

Projet Initiation à l'Algorithmique
HETIC H1 P2019
Jeu de GO

v_1 *

Nora Izri nora.izri@platypus.academy
Franck Lepoivre franck.lepoivre@platypus.academy
Grégoire Puget gregoirepuget@gmail.com



***Première** version pour questions, commentaires et demandes de précisions. Cette version vous permet de commencer votre travail de documentation et de réflexion. Elle est susceptible de très nombreuses évolutions d'ici quelques semaines, notamment en fonction de nos interactions.

Contents

1	Jeu	3
1.1	Origines	3
1.2	Termes	4
1.3	Règles du jeu	5
1.3.1	Déroulement de la partie	5
1.3.2	Capture	5
1.4	Vie et mort	5
1.4.1	Répétition	6
1.4.2	Fin de partie	6
1.4.3	Handicaps	6
1.4.4	Décompte des points	6
2	Travail à réaliser	6
3	Organisation	7
3.1	Encadrement et calendrier	7
3.2	Quelques précisions	8
3.2.1	Constitution des équipes	8
3.2.2	Point intermédiaire	8
3.2.3	Soutenance	8
3.3	Périmètre technologique autorisé	9
3.4	Évaluation, barème	9
3.4.1	Comment gagner des points	9
3.4.2	Comment perdre des points	9
3.4.3	Travail minimum pour espérer avoir 12	9
3.5	Précisions sur l'évaluation et sur le barème	10
3.5.1	Principe général	10
3.5.2	Note collégiale	10
3.5.3	Modulation des notes au sein des équipes	11
3.5.4	Notation	11
4	Packaging du livrable	12
4.1	Contenu du livrable	12
4.2	Contenu du rapport	12
4.3	Livraison	12
5	Bonus	12
6	Références	13

Sujet

L'objectif de ce projet est de réaliser un jeu de Go sous forme d'application Web.

Objectifs pédagogiques

Le projet vous place dans une situation d'ingénierie qui vous oblige à :

1. prendre le temps de bien vous documenter avant de vous lancer : rechercher et synthétiser l'information, devenir expert du sujet ;
2. puis prendre le temps d'analyser le problème, de le décomposer en sous-problème d'en extraire des sous-problèmes implicites ;
3. puis prendre le temps de concevoir le plan général d'une solution à travers un diagramme de classes ;
4. puis organiser le travail en équipe en vous répartissant les tâches de réalisation suivant une feuille de route avec des objectifs intermédiaires qui constituent des étapes synchrones de l'équipe.
5. puis réaliser le travail de telle manière que toute cette analyse, cette conception, cette organisation n'accouchent pas d'une souris.

Toutes ces phases et étapes sont à réaliser en équipe, en coordonnant vos actions et en intégrant leurs fruits (le chef de projet est en charge de cette coordination et de cette intégration).

Nous vous conseillons une approche incrémentale, par itération successives des 5 phases (esprit méthodes agiles), plutôt qu'une approche séquentielle (inefficace et dépassé).

Conseil

Ne pas programmer avant d'avoir analysé, modélisé, réalisé un diagramme de classes, défini le périmètre de son projet et ses potentielles extensions.

1 Jeu

Le jeu de GO est un jeu de stratégie basé sur la réflexion. Le but étant de s'emparer du plus grand territoire possible du plateau. Le mieux pour résumer une partie c'est cette citation :

**"Tout le succès d'une opération réside dans sa préparation."
Sun Tzu, L'Art de la guerre.**

1.1 Origines

Le jeu de Go est originaire de Chine. C'est là la seule certitude à son sujet. Il a été inventé au plus tard en 1200 avant JC. Certaines légendes estiment que ça remonte à -4000 avant JC. Ces légendes attribues sa création tantôt à des empereurs pour éduquer leur fils, tantôt à un Vassal pour distraire son empereur.

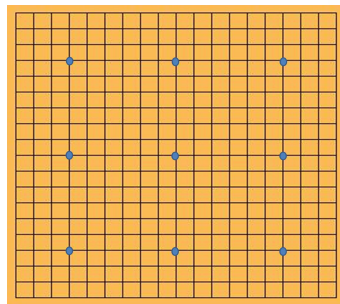


Figure 1: Goban 19x19

C'est donc un jeu qui a su traverser les âges, les crises (seconde guerre mondiale au Japon) et les révolutions (révolution culturelle en Chine) et ce depuis plus de 3 millénaires qui vous est proposé aujourd'hui.

