**Project Go Game**

Recently, Google DeepMind has created the first artificial intelligence for the go game, able to beat a professional player partly balanced. To celebrate this remarkable advance, you will make a go game project. Rest assured, you will not be asked to do an AI.

**1 Prerequisites - The game of Go**

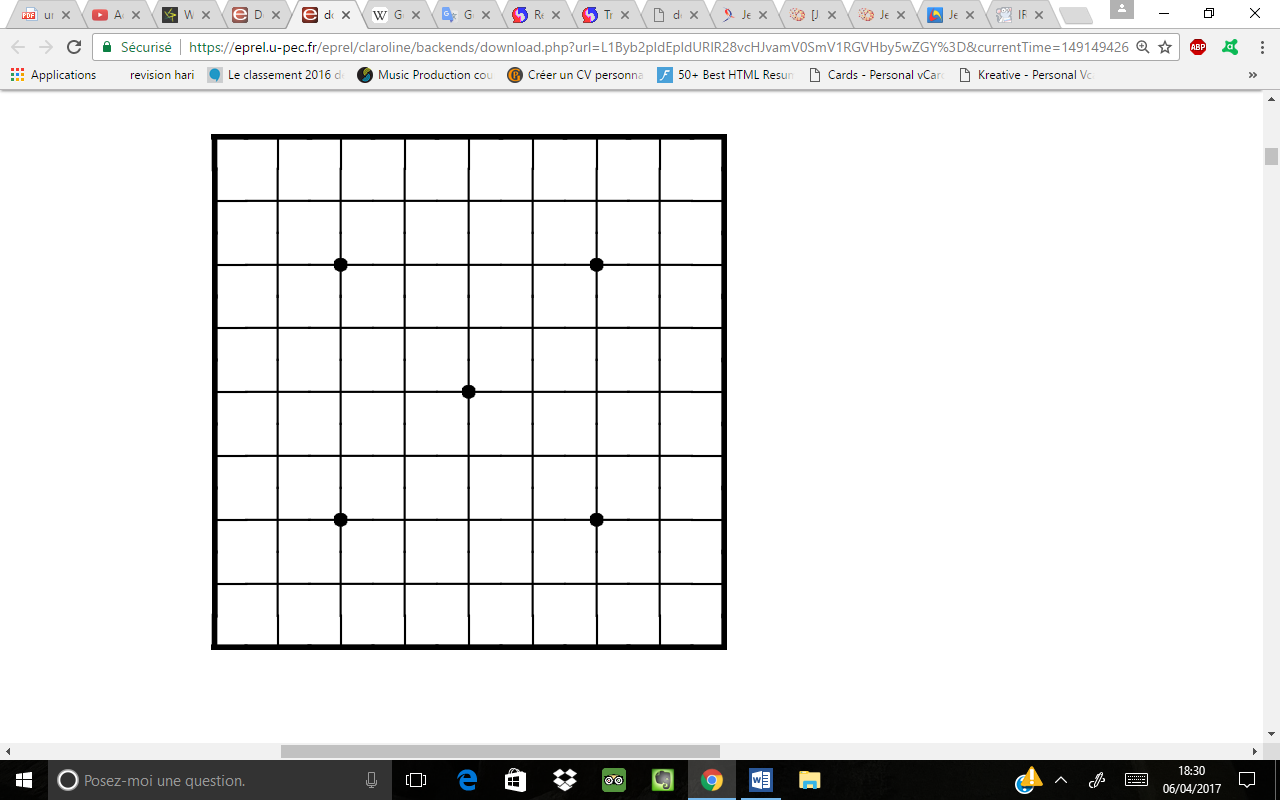
**1.1 Introduction**

Go is a board game from China. He opposes two opponents who place in turn black and white stones respectively on the intersections of a grid-shaped game board called goban. The goal is to control the game plan by constructing territories. The encircled stones become prisoners, the winner being the player with the most territories and prisoners.

This is the oldest known abstract combinatorial strategy game. Despite its age, go continues to enjoy great popularity in China, Korea and Japan. In the rest of the world, where his discovery is recent, his notoriety is increasing. Its success stems as much from the simplicity of its rules as from its great combinatorial richness and its strategic depth.

