

# Projet Jeu de go - Programmation orienté objet

13 Mars 2018

Récemment, Google DeepMind a créé la première intelligence artificielle pour le jeu de go, capable de battre un joueur professionnel en partie équilibrée. Pour fêter cette avancée remarquable, vous allez faire un projet de jeu de go. Rassurez-vous, on ne vous demandera de faire une IA.

## 1 Prérequis - Le jeu de Go

### 1.1 Introduction

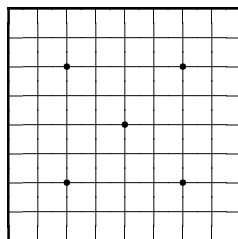
*Le go est un jeu de plateau originaire de Chine. Il oppose deux adversaires qui placent à tour de rôle des **pierres**, respectivement noires et blanches, sur les intersections d'un plateau de jeu quadrillé appelé **goban**. Le but est de contrôler le plan de jeu en y construisant des **territoires**. Les pierres encerclées deviennent des **prisonniers**, le gagnant étant le joueur ayant totalisé le plus de territoires et de prisonniers.*

*Il s'agit du plus ancien jeu de stratégie combinatoire abstrait connu. Malgré son ancienneté, le go continue à jouir d'une grande popularité en Chine, en Corée et au Japon. Dans le reste du monde, où sa découverte est récente, sa notoriété est croissante. Son succès tient autant à la simplicité de ses règles qu'à sa grande richesse combinatoire et sa profondeur stratégique.*

Source : Wikipedia

### 1.2 Matériel

- Le **goban**: il s'agit du plateau de jeu:



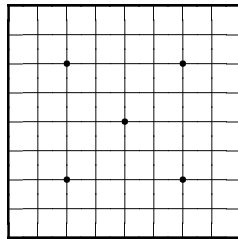
- Les **hoshi** sont les points noirs présents sur le goban.
- Les **pierres**: il s'agit des "pions" en quelque sortes. Un des deux joueurs utilise les noirs, l'autre utilise les blancs.

### 1.3 Les règles du jeu

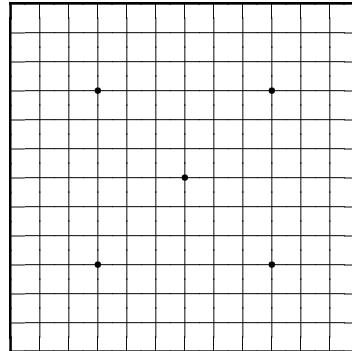
Il y a plusieurs variantes minimales des règles du jeu de go. La plupart des joueurs européens ont appris le jeu avec la règle Japonaise, qui est la plus naturelle pour les humains. Toutefois cette règle présente certains défauts (particulièrement quand il s'agit de faire un programme de go). Pour ces raisons, la règle française (qui est aussi la règle chinoise) diffère quelque peu de la règle japonaise. C'est cette règle que nous expliquons ici, et qui doit être implémentée pour votre projet.

### 1.3.1 Déroulement d'une partie

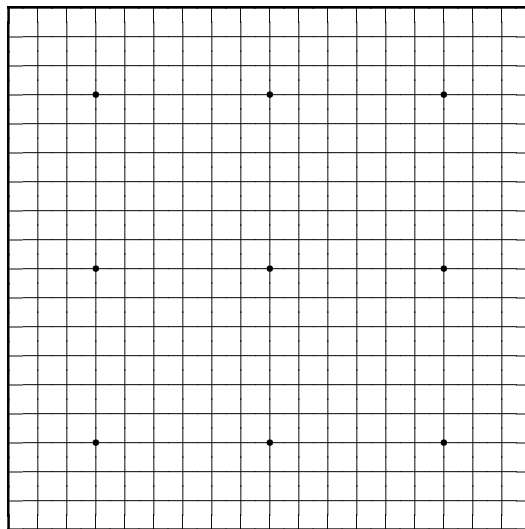
1. Le jeu de go se joue à deux joueurs. L'un d'eux joue avec les pierres noires, l'autre avec les pierres blanches. La taille standard d'un plateau de jeu est de 19 lignes par 19 lignes. On peut toutefois aussi jouer sur des plateaux de 9 lignes par 9 lignes, ou bien de 13 lignes par 13 lignes:



9x9

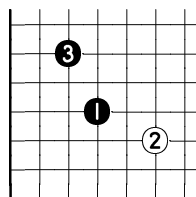


13x13



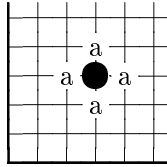
19x19

2. Le jeu se joue tour par tour, le joueur qui a les pierres noires commence à jouer.
3. A chaque tour, le joueur qui a le trait pose une pierre sur une intersection du plateau:



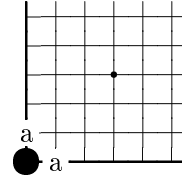
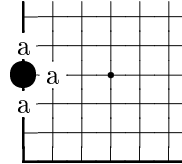
Ici noir joue en 1, blanc répond en 2 et noir répond en 3.

4. Une fois posée, une pierre ne peut être déplacée.
5. Une pierre posée possède un nombre de libertés égal au nombre d'intersections vide en haut, en bas, à droite et à gauche.

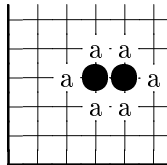


Les **a** indiquent les libertés de la pierre noire.

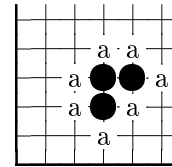
6. Une pierre posée sur un bord n'a que trois libertés. Une pierre posée dans un coin n'en a que deux:



7. Quand la liberté d'une pierre est prise par une pierre de la même couleur, les deux pierres forment alors un **groupe** de pierres, dont les libertés se cumulent. On dit que les pierres d'un même groupe sont **connectées** entre elles:

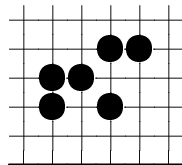


Groupe de deux pierres noires ayant six libertés



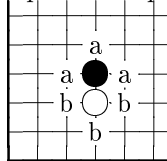
Groupe de trois pierres noires ayant sept libertés

8. Attention : Les pierres ne se connectent pas en diagonal:



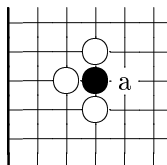
Ici, il y a un groupe de trois pierres noires, un groupe de deux pierres noires, et un groupe d'une pierre noire.

9. Quand un joueur pose une pierre sur la liberté d'un groupe adverse, celle-ci disparaît:

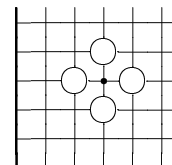


Les **a** indiquent les libertés de la pierre noire (trois libertés), et les **b** indiquent les libertés de la pierre blanche (trois libertés).

10. Quand un groupe n'a plus de liberté, il est capturé par le joueur adverse.



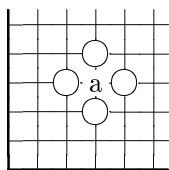
Si blanc joue en **a**, la pierre noire sera capturée



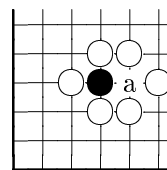
La pierre noire est capturée

11. Quand une pierre est capturée, on l'enlève du plateau et elle devient un **prisonnier** de l'adversaire.

12. Il est interdit de se suicider, c'est à dire de poser une pierre qui serait immédiatement capturée:

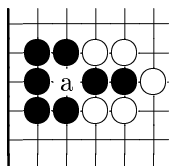


Noir ne peut pas jouer en a

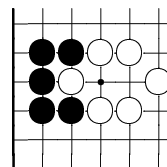


Noir ne peut pas jouer en a

13. Un coup qui capture des pierres n'est jamais un coup suicide:



Blanc peut jouer en a...



... en capturant les pierres noires, la pierre blanche gagne une liberté

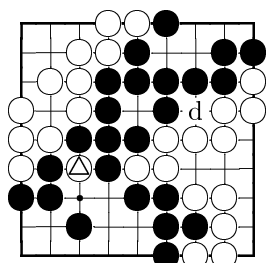
14. Un joueur peut abandonner la partie à la place de jouer son coup. Dans ce cas la partie s'arrête et est gagnée pour son adversaire.
15. A n'importe quel moment, un joueur peut **passer** son coup à la place de jouer.
16. Si deux joueurs passent leur coup l'un après l'autre, la partie s'arrête. Toutefois, si les joueurs ne sont pas d'accord sur les groupes **morts**, la partie doit continuer jusqu'à ce que les choses se clarifient. Nous reviendrons sur ce point dans la section sur les subtilités du jeu.