1.2 Termes

Tout d'abord les lignes et les colonnes sont les traits et pas les cases. L'image ci-contre (figure 1) montre 19 lignes et autant de colonnes (et pas 18).

Goban : il s'agit du plateau (figure 1). Traditionnellement composés de 19 lignes et 19 colonnes, il forme ainsi 361 intersections. Il existe aussi des versions simplifiées de 13x13 (169 intersections) ou 9x9 (81 intersections). Attention, le goban n'est pas carré. En effet, chaque "case" fait 24mm de long pour 22mm de large (par abus de langage, on acceptera "case carrée").

Hoshi : Il s'agit des points foncés aux intersections de la figure 1, c'est à dire le nom des neuf points renforcés d'un goban. Une de leurs fonctions est d'indiquer les endroits où l'on place traditionnellement les pierres d'handicap. Ils facilitent également le repérage des intersections. Le terme hoshi est employé souvent pour désigner les points 4-4. Mais les points 4-10 et le point 10-10 sont aussi des hoshis.



Figure 2: Pierres

Pierres : ce sont les pions. Le jeu est composé de pierres blanches et de pierres noires comme l'illustre la figure 2.

Intersections/pierres voisines : 2 intersections sont voisines si elles se suivent sur la même ligne (ou colonne). Dans la figure 3, *a* et *b* sont des intersections voisines. *b* et *c* ne le sont pas. Deux pierres sont donc voisines si elles sont sur des intersections voisines.

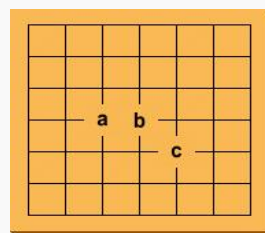


Figure 3: Intersections voisines

Chaîne : Une chaîne est un ensemble de pierres de même couleur voisines de proche en proche. Celles-ci ne sont pas forcément en ligne.

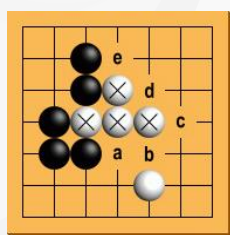


Figure 4: Libertés d'une chaîne

Libertés : les libertés d'une chaîne sont toutes les intersections inoccupées voisines des pierres de cette chaîne. La figure 4 montre les libertés de la chaîne blanche.

Territoire : C'est l'ensemble d'intersections inoccupées voisines de proche en proche, délimitées par des pierres de même couleur. La figure 5 montre un territoire de 7 intersections des noirs. Point important sur les territoires, le bord du goban offre une limite naturelle au territoire. En effet, il n'y a pas d'intersections au-delà (un peu comme la mer au temps de l'homme de Néandertal).

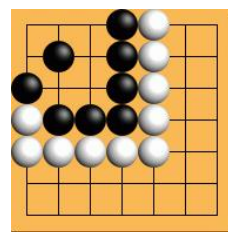


Figure 5: Exemple de territoire

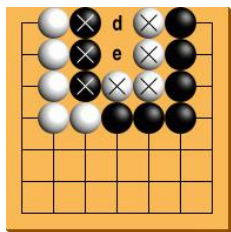


Figure 10: Seki

Un seki (traduit par impasse) est une position où aucun des deux joueurs ne peut tenter de prendre l'autre sans se faire prendre avant. La figure 10 montre un cas de seki simple. Que ce soit noir ou blanc joue en position *d* ou *e*, entrainera l'adversaire à jouer l'autre possibilité et ainsi prendre toutes les pierres. Aucun des 2 joueurs n'a donc intérêt à jouer (*d* ou *e*). les pierres marquées d'un X sont donc dites vivantes par seki et les intersections *d* et *e* sont des intersections neutres.

1.4.1 Répétition

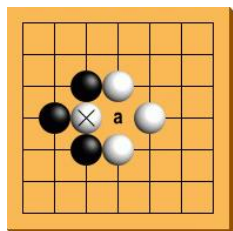


Figure 11: Situation initiale

Le cas suivant est un exemple simple de répétition (avec l'exemple du ko). La figure 11 montre la situation initiale. Le noir va jouer en *a* pour prendre la pierre blanche marquée d'un X.

La figure 12 montre la situation ainsi obtenue. Afin d'éviter que chaque joueur joue la même chose, il est interdit à blanc de jouer tout de suite en *b*. il doit donc jouer ailleurs sur le goban.