Source: Wikipedia

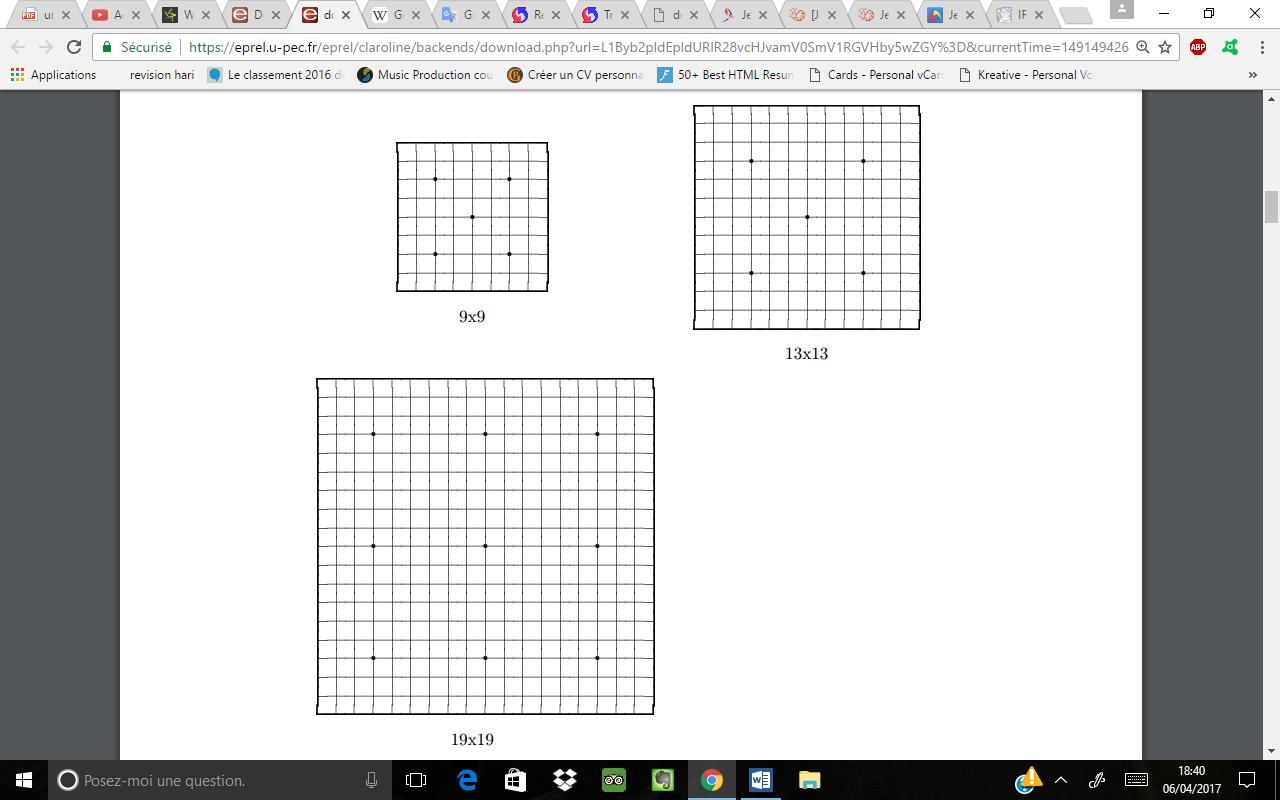
**1.2 Hardware**

* The **goban**: this is the game board:
* The **hoshi** are the blackheads on the goban.
* **Stones**: these are "pieces" of some kind. One player uses the blacks, the other uses the whites.