## Le score

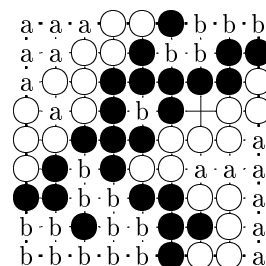
Une fois la partie achevée autrement que par abandon, on compte les points de la manière suivante:

1. Les joueurs commencent par enlever les pierres **mortes** en les prenant comme prisonniers. Nous reviendrons sur ce point important à la fin des règles du jeu.
2. Chaque pierre sur le goban rapporte un point au joueur correspondant.
3. Chaque intersection du goban entourée par des pierres de la même couleur rapporte un point au joueur correspondant. On dit que ces intersections font partie du **territoire** du joueur. Les intersections qui ne sont pas entourées par des pierres de la même couleur sont appelées des **damés**.
4. Le joueur qui a le plus de points gagne la partie. Si chaque joueur a le même nombre de points, la partie est nulle.

Dans l'exemple suivant, les deux joueurs viennent de passer leur coup.



la pierre ⊗ est morte, et d est un damé.



On enlève la pierre morte. Les a sont des points pour blanc et les b sont des points pour noir. Chaque joueur a aussi un point pour chaque pierre de sa couleur, restante sur le goban.

Remarque: En règle japonaise, seul les points de territoire comptent, auxquels les joueurs ajoutent un point par pierre capturée. La règle française peut sembler équivalente: A la place de

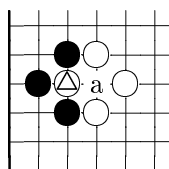
compter les prisonniers comme des points en plus pour nous, ceux-ci sont comptés comme autant de points en moins quand l'adversaire compte ses pierres restantes sur le plateau. En pratique, il y a des différences subtiles entre ces deux manières de compter: la règle française est plus robuste en cas de litige. Nous y reviendrons dans la section sur les subtilités du jeu.

### 1.3.2 Le komi

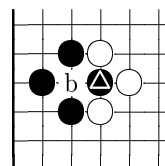
Le joueur qui commence a un avantage sur celui qui ne commence pas. Pour équilibrer cela, on attribut un nombre de points d'avance à celui qui ne commence pas la partie (le joueur blanc). Ce nombre, appelé **komi** est actuellement de 7,5 dans les règles françaises (ce qui empêche en particulier les matchs nuls).

### 1.3.3 Le ko

Cette règle a pour but d'empêcher la possibilité de faire des parties infinies. Supposons que nous sommes dans une situation où un joueur a la possibilité de capturer exactement une pierre adverse en position b, en posant une pierre en position a. Supposons que le joueur adverse puisse faire la même chose à l'inverse: Si il pose une pierre en position b, celle-ci capturera la pierre en position a qui vient d'être posée, et seulement celle là. Dans ce cas, le joueur adverse n'a pas le droit de jouer en position b ce tour-ci, mais doit attendre au moins un tour pour le faire:



Ici noir peut jouer en a pour capturer ⊕



Blanc ne peut pas répondre immédiatement en b pour capturer ⊙: c'est la règle du ko.

Ce genre de situation s'appelle un **ko**. On voit aisément que sans ça, chaque joueur pourrait jouer le même coup tour à tour indéfiniment, ce qui bloquerait la partie.

### 1.3.4 Empêcher les parties infinies

La règle du ko est suffisante dans la plupart des cas pour empêcher les parties infinies, mais pas dans tous les cas. Il y a plusieurs autres exemples, le plus connu étant la situation de triple ko:

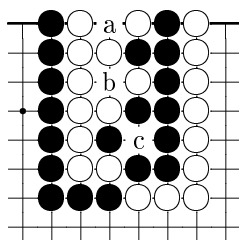


diagramme 1 : triple ko

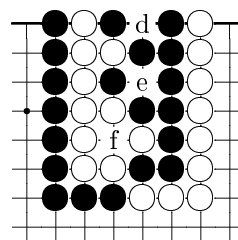


diagramme 2 : triple ko

Dans le diagramme 1 ci-dessus, il y a trois kos distincts, en a, en b et en c. Supposons que ce soit à noir de jouer. Il peut capturer en a, puis blanc peut capturer en c et noir peut capturer en b. On arrive alors dans le diagramme 2. Comme blanc a joué un coup depuis que noir a capturé en a, il peut à son tour capturer en d, puis pour les mêmes raisons noir peut capturer en f et blanc peut capturer en e. On revient alors au diagramme 1, et on peut recommencer à jouer ainsi indéfiniment.

