réponse.

Présentation technique

Le Java étant un langage de programmation objet, nous avons développé notre projet en conséquence. Chaque classe représente un objet du jeu. Les classes sont classées dans des packages pour rendre la navigation dans les fichiers plus simple et plus lisible.

Package Joueur



Le package joueur contient 5 classes dont la classe **Joueur** qui est la plus importante.

Cette classe représente un joueur enregistré dans le jeu. Un joueur possède un numéro incrémenté automatiquement, un nom choisi par l'utilisateur au début de la partie, un score débutant à 0 puis augmentant en fonction de ses réponses.

Un joueur est aussi représenté par un état codé comme un enum **Etat**. Comme précisé dans le sujet, 5 états sont possibles : gagnant, selectionne, superGagnant, elimine et enAttente même si superGagnant n'est pas utilisé car utile seulement dans la partie bonus des tournois que nous n'avons pas codé. L'état du joueur change en fonction de l'avancement de la partie.

Un joueur peut aussi être une **IA**. La classe IA hérite de la classe Joueur et l'IA est initialisée avec un nom aléatoire choisi dans une liste. Une IA se comporte comme un joueur. Lorsqu'un joueur est détecté comme étant une IA, une fonction en fonction du type de la question est lancée pour y répondre au hasard.

Enfin, tous les joueurs sont stockés dans l'objet **Joueurs** qui est un tableau de joueurs. Cet objet peut contenir la liste complète des joueurs ou seulement la liste des joueurs encore en jeu.

Enfin, la classe **TestJoueur** permet de tester les fonctions du package et la création des objets.

Package Question

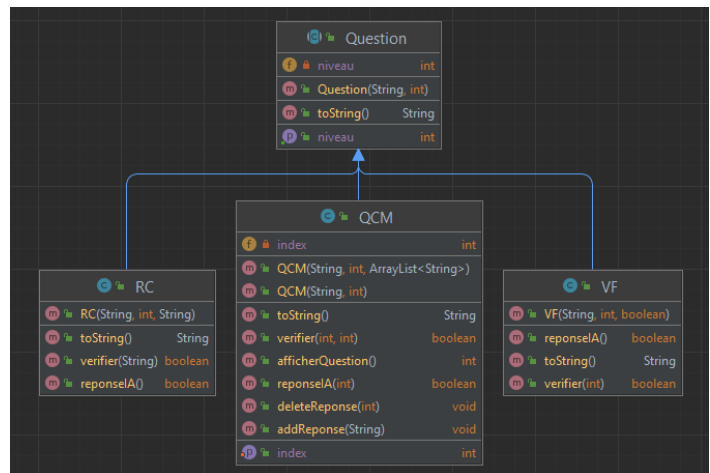
Ce package regroupe l'ensemble des classes permettant de créer les questions du jeu. Toutes les classes (à part la classe `TestQuestion`) héritent de la classe abstraite **Question**. Toutes les questions sont composées d'un énoncé et d'un niveau.

Une question **QCM** possède en plus une liste de réponses et l'index de la bonne réponse. On peut ajouter des réponses et les supprimer à partir de leur index.

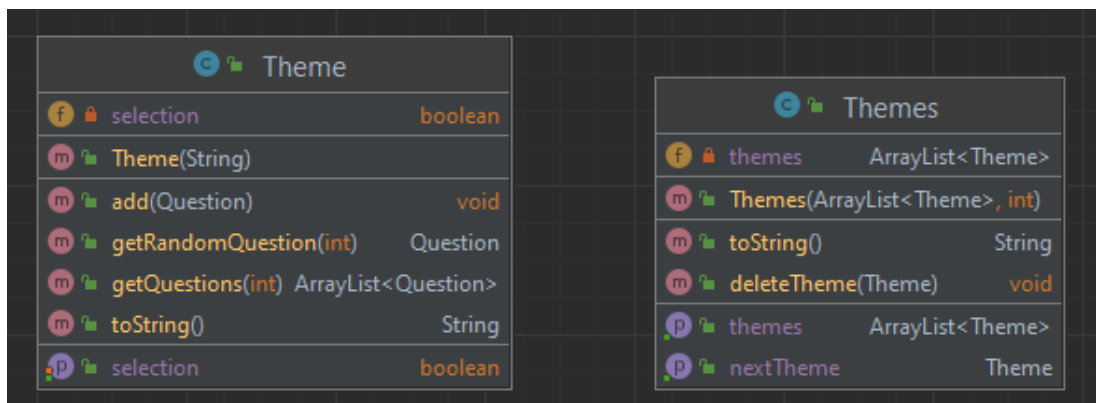
La fonction `afficherQuestion` est différente de `toString` car les questions doivent être mélangées. Chaque classe génère une réponse pour les IA, aléatoire.