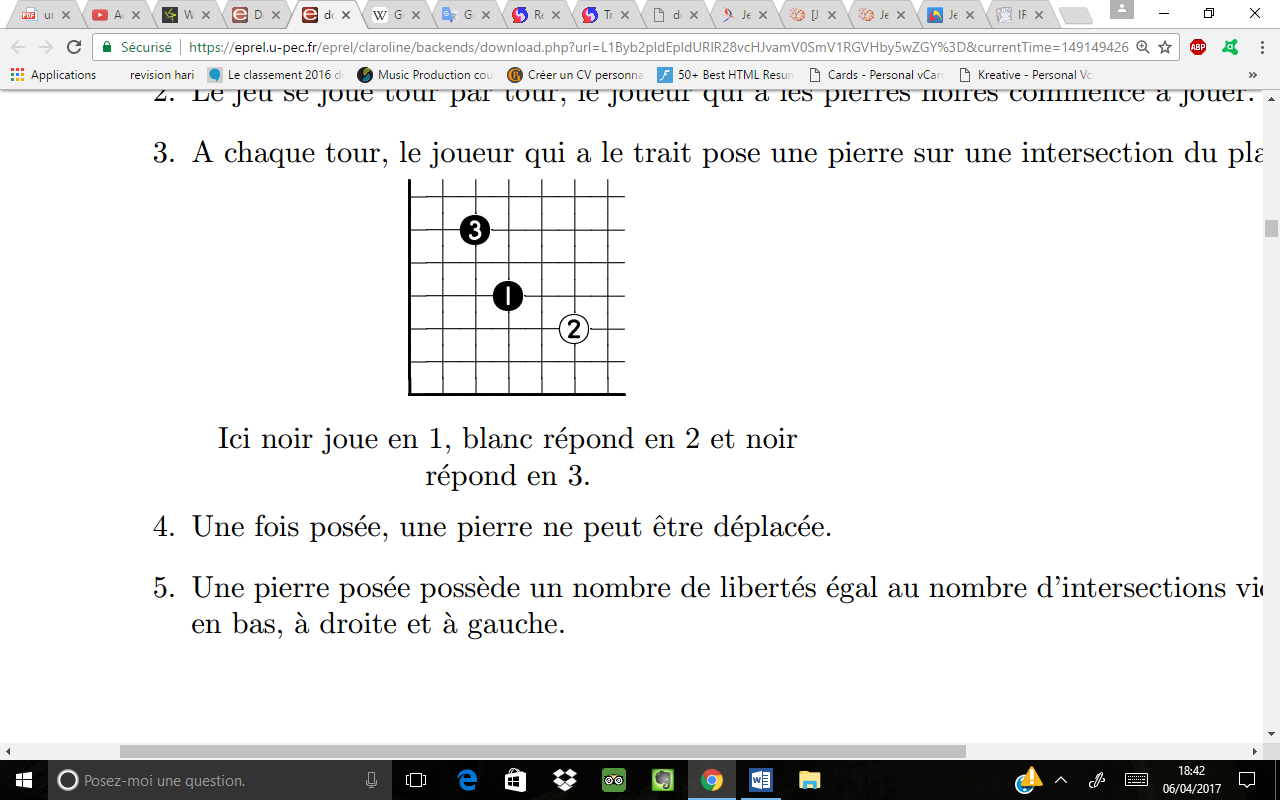
**1.3 The rules of the game**

There are several minimal variations of the rules of the game go. Most European players have learned the game with the Japanese rule, which is the most natural for humans. However this rule has some flaws (especially when it comes to making a go program). For these reasons, the French rule (which is also the Chinese rule) differs somewhat from the Japanese rule. It is this rule that we explain here, and that must be implemented for your project.

**1.3.1 Conduct of a Game**

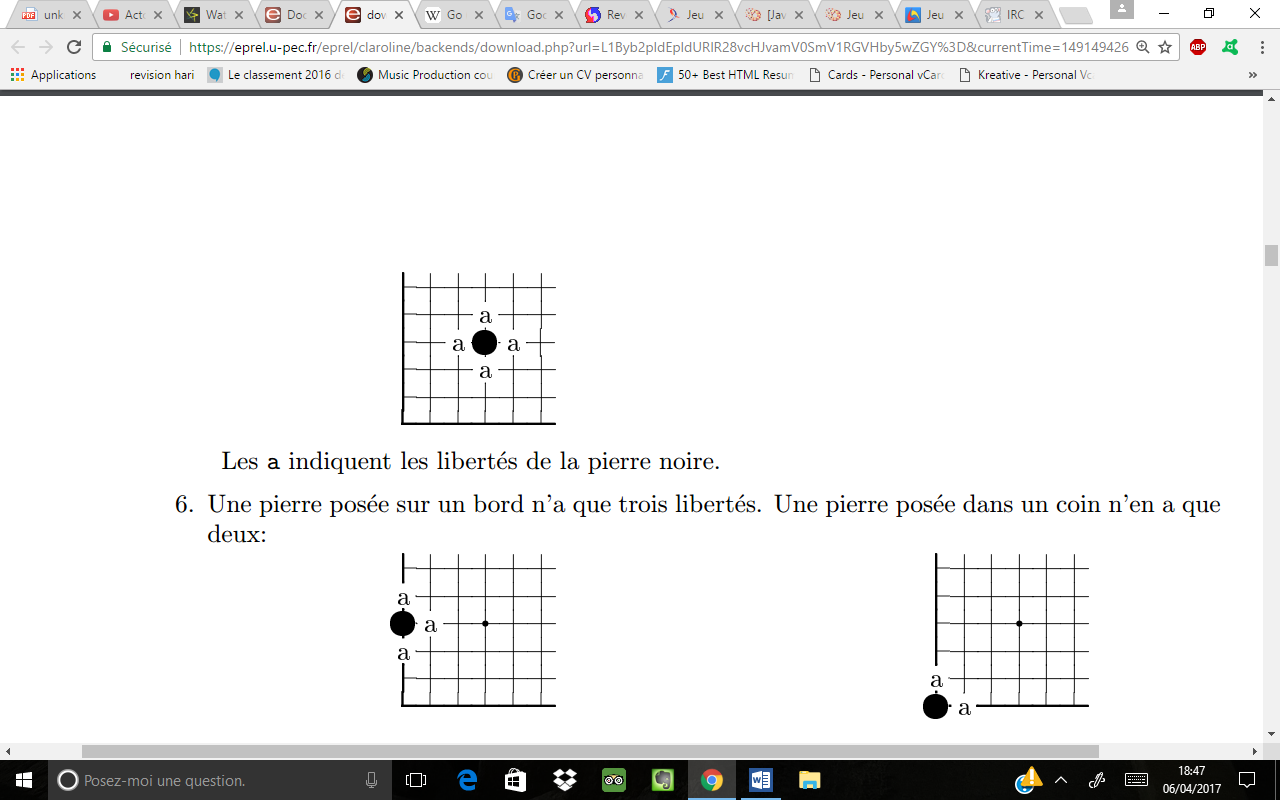
1. The game of go is played with two players. One of them plays with the black stones, the other with the white stones. The standard size of a game board is 19 lines per 19 lines. However, it is also possible to play on 9 lines per 9 lines or 13 lines per 13 lines :