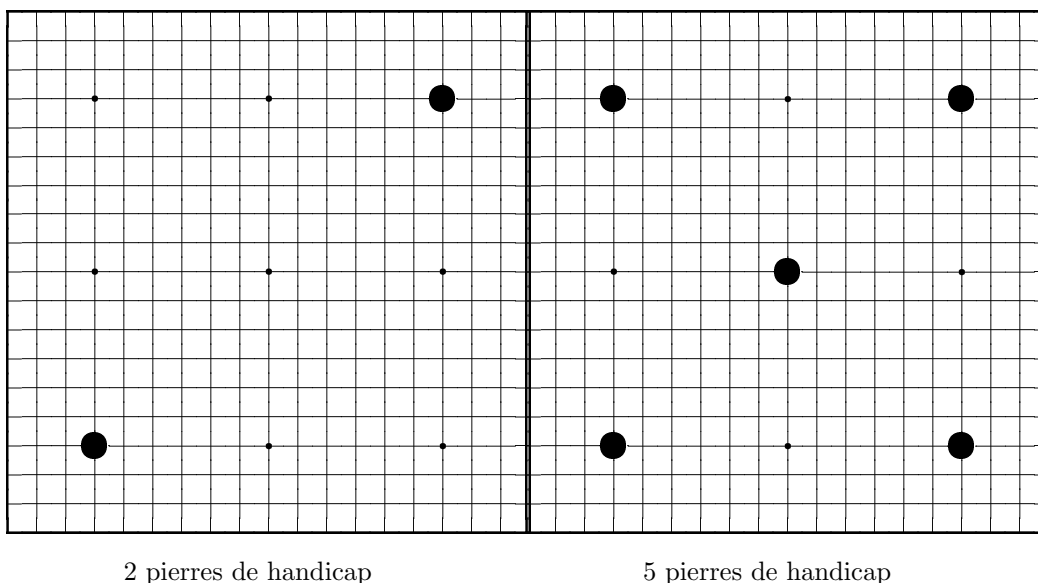
Pour empêcher définitivement toutes les parties infinies, on introduit la règle suivante: Si en jouant un coup, le plateau de jeu arrive dans une configuration déjà rencontrée pour la troisième fois, alors ce coup est interdit.

Remarque: En règle japonaise, si une même configuration arrive trois fois de suite, la partie est match nul. En pratique, ce genre de situation est extrêmement rare.

### 1.3.5 Le handicap

Quand il y a une différence de niveaux entre deux joueurs, il est possible de donner des pierres d'avance au joueur le moins fort. Dans ce cas le joueur ayant des pierres d'avance est toujours le joueur noir. On appelle ces pierres les pierres de **handicap**. Une partie peut se jouer de une à neuf pierres de handicap sur un goban 19x19, et de une à cinq pierres de handicap sur un 13x13 ou un 9x9. Une pierre de handicap correspond à jouer sans komi (le joueur noir commence à jouer normalement et il n'y a pas de point d'avance pour blanc).

Pour  $n > 1$  pierres de handicap, le joueur noir pose les  $n$  pierres sur les **hoshis** du **goban**, le plus symétriquement possible:



Ensuite c'est le joueur blanc qui commence à jouer, et il n'y a pas de komi.

### 1.3.6 Le temps

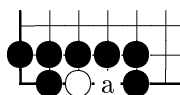
Les parties de go peuvent être très longues. On décide généralement de limiter le temps de réflexion des deux joueurs. Il existe plusieurs types d'horloge pour ça:

- Aucune horloge : Pas de limite de temps pour jouer.
- Horloge absolue : Chaque joueur possède un certain temps de réflexion  $t$  durant son propre tour pour toute la partie. Si il dépasse ce temps, il a perdu.
- Byo-Yomi : Chaque joueur possède un certain temps de réflexion  $t$  durant son propre tour pour toute la partie. Une fois ce temps écoulé, il passe en période de Byo-Yomi. Il a alors  $n$  périodes de Byo-Yomi de  $k$  secondes: le décompte des  $k$  secondes recommence depuis le début à chaque tour tant que le joueur joue en moins de  $k$  secondes. Si le joueur dépasse les  $k$  secondes durant son temps de réflexion, on lui enlève une période de Byo-Yomi (si il dépasse  $2k$  secondes on lui en enlève 2, etc...). Si le joueur arrive à zéro période de Byo-Yomi, il a perdu la partie.