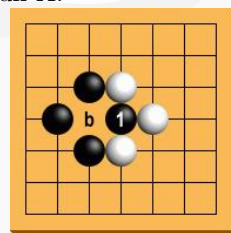


Figure 12: Répétition

A noter que blanc, a tout intérêt à forcer noir à jouer ailleurs que *b* pour ne pas lui bloquer cette opportunité.

1.4.2 Fin de partie

La partie s'arrête lorsque les deux joueurs passent consécutivement. Un comptage des points s'impose. Un point est rapporté par chacune des pierres d'un joueur encore présentes sur le goban, ainsi que chaque intersection du territoire.

Le gagnant est celui qui a le plus de points.

1.4.3 Handicapes

Il est possible de donner un handicap en cas de différence de niveau entre 2 joueurs. Pour cela, le moins fort commence avec les noirs et pose sur le terrain entre 2 et 9 pierres pour commencer. Plus le nombre de pierres posées au début est important, plus le handicap pour blanc est grand.

1.4.4 Décompte des points

Le décompte des points correspond à la somme des pierres sur le goban (1 point par pierre) et 1 point par intersection d'un territoire.

Les noirs ayant un avantage au départ, les blancs ont un bonus de 7.5 points dès le départ (0.5 pour éviter les égalités).

La couleur qui a le plus de point en fin de partie gagne.

2 Travail à réaliser

Le projet consiste à permettre à réaliser un jeu de go.

Le programme sera développé dans le langage objet de votre choix (C++, Java, PHP, etc). Il n'est pas nécessaire de faire appel à une librairie graphique, cependant cela est vivement conseillé.

Il est obligatoire que votre programme :

- Puisse charger une partie en prenant un fichier "partie.txt".
- Puisse sauvegarder une partie dans le même fichier "partie.txt".
- Permette au joueur de jouer contre un humain.
- Permette à un joueur d'affronter la machine (IA)
- Interdire au joueur de placer une pierre s'il n'en a pas le droit.
- Compte les points.
- Identifie la fin de la partie et indique le gagnant.

Le format du fichier texte (.txt) est le suivant :

Pour un goban de 9x9 il aura 10 lignes (9 lignes + 1) avec 9 caractères par ligne sauf la 10e ligne qui n'aura qu'un caractère. Les 9 premières lignes représenteront le goban en cours avec ' ' (espace) pour une intersection sans pierre, 'n' (*n* majuscule !) pour les pierres noires et 'b' (*b* minuscule !) pour les pierres blanches. La dernière ligne contiendra uniquement 'n' ou 'b' selon à qui est le tour.

La partie joueur contre machine. Plus les étudiants iront loin dans les possibilités de l'IA, plus il y aura de points accordés. L'IA la plus basique étant de mettre une pierre au hasard sur une intersection où cela est possible.

3 Organisation

3.1 Encadrement et calendrier

Votre projet est encadré par vos deux enseignants respectivement d'initiation algorithmique, Franck Lepoivre et de développement Web Grégoire Puget, à l'occasion de 6 doubles séances de 3h réparties sur 6 semaines, soit 36h d'accompagnement.

Nous estimons que le projet devrait représenter pour chacun d'entre vous, un temps de travail personnel à peu près équivalent à 36h, soit environ 70h par élève et environ 280h par équipe.

Les séances s'organiseront de la manière suivante :

Semaine 19 séances du 05 et 07/05, respectivement avec vos deux enseignants : prise de connaissance du sujet, constitution des équipes, travail de recherche d'information, demandes d'éclaircissements relativement au sujet.

Semaine 21 séances du 19 et 20/05, respectivement avec vos deux enseignants : début de l'analyse, de la conception et de l'élaboration de la roadmap. Sollicitation de vos enseignants en séance pour obtenir leurs conseils, mais pas leurs prescriptions : vous devez faire vos propres choix pour lesquels ils se contenteront de vous aider à peser le pour et le contre.

Semaine 22 séances du 26 et 28/05, respectivement avec vos deux enseignants : point d'avancement intermédiaire : les équipes travaillent en salle, mais présenteront à tour de rôle, à leur deux enseignants leur état d'avancement et idéalement leur premier prototype. Ce point d'avancement donnera lieu à 0 (rien de concret), 1 (un bon début) ou 2 (très sérieuse amorce) points bonus sur la note finale. Vos deux enseignants se concerteront et confronteront leurs appréciations pour coordonner leur effort d'accompagnement.

Semaine 23 séances du 02 et 04/06, respectivement avec vos deux enseignants : vous avancez aussi vite, aussi loin et aussi efficacement que possible. A ce stade, nous nous contenterons d'intervenir pour vous aider à résoudre des problèmes techniques ou des problèmes d'équipe bloquants.