2. The game is played turn by turn, the player with the black stones begins to play.

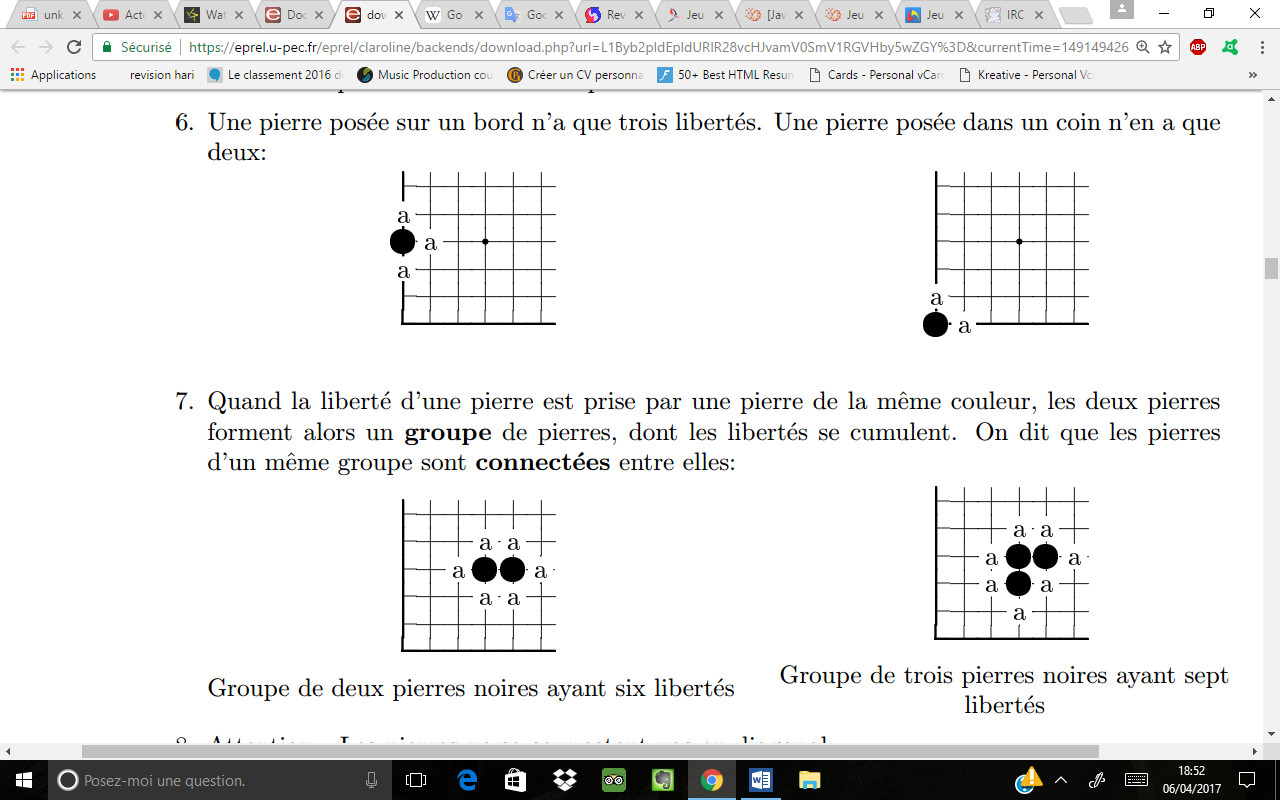
3. On each turn, the player who has the line lays a stone on an intersection of the board:

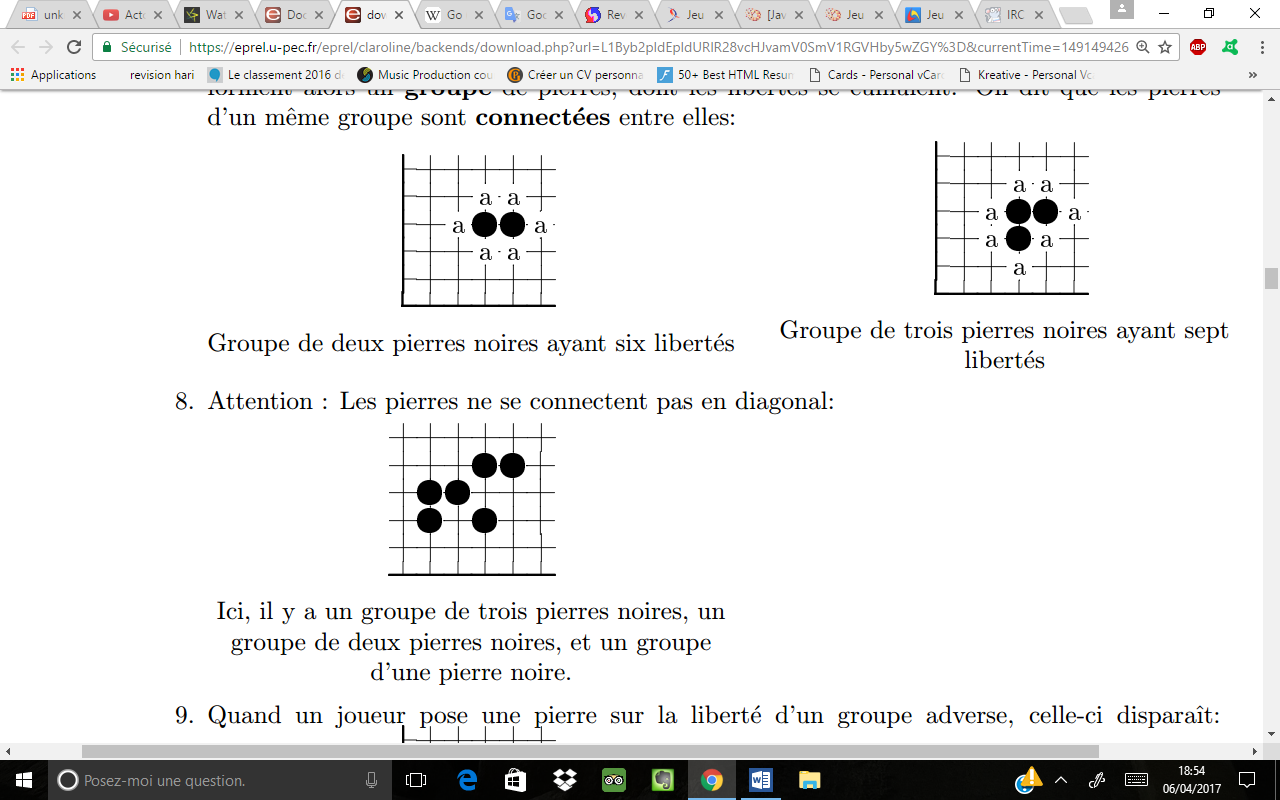
*Here black plays in 1, white answers in 2 and black answers in 3.*