## 1.4 Les subtilités du jeu

### 1.4.1 Les groupes morts

Au go, il arrive souvent qu'un joueur ne prenne pas la peine de capturer un groupe de pierres quand il est certain de pouvoir le faire, quoi que joue son adversaire:



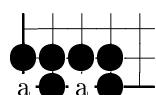
Ici noir est sûr de pouvoir capturer la pierre blanche en jouant en **a**.

Dans l'exemple ci-dessus, noir ne jouera normalement pas en **a**. Ainsi à la fin de la partie, on enlèvera la pierre blanche qui sera un prisonnier de noir, et comptera donc un point: pour le score final, durant le décompte des pierres blanches restantes sur le goban, blanc en aura une de moins. Le groupe noir rapportera ensuite trois points de territoire, un pour chaque intersection vide, ce qui fait en tout quatre points. Les groupes de pierres dont on est sûr qu'ils seront capturés quoi qu'il arrive sont des groupes **morts**.

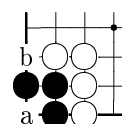
Le cas ci-dessus est simple à déterminer, mais il existe des cas plus compliqués. En particulier, si les deux joueurs ne sont pas d'accord sur le statut d'un groupe : mort ou vivant, c'est au joueur qui pense que le groupe est mort de le prouver, en le capturant. On voit ici l'avantage de la règle française sur la règle japonaise: toujours dans l'exemple ci-dessus, si noir joue en **a** pour capturer la pierre blanche, il n'aura que trois points de territoire, car une intersection préalablement vide sera prise par une de ses pierres. Il aurait donc un point de moins en règle japonaise, mais pas en règle française, car la pierre qu'il vient de poser lui rapporte un point: on peut donc régler simplement les cas de litige.

### 1.4.2 Les groupes vivants

Mais comment être sûr qu'un groupe ne peut pas mourir? Pour cela on essaye de faire ce que l'on appelle des **yeux**. Un oeil est une intersection unique entourée par des pierres de la même couleur. Un groupe ayant deux yeux ne peut jamais être capturé:



Chaque intersection en **a** est un oeil. Blanc ne pourra jamais capturer le groupe noir, car celui-ci a deux yeux.

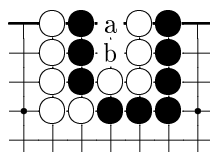


Le groupe noir n'a qu'un oeil. Blanc ne pas jouer en **a** car c'est un suicide. En revanche il peut d'abord jouer en **b**, et ensuite en **a**. Les pierres noires seront alors capturées.

Chaque joueur essaiera donc de faire des points de territoire avec des groupes ayant deux yeux. A la fin de la partie, il ne devrait plus rester que des groupes avec deux yeux, ou la possibilité de les faire facilement, quoi que fasse l'adversaire. Il existe toutefois une exception:

### 1.4.3 Les sekis

Ce sont des situations où plusieurs groupes adverses n'ont pas deux yeux, mais où aucun des deux joueurs n'a intérêt à faire un coup pour essayer de faire une capture:



Situation de seki: si un des deux joueurs joue en **a** (respectivement en **b**), l'autre capture ses pierres en jouant en **b** (respectivement en **a**).

## 2 Le projet

Vous devez faire un programme en java, permettant à deux joueurs de faire des parties de go. Le projet est à réaliser individuellement. Il est formellement interdit de recopier des lignes de code d'une tierce personne. Toute tricherie avérée entraînera la note de zéro (l'entraide reste bien sûr autorisée et encouragée). Il est interdit d'utiliser d'autres packages que les packages standards de Java (javax.swing, java.util, etc...), sauf quand cela est spécifié explicitement dans les instructions qui suivent.

Le logiciel **CGoban**, disponible gratuitement sur Internet, peut vous donner une bonne idée de ce à quoi pourrait ressembler votre projet (le logiciel est programmé en Java et doit se lancer avec "Java Web Start").