La classe **RC** permet la création de questions à réponse courte. La bonne réponse est une chaîne de caractères. La vérification de la bonne réponse par rapport à une chaîne de caractère donnée par l'utilisateur se fait sans prendre la casse en compte.

La dernière classe permet la création de questions **VF** avec comme attribut la bonne réponse sous forme de booléen.



Package Theme



Le package Theme réunit les classes gérant les groupes de questions.

La classe **Theme** est un groupe de questions avec un nom. Ces questions sont de tous types. Il possède aussi un booléen représentant si le thème est sélectionné et en cours d'utilisation dans le jeu ou non.

La classe **Themes** est un regroupement de thèmes. La création de l'objet est réalisée en fonction de la phase de jeu dans laquelle nous sommes. L'objet Themes de la phase 1 contient l'ensemble des thèmes, celui de la phase 2 comprend 6 thèmes pris au hasard et l'objet de la phase 3 en sélectionne 3 au hasard.

Nous utilisons la fonction `getNextTheme` pour avoir les thèmes dans l'ordre. Et changer de thème à la fin de son utilisation.

Déroulement de la partie

Notre fonction main est découpée en 3 grandes parties, représentant respectivement les 3 phases de notre jeu, chacune ayant des caractéristiques spécifiques qui nous imposent des fonctions et des boucles différentes.

Phase 1

Pendant la première phase, 4 joueurs maximum sont présents et tous les thèmes sont utilisés. Il y a donc 10 thèmes qui vont être posés. Les jours présents sont choisis aléatoirement parmi une liste de joueurs en attente.

Le premier thème choisi correspond au premier thème disponible dans la liste de Thèmes qui est créé à l'initialisation du jeu. On pose ensuite une question à chaque joueur avec ce thème.

Ensuite, nous prenons le thème suivant dans la liste de Thèmes et on pose une question de difficulté facile à chaque joueur avec ce thème. On fait cette boucle jusqu'à ce qu'on ait posé des questions à chacun sur tous les thèmes disponibles. On fait alors cette boucle dix fois, car il y a 10 thèmes.

Chaque bonne réponse à une question de cette phase permet au joueur à qui on a posé la question de récolter 2 points supplémentaires sur son score. Le joueur avec le moins de points à la fin de la phase est éliminé de la partie.

Phase 2

La deuxième phase du jeu concerne les trois joueurs sélectionnés à l'issue de la première.

Une liste de 6 thèmes aléatoires va être initialisée, puis, la liste sera proposée aux joueurs et chacun devra choisir un thème tour à tour. Si le joueur est une IA, le premier thème de la liste sera choisi à chaque fois. Nous avons décidé que le premier joueur à choisir un thème serait tiré aléatoirement par le jeu.

Deux thèmes seront donc attribués à chaque joueur dans un tableau, et les questions qui leur seront posées ne porteront que sur les thèmes choisis. Pour cela, on parcourt le tableau pour le premier thème de chaque joueur, puis le deuxième et l'on pose une question de difficulté moyenne à chaque fois.

Une bonne réponse octroie 3 points. A nouveau, le joueur au total de points le plus faible est éliminé, les deux autres sont sélectionnés pour la phase suivante.

Phase 3

La phase 3 est la dernière phase du jeu, elle oppose les deux joueurs restants de la phase 2.

Durant cette phase, 3 thèmes sont choisis par le concepteur. Pour réaliser cette dernière expression, nous avons décidé que les 3 thèmes seront sélectionnés aléatoirement parmi les 10 disponibles. Le jeu va donc poser aux joueurs (à tour de rôle), 2 questions de

niveau difficile sur chacun des 3 thèmes sélectionnés. Chaque bonne réponse octroie 5 points lors de cette phase.