4. Once laid, a stone can not be moved.

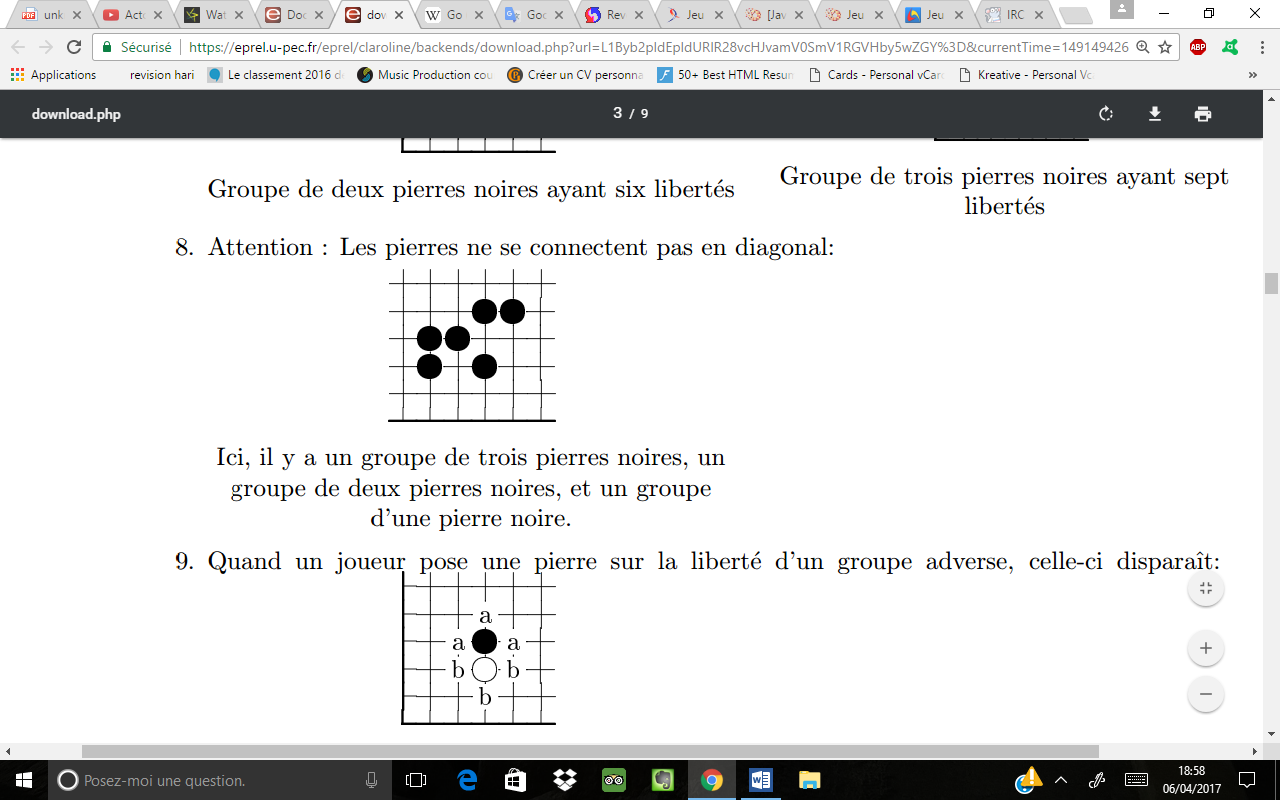
5. A laid stone has a number of freedoms equal to the number of empty intersections at the top, bottom, right and left.