Semaine 24 séances du 09 et 11/06, respectivement avec vos deux enseignants : vous convergez vers la finalisation de votre projet. Terminer n'est jamais évident : problèmes d'intégration de parties, tests et corrections de bugs, documentation à finaliser, finitions (notamment du design), préparation de la soutenance. Ce n'est plus le moment de la réflexion mais c'est le moment d'avoir le sens de la deadline. La livraison de vos travaux doit impérativement être effectuée le Mercredi 24 Juin avant que ne retentissent les 12 coups de minuit.

Semaine 26 double séances du 26/06, avec vos deux enseignants, les 30 équipes soutiennent devant l'ensemble de la promotion. Cela permet de prendre le temps d'une soutenance de 10 minutes. Notation collégiale (sur un compartiment de la note, cf. barème), c'est-à-dire que toutes les équipes présentes notent toutes les équipes présentes à l'aide d'un formulaire en ligne qui permet d'évaluer chaque équipe sur la base de 6 critères.

3.2 Quelques précisions

3.2.1 Constitution des équipes

Seuls les équipes de 4 sont autorisés. Il peut y avoir, à titre dérogatoire, une ou deux équipes X dans les groupes de TD dont le nombre d'étudiants n'est pas un multiple de 4.

Les équipes seront constituées dès le lancement du projet. Au sein de chaque équipe devront être déterminées des responsabilités individuelles (tous les aspects du projet restant basés sur un travail d'équipe) :

un chef de projet en charge de la roadmap, de la coordination, etc.

un responsable du design en charge de la conception graphique, de l'ergonomie, de l'expérience utilisateur.

un responsable technique en charge de choix de conception technique et du choix des technologies, frameworks et outils à mettre en œuvre pour la réalisation, de l'intégration et des tests des parties réalisées par chacun des membres de l'équipe.

un responsable de la communication en charge des rapports (écrits), de la soutenance (oral). Il joue un rôle déterminant pour valoriser ce que l'équipe a réalisé. Le responsable communication s'assure par exemple de la qualité des rapports à rendre (orthographe, structures, synthèse, etc), de la scénarisation et des répétitions de la soutenance qui doit être un spectacle dont le point d'orgue est la démonstration du produit.

En cas de dysfonctionnement problématique au sein d'une équipe, n'attendez pas le dernier moment pour l'évoquer : nous pouvons vous aider à trouver une solution.

3.2.2 Point intermédiaire

Les attentes pour le point intermédiaire sont : travail de recherche d'information bien avancé, modélisation / conception des structures de données et des algorithmes bien avancées, un prototype qui fonctionne sur un périmètre fonctionnel restreint (par exemple pouvoir charger et jouer à un labyrinthe 2D à maille carrée, sans monstres ni bonus), une équipe où les rôles ont été partagés relativement à un plan de réalisation défini.

3.2.3 Soutenance

La soutenance est d'abord un exercice de communication.

20 minutes passent très rapidement, et il convient de structurer votre présentation et de la répéter pour la tester et la calibrer.

Votre enjeu est de démontrer :

1. la valeur ajoutée fonctionnelle de votre programme,
2. la performance technique que vous avez réalisée.

Après avoir rappelé le contexte et les enjeux du projet (ce qu'il fallait faire, les principales difficultés à surmonter), la première partie de votre présentation doit être résolument orientée démonstration utilisateur (des fonctionnalités) : il faut 'vendre' votre programme à ses utilisateurs potentiels en déroulant un scénario d'utilisation sous forme d'une démonstration, que vous commentez en évitant tout discours technique (story telling).

Cette première partie doit prendre environ 10 minutes.

La seconde partie de votre présentation doit démontrer la performance technique que vous avez réalisée, notamment en :

1. présentant les questions que vous vous êtes posées et les choix que vous avez effectués, en les justifiant, pendant la phase d'élaboration,
2. en montrant ensuite quels ont été vos choix de conception, idéalement avec le support de diagrammes d'architecture (de principe), et éventuellement de quelques diagrammes techniques.

Vous conclurez en expliquant comment vous vous êtes répartis le travail et comment a fonctionné l'équipe, puis en résumant les points forts et les améliorations possibles de votre réalisation.

Cette seconde partie doit prendre environ 5 minutes, au pire 7 minutes.

Elle laisse le temps à 5 à 3 minutes de questions / réponses.