Nous avons utilisé à nouveau la boucle for ainsi que fonction associant et posant les questions aux joueurs que nous avons utilisé lors de la phase 1, à ceci près que nous réalisons cette boucle trois fois, puisque nous avons trois thèmes lors de cette phase.

Une fois tous les thèmes traités, nous éliminons le joueur ayant le score le moins élevé. Le dernier joueur présent est déclaré vainqueur !

Présentation des résultats

Les classes Test permettent de tester toutes les fonctions des classes de manière unitaire. Elles peuvent être lancées dans leur package respectif.

Package Joueur

```
TEST Joueur : Joueur (joueur 100) : 0 points [en attente]
TEST setEtat : Joueur (joueur 100) : 0 points [éliminé]
TEST ajouterScore : Joueur (joueur 100) : 30 points [éliminé]
TEST IAOptimus Prime (joueur 110) : 0 points [en attente]
```

Test des fonctions de la classe Joueur dans TestJoueur

```
TEST Joueurs : liste vide attendue

TEST addJoueur :
Joueur (joueur 100) : 30 points [éliminé]

Entrez le nom du joueur :
Metropolis
Le joueur Mathilde a été ajouté.

TEST creerJoueur :
Joueur (joueur 100) : 30 points [éliminé]
Mathilde (joueur 120) : 0 points [en attente]

Le joueur Terminator a été ajouté.

TEST creerIA :
Joueur (joueur 100) : 30 points [éliminé]
Mathilde (joueur 120) : 0 points [en attente]
Terminator (joueur 130) : 0 points [en attente]
```

```
Entrez le numéro du joueur à supprimer :
1
Le joueur Mathilde a été supprimé.

TEST deleteJoueur :
Joueur (joueur 100) : 30 points [éliminé]
Terminator (joueur 130) : 0 points [en attente]

Le joueur Robocop a été ajouté.
Le joueur HAL-9000 a été ajouté.
Le joueur Deception a été ajouté.
Le joueur Terminator a été ajouté.

TEST choisirDesJoueurs :
Joueur (joueur 100) : 30 points [sélectionné]
Terminator (joueur 130) : 0 points [en attente]
Robocop (joueur 140) : 0 points [sélectionné]
HAL-9000 (joueur 150) : 0 points [sélectionné]
Deception (joueur 160) : 0 points [sélectionné]
Terminator (joueur 170) : 0 points [en attente]

TEST getJoueursRestants :
Joueur (joueur 100) : 30 points [sélectionné]
Robocop (joueur 140) : 0 points [sélectionné]
HAL-9000 (joueur 150) : 0 points [sélectionné]
```

```
TEST getJoueurFinal : erreur attendue
Erreur : plusieurs joueurs détectés
```

```
TEST getJoueurFinal : erreur attendue
Erreur : aucun joueur détecté
```

```
TEST getJoueurFinal : aucune erreur attendue
Metropolis (joueur 110) : 0 points [sélectionné]
Metropolis a été éliminé.
TEST elimineDernier :
Joueur (joueur 100) : 60 points [sélectionné]
Metropolis (joueur 110) : 0 points [éliminé]
```

Test des fonctions de la classe Joueurs dans TestJoueur

Package Question

```
TEST QCM :
Enoncé

TEST addReponse :
Enoncé
1) Réponse 1
2) Réponse 2
3) Réponse 3
4) Réponse 4

TEST deleteReponse :
Enoncé
1) Réponse 2
2) Réponse 3
3) Réponse 4

TEST setIndex :
Enoncé
1) Réponse 2
2) Réponse 3
3) Réponse 4

TEST verifier : mauvaise réponse
false

TEST verifier : bonne réponse
true
```

```
TEST afficherQuestion :
Enoncé
1) Réponse 3
2) Réponse 2
3) Réponse 4

TEST reponseIA :
2
true

TEST RC :
Enoncé
Répondez en tapant votre réponse.

TEST verifier : "Réponse"
true

TEST verifier : "réponse"
true

TEST verifier : "ezgzrg"
false

TEST reponseIA :
Je ne sais pas.
```