### 2.1 Modalité de rendu

Le projet est à réaliser seul en utilisant le langage Java.

#### 2.1.1 Le projet

Le projet est à envoyer sur Eprel avant le 09/06/2018 à minuit. Aucune soumission après cette date ne sera acceptée. Votre soumission doit consister en un fichier **.tar** contenant vos fichiers sources. Vous devez aussi y inclure un fichier **AUTHORS** (avec vos noms et prénom) et un fichier **README** qui explique brièvement le contenu de chacun de vos fichiers sources, ainsi que les directives pour lancer le programme (dans quel fichier se trouve le point d'entrée, etc...), et au cas où vous jugez cela nécessaire, des instructions expliquant comment utiliser votre programme.

### 2.2 La représentation interne

Vous devez créer une structure représentant votre jeu de go, avec des méthodes permettant de jouer des coups.

1. Un coup joué doit automatiquement capturer les pierres du goban si nécessaire.
2. Un coup interdit ne doit pas pouvoir être joué (vous devez aussi gérer la règle du ko et la règle qui empêche les parties infinies).
3. Une fois une partie terminée par deux joueurs ayant passé à tour de rôle, vous devez avoir une fonction qui vous indique le vainqueur. On ne demande pas de détecter automatiquement les groupes morts: la fonction doit prendre ces informations en paramètre.

### 2.3 L'historique

Vous devez stocker l'historique de tous les coups joués. Vous devez faire des méthodes permettant de naviguer dans l'historique comme on le souhaite: Laisser la possibilité d'annuler autant de coups que l'on veut, mais aussi les "désannuler". En revanche si on annule un coup et que l'on joue autre chose, cela efface dans l'historique le coup que l'on avait annulé.

### 2.4 L'interface graphique

Vous devez faire une interface graphique pour votre jeu de go. Vous devez en particulier avoir:

1. Un menu permettant de choisir les paramètres de jeu et de commencer une partie.
2. La représentation du jeu, avec le goban, les informations relatives aux deux joueurs (temps restant, nombre de prisonniers, etc...)
3. la possibilité de naviguer dans l'historique des coups.
4. la possibilité de redimensionner votre fenêtre de jeu.



## 2.5 Le jeu

Tout cela ensemble doit permettre à deux joueurs de faire une partie selon les règles françaises énoncées ci-dessus. Vous devez également gérer le choix des pierres de handicap, de la taille du goban, et du type d'horloge. Une fois une partie terminée, vous devez automatiquement permettre aux joueurs de sélectionner les groupes morts, et votre programme doit décompter les points.

## 3 Bonus possibles

### 3.1 Le score

Rajouter la possibilité de pouvoir estimer le score d'une partie en cours. Rajouter aussi la possibilité d'enlever automatiquement les groupes morts quand une partie est finie.

### 3.2 Les fichiers SGF

Vous pouvez gérer la lecture et la sauvegarde de fichiers au format `sgf`. Vous pouvez utiliser une bibliothèque déjà existante pour cela si vous le souhaitez.

### 3.3 Le Go text protocole

Vous devez laisser la possibilité qu'un des deux joueurs (ou les deux joueurs) de votre programme soit une IA. Pour cela il n'y a pas besoin de faire d'IA: votre programme doit pouvoir s'interfacer avec une IA existante utilisant le `Go text protocole`. Vous pouvez utiliser une bibliothèque déjà existante pour cela si vous le souhaitez. Par ailleurs, vous pouvez tester diverses IA open source, comme `Gnu Go` ou encore `Fuego`.

### 3.4 IA

Faites une Intelligence artificielle pour votre jeu de go...

## 4 Evaluation

Il y aura une soutenance pour votre projet. L'évaluation se fera selon les critères suivants:

1. Le respect des règles du jeu (toutes les règles).
2. L'interface graphique (jeu et menu).
3. La bonne gestion des parties (gérer le handicap, le temps, le calcul du score, etc...).
4. La bonne gestion de l'historique.
5. La qualité du code (modularité, propreté, etc...), ainsi que la clarté du rendu (pas de fichiers inutiles, fichier `README` bien expliqué, etc...)
6. La finition (Est-ce que votre programme est facile et agréable à utiliser).
7. La qualité de la présentation durant la soutenance.