Il est important d'amener votre propre portable (un par équipe) pour pouvoir exécuter et projeter votre démonstration ainsi que vos documents supports.

3.3 Périmètre technologique autorisé

Vous pouvez utiliser les technologies de votre choix.

Les technologies, frameworks, outils utilisés devront être référencés dans votre livrable, et notamment rappelées dans les annexes de votre rapport.

3.4 Évaluation, barème

3.4.1 Comment gagner des points

Critères par ordre décroissant d'importance :

- Le programme fait ce qu'on attend de lui (partiellement, complètement, avec des plus).
- L'interaction utilisateur est ergonomique (facilité d'utilisation) et esthétique.
- Le code source est efficace (structures de données et algorithmes efficaces).
- Le code source est propre (indentations, identifiants intelligibles, commentaires bien dosés).

3.4.2 Comment perdre des points

Critères par ordre décroissant de gravité :

- Même code, même partiel, livré par plusieurs équipes : 0 pour les équipes impliquées.
- Retard de livraison : 3 points par jour au *pro rata temporis*.
- Non respect des consignes de ¹ du livrable : jusqu'à -3 pts suivant les cumuls.

3.4.3 Travail minimum pour espérer avoir 12

Le périmètre fonctionnel minimal consiste à permettre de jouer au go selon les règles de base (on peut supprimer quelques conditions un peu spéciales).

Ce périmètre fonctionnel minimum permet d'être noté sur 12 points : obtenir 12 serait alors la marque d'une réalisation parfaite sur ce périmètre (fonctionnement, ergonomie, etc).

Pour aller au delà, cela va dépendre :

1. de la complétude fonctionnelle par rapport aux règles,
2. de la qualité du design
3. de fonctionnalités avancées (pas forcément exhaustif) :
 - a jeu en réseau,
 - b jeu contra la machine IA

¹voir section suivante.

3.5 Précisions sur l'évaluation et sur le barème

3.5.1 Principe général

Trois notes :

1. Note A de Franck Lepoivre,
2. Note B de Grégoire Puget,
3. Note C collégiale par les pairs (chaque équipe note chaque équipe sauf elle-même).

La note de trimestre (contrôle continu et partiel) pour chacun des deux modules « Initiation à l'algorithmique » et « Développement Web » est constituée de la moyenne entre la note de l'enseignant et la note collégiale, soit respectivement $(A + C)/2$ et $(B + C)/2$.

3.5.2 Note collégiale

Les ordres de passage ont été donnés dès le lancement du projet.

Tous les élèves doivent être présents pour toute la durée de la séance de soutenance. En début de séance, chaque élève reçoit une fiche qui lui permet de noter toutes les équipes (sauf la sienne) suivant 6 critères et d'après une échelle qualitative :

- Qualité de la communication.
- Qualité de la démonstration.
- Qualité du gameplay.
- Qualité du design.
- Performance technique.
- Qualité de l'équipe (est-ce une équipe qui marche ?).

Échelle les échelons donnent une indication qualitative sur le critère évalué :

A Il y a tout ce qu'il faut

A+ c'est même parfait / spectaculaire !

A c'est excellent sans être parfait !

A- c'est très bien mais il y a un petit point qui cloche !

B Il manque quelque chose d'important...

B+ mais c'était à deux doigts du A.

B mais c'est pas mal quand même.

B- et c'est moyen.

C Il y a le début de quelque chose...

C+ à deux doigts de la moyenne.

C peut mieux faire.

C- pour ne pas mettre moins.

X+ merci d'être venu.

X ∅.

À chaque échelon est associé un pourcentage des points accordés sur le critère évalué :

A+	A	A-	B+	B	B-	C+	C	C-	X+	X
100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	0%

3.5.3 Modulation des notes au sein des équipes

À fin d'équité, pour moduler les disparités éventuelles au sein des équipes, chaque membre de l'équipe peut choisir de transférer une partie de ses points vers un autre membre, qui aurait par exemple, largement plus contribué au projet que les autres.

Par exemple, si John, Bertrand, Alphonse et Gertrude ont obtenu 11, mais que Gertrude a fourni l'essentiel du travail, et secondairement Bertrand, tandis que John et Alphonse, moins forts techniquement, ont fait ce qu'ils pouvaient, mais sans toutefois mériter la même note que Gertrude et Bertrand, il est possible, avec l'accord de l'ensemble des membres de l'équipe, que John et Alphonse transfèrent (jeu à somme nulle) chacun 1,5 point vers Gertrude et 0,5 point vers Bertrand, passant ainsi leurs notes à 9, et celles de Gertrude et Bertrand, respectivement à 14 et 12.