Test des fonctions de la classe QCM dans TestQuestion

```
TEST VF :
Enoncé
1) Vrai
2) Faux

TEST verifier : mauvaise réponse
false

TEST verifier : bonne réponse
true

TEST reponseIA :
2
false
```

```
TEST RC :
Enoncé
Répondez en tapant votre réponse.

TEST verifier : "Réponse"
true

TEST verifier : "réponse"
true

TEST verifier : "ezgzrg"
false

TEST reponseIA :
Je ne sais pas.
```

Test des fonctions de la classe VF

Test des fonctions de la classe RC

Package Theme

```
TEST Thème :
Thème 1

TEST add + getQuestions :
[Enoncé
1) Vrai
2) Faux, Enoncé 2
1) Vrai
2) Faux]

TEST getRandomQuestion :
Enoncé 2
1) Vrai
2) Faux
```

Test des fonctions de la classeTheme

```
TEST Thèmes : phase 1
Thème 1
Thème 2
Thème 3
Thème 4
Thème 5
Thème 6
Thème 7
Thème 8
Thème 9
Thème 10

TEST Thèmes : phase 2
Thème 2
Thème 9
Thème 5
Thème 6
Thème 3
Thème 4

TEST Thèmes : phase 3
Thème 3
Thème 9
Thème 1
```

Test des fonctions de la classe Themes

Main

```
QUESTIONS TRIVIALES

Bienvenue sur "Questions Triviales".
Pour commencer une partie, ajoutez au moins 4 joueurs.

Que souhaitez vous faire ?
1) Ajouter un joueur
2) Supprimer un joueur
3) Afficher les joueurs
4) Ajouter une IA
5) Lancer la partie
1

Entrez le nom du joueur :
Mathilde
Le joueur Mathilde a été ajouté.

Que souhaitez vous faire ?
1) Ajouter un joueur
2) Supprimer un joueur
3) Afficher les joueurs
4) Ajouter une IA
5) Lancer la partie
1

Mathilde (joueur 100) : 0 points [en attente]

Que souhaitez vous faire ?
1) Ajouter un joueur
2) Supprimer un joueur
3) Afficher les joueurs
4) Ajouter une IA
5) Lancer la partie
1

Le joueur WALL-E a été ajouté.
```

Sélection des joueurs

```
La phase 1 est terminée ! Voyons la liste des joueurs :
Mathilde (joueur 100) : 2 points [sélectionné]
WALL-E (joueur 110) : 8 points [sélectionné]
HAL-9000 (joueur 120) : 12 points [sélectionné]
C-3PO (joueur 130) : 0 points [sélectionné]

Le joueur avec le score le plus bas va être éliminé.
Mathilde a été éliminé.
===== PHASE 2 =====
Chaque joueur va être interrogé sur les thèmes qui aura choisi dans la liste.

HAL-9000, choisissez un thème parmi :
1) Mathématiques
2) Histoire
3) Nature
4) Arts
5) Sports
6) Divertissement
4

C-3PO, choisissez un thème parmi :
1) Mathématiques
2) Histoire
3) Nature
4) Sports
5) Divertissement
1

WALL-E, choisissez un thème parmi :
1) Histoire
2) Nature
3) Sports
4) Divertissement
1
```

Fin de la partie 1 et partie 2

```
Lancement de la partie...

Le jeu se déroule en 3 phases différentes.
Après chaque phase, un joueur est éliminé jusqu'à ce qu'il n'en reste qu'un.
Bonne chance !

===== PHASE 1 =====
Chaque joueur va être interrogé sur une question de chaque thème.

Liste des thèmes :
Histoire
Géographie
Nature
Littérature
Cinéma
Science
Mathématiques
Sports
Arts
Divertissement

===== Thème : Sports =====

Mathilde, à vous de jouer !
Messi porte le maillot 30 au PSG ?
1) Vrai
2) Faux

Mauvaise réponse ! Mathilde ne gagne aucun point.

WALL-E, à vous de jouer !
Quelle est la durée en minutes d'un match de football ?
Répondez en tapant votre réponse.
90
Bonne réponse ! WALL-E gagne 2 points.

HAL-9000, à vous de jouer !
Que signifie GRS ?
1) Gymnastique Rythmique et Sportive
```