*The a indicate the freedoms of the black stone.*

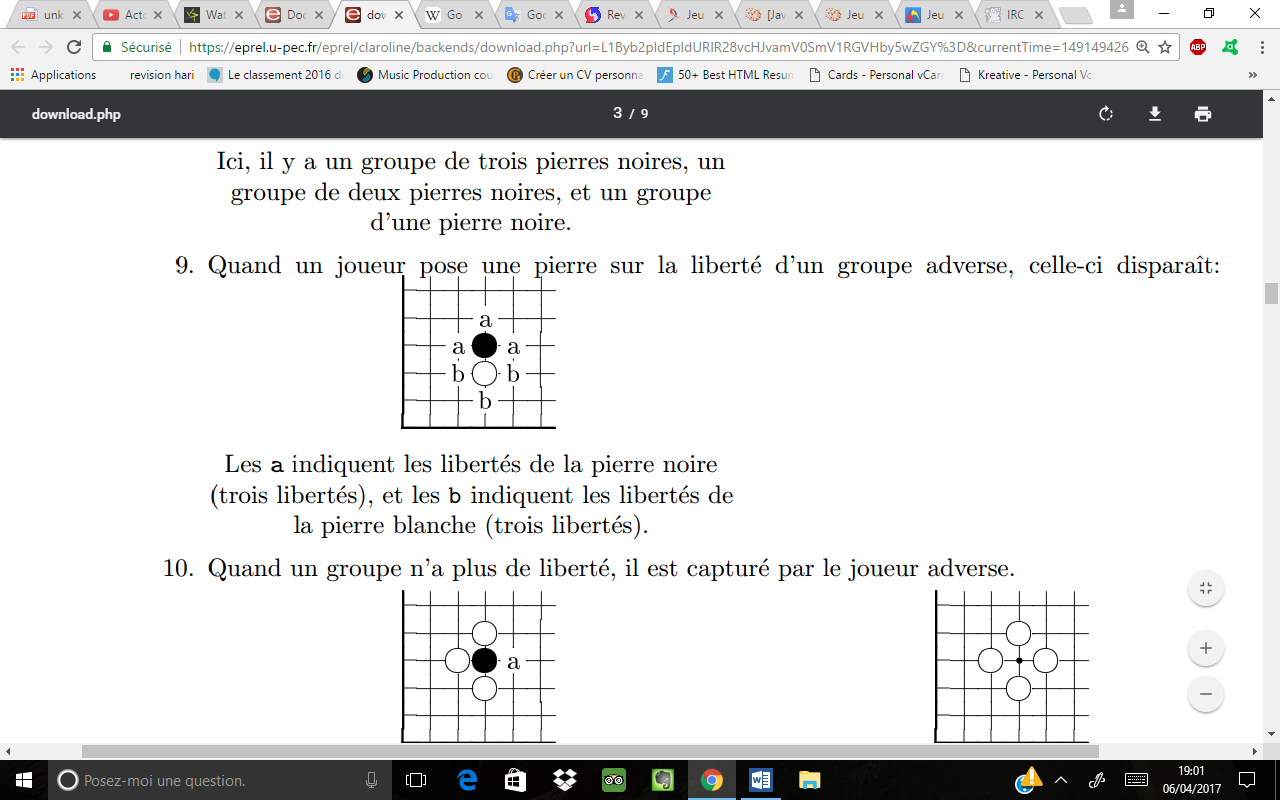
6. A stone placed on an edge has only three freedoms. A stone placed in a corner has only two:

7. When the freedom of a stone is taken by a stone of the same color, then the two stones form a group of stones, the liberties of which are cumulated. It is said that the stones of the same group are connected together:

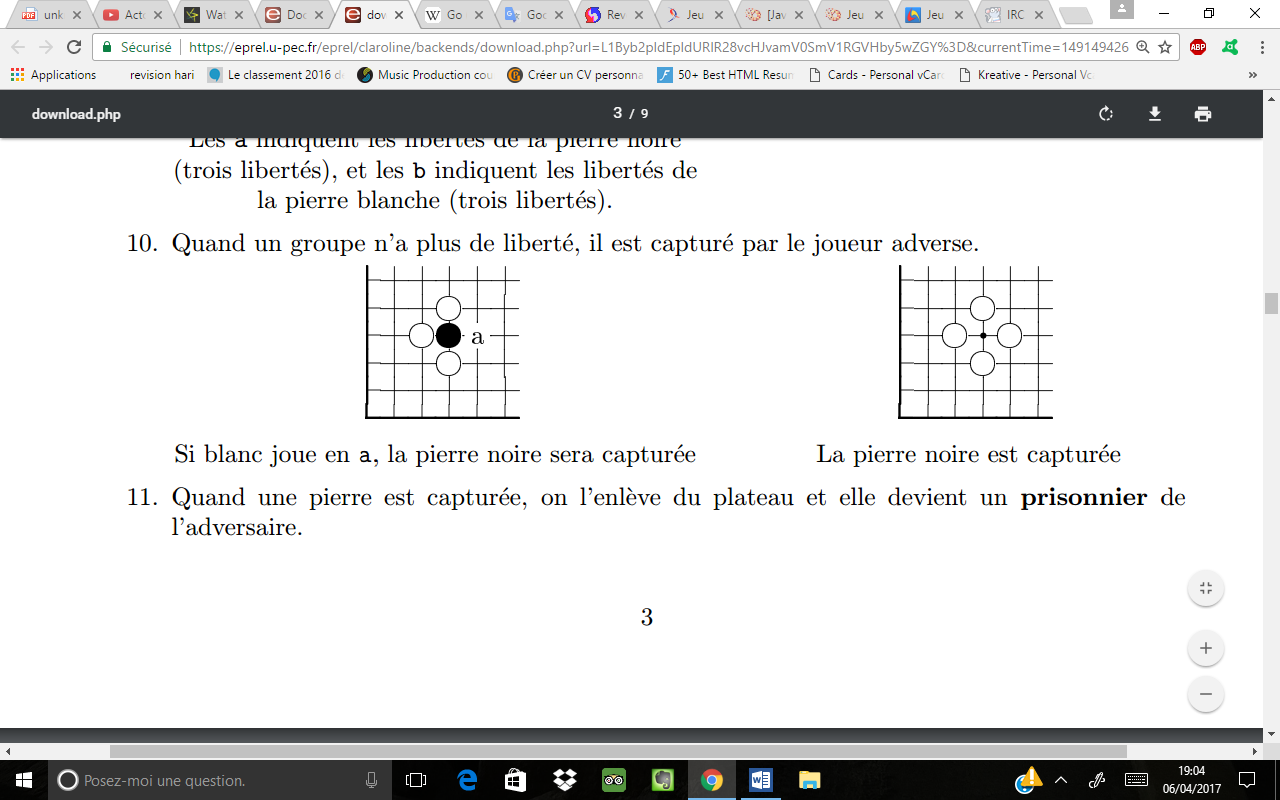
*Group of two black stones with six freedoms Group of three black stones with seven freedoms*