Si une telle modulation est décidée entre les membres de l'équipe, elle devra être notifiée au plus tard le 28/06 à 23h59 par la voix du chef de projet, par un email adressé aux deux enseignants, avec pour objet « H1 - Modulation note projet – équipe numéro » dont le corps indique précisément les transferts de points (somme nulle). Par exemple :

Objet : H1 - Modulation note projet – équipe 04

Gertrude : +3

Bertrand : +1

John : -2

Alphonse : -2

3.5.4 Notation

2 pts bonus max en fonction de l'avancement au moment du point de la semaine 1

+

3 pts sur le synopsis

-

Pénalité de retard (le cas échéant)

+

Respect des consignes de remise, qualité du livrable (packaging, README, etc.).	Qualité du code (conventions d'écriture, dosage commentaire, efficacité des algorithmes).	Dossier technique analyse fonctionnelle, architecture technique.	Jouabilité : élaboration / scénarisation du gameplay.	Intelligence Artificielle,	Bonus (prévu ou imprévu), éléments fonctionnels supplémentaires qui démarquent significativement (multi-joueurs jeu en réseau, etc.)
2 pts	2 pts	4 pts	4 pts	4 pts	4 pts

Noté sur 25 sans coefficient de normalisation.

4 Packaging du livrable

4.1 Contenu du livrable

- Les fichiers sources de votre programme.
- SURTOUT PAS D'EXÉCUTABLE !
- Votre documentation (Rapport et autres documents) au format PDF.
- Un fichier `README.txt` qui comprend impérativement deux sections :
 - Les instructions d'installation, déploiement, configuration de votre application.
 - Un état d'avancement honnête et fiable de vos travaux en 10 lignes maximum.
 - Les références aux sources qui ont pu inspirer votre conception, la reprise de code étant formellement interdite.
 - Et en option, une mise en avant de fonctionnalités supplémentaires que vous aurez choisi de réaliser dans le but d'obtenir une meilleure note.

4.2 Contenu du rapport

Un **sommaire** détaillé avec pages numérotées.

L'analyse fonctionnelle générale D'environ cinq pages, elle explique quelles sont les principales données traitées et quelles sont les principales fonctionnalités du programme, comment on peut les regrouper, comment elles sont organisées dans le programme. Cette analyse aboutit à la présentation des modules fonctionnels qui sont décrits en détail dans l'analyse détaillée. L'analyse générale doit préciser le ou les choix importants que vous avez faits pour la conception de votre projet.

L'analyse fonctionnelle détaillée D'environ dix pages, elle précise, pour chaque module, son utilité, ses entrées et ses sorties, les données qu'il modifie. Il faut décrire les données utilisées et les algorithmes mis au point avec rigueur, concision et clarté, le tout accompagné de schémas. Cette description est abstraite et s'appuie sur le langage algorithmique utilisé en cours et en TD. C'est un travail de synthèse, de pédagogie et de communication; les extraits de code ou autre sont proscrits de vos rapports.

Annexes Elles présentent au moins les technologies, frameworks et outils employés. Et comme option recommandée, bibliographie et webographie listent les références qui vous ont permis d'étayer votre analyse et votre conception.

4.3 Livraison

Une seule remise des travaux par équipe !! (la double remise peut être sanctionnée dans la composante respect-des-consignes de la note). Remise au plus tard le **Mercredi 24/06 à 23h59**. Dès le lendemain, application d'une pénalité de 3 points par jour (au *prorata temporis*). Le livrable se présente sous la forme d'une archive ZIP de nom : `PROJET GO H1 GP NOM 1 NOM 2`.ZIP envoyé par email à chacun de vos deux tuteurs de projet.

Objet `PROJET GO H1 GP NOM 1 NOM 2`

A

Grégoire Puget : gregoirepuget@gmail.com

Franck Lepoivre : franck.lepoivre@gmail.com,

5 Bonus

Si des projets ont des IA suffisamment élaborées, un bonus de 2 points sera accordé au binôme qui arrivera à battre toutes les autres IA. Pour cela il est impératif de respecter le format du fichier de sauvegarde et de chargement (le non respect de cette consigne est éliminatoire pour obtenir le bonus).

6 Références

- http://fr.wikipedia.org/wiki/Jeu_de_go
- http://fr.wikipedia.org/wiki/R%C3%A8gles_du_jeu_de_go
- <http://jeudego.org/>
- <http://www.jeudego.com>