Déroulement de la phase 1

```
Le joueur avec le score le plus bas va être éliminé.
C-3PO a été éliminé.
===== PHASE 3 =====
Les deux derniers joueurs vont s'affronter sur 3 thèmes choisis aléatoirement.

===== Thème : Sports =====

WALL-E, à vous de jouer !
Combien de titres de moto Valentino Rossi a-t-il remporté ?
Répondez en tapant votre réponse.
9
Bonne réponse ! WALL-E gagne 5 points.

HAL-9000, à vous de jouer !
Quelle est la périodicité des jeux Olympiques d'été ?
1) Tous les quatre ans
2) Tous les deux ans
3) Tous les trois ans
1
Bonne réponse ! HAL-9000 gagne 5 points.

===== Thème : Arts =====

WALL-E, à vous de jouer !
The American Gothic est l'une des peintures les plus reconnaissables au monde. Qui l'a peint ?
Répondez en tapant votre réponse.
Je ne sais pas.
Mauvaise réponse ! WALL-E ne gagne aucun point.

HAL-9000, à vous de jouer !
Qui a été la première personne vivante à avoir vu ses œuvres exposées au Louvre ?
Répondez en tapant votre réponse.
Je ne sais pas.
Mauvaise réponse ! HAL-9000 ne gagne aucun point.

===== Thème : Mathématiques =====

WALL-E, à vous de jouer !
Quel est le carré du quart du tiers de 12 ?
```

Fin de la partie 2 et partie 3

```
HAL-9000, à vous de jouer !
9-9/9+9-9/9=
Répondez en tapant votre réponse.
Je ne sais pas.
Mauvaise réponse ! HAL-9000 ne gagne aucun point.

La phase 3 est terminée ! Voyons la liste des joueurs :
Mathilde (joueur 100) : 2 points [éliminé]
WALL-E (joueur 110) : 16 points [sélectionné]
HAL-9000 (joueur 120) : 20 points [sélectionné]
C-3PO (joueur 130) : 6 points [éliminé]

Le joueur avec le score le plus bas va être éliminé.
WALL-E a été éliminé.

HAL-9000 gagne la partie !!!

Scores finaux :
Mathilde (joueur 100) : 2 points [éliminé]
WALL-E (joueur 110) : 16 points [éliminé]
HAL-9000 (joueur 120) : 20 points [grand gagnant]
C-3PO (joueur 130) : 6 points [éliminé]
```

Fin de la partie 3 et de la partie

Conclusion

Venant tous de DUT ou de bachelor, nous avons déjà plus ou moins touché à la programmation mais pour deux d'entre nous, Thomas et Antoine, la programmation en Java était nouvelle. Ce projet nous a permis d'appliquer tout ce que nous avons appris en cours durant ce premier semestre. Nous n'avons pas rencontré de problèmes majeurs liés à la pratique car nous étions 2 à avoir déjà réalisé des projets semblables dans le même langage.

Sur le plan de l'organisation du travail, nous avons trouvé un bon équilibre en nous répartissant les tâches en fonction de notre temps et en offrant des challenges aux débutants de Java qui leur permettraient de bien comprendre le fonctionnement de Java. Chacun a travaillé avec son niveau et à sa rapidité pour que le ratio production/temps soit égal pour chaque membre du projet.

Nous avons commencé à travailler sur le projet dès son annonce pour que nous n'ayons pas de souci à la fin du projet. Nous avons même réussi à réaliser l'IA.

Nom	Pourcentage de travail estimé	Description
Mathilde PAYSANT	40 %	<ul style="list-style-type: none">- Création des classes- Sélection et création des joueurs en début de partie- Développement et intégration de l'IA dans le programme- Question des thèmes Histoire, Divertissement & Nature
Emmie KIEFFER	30 %	<ul style="list-style-type: none">- Liaison des questions avec les joueurs en fonction des phases et des thèmes- Poser les questions aux joueurs- Questions des thèmes Maths & Sports
Thomas GENDRON	15 %	<ul style="list-style-type: none">- Choix et attribution des thèmes aux joueurs- Questions des thèmes Géographie & Science
Antoine AOUAT	15 %	<ul style="list-style-type: none">- Main : Regroupement des fonctions selon les phases de jeu- Questions des thèmes Cinéma & Littérature