8. Caution: Stones do not connect diagonally:

*Here there is a group of three black stones, a group of two black stones, and a group of a black stone.*

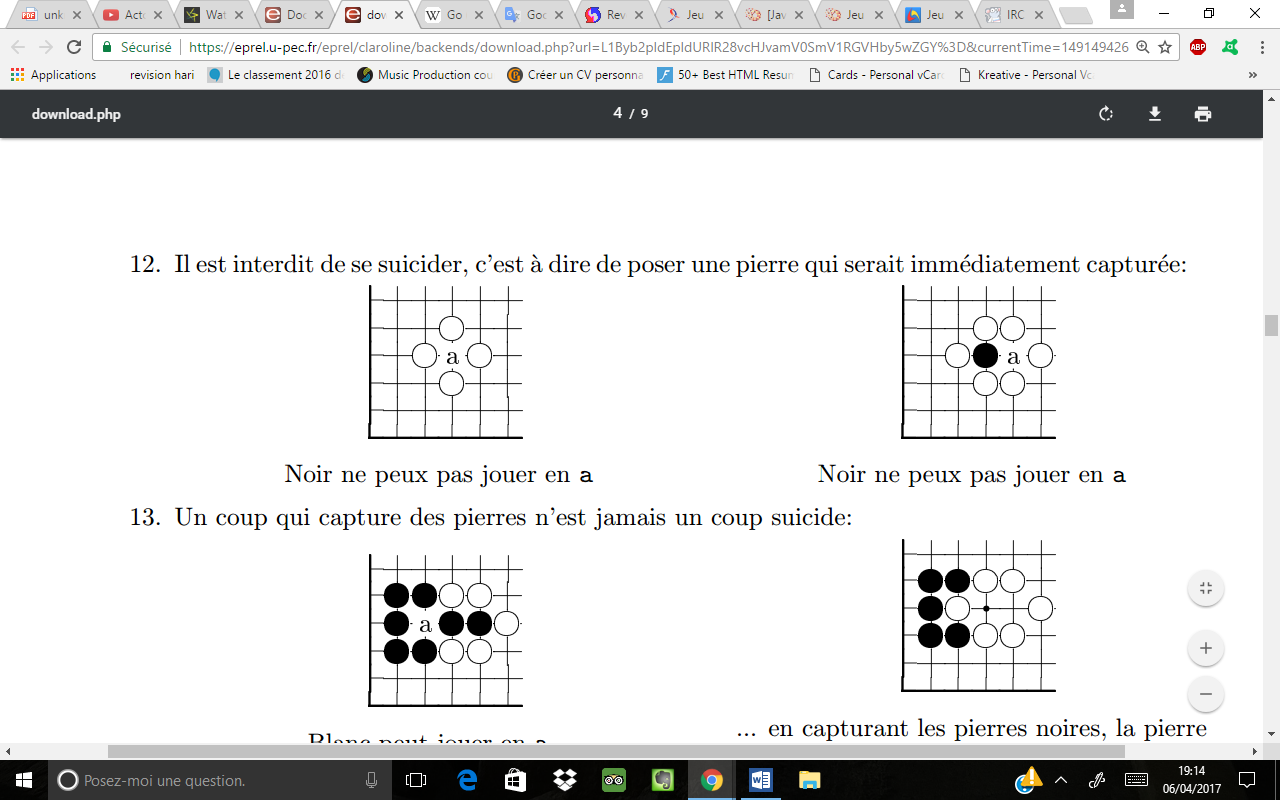
9. When a player puts a stone on the freedom of an opposing group, it disappears:

*The a indicate the freedoms of the black stone (three freedoms), and the b indicate the freedoms of the white stone (three freedoms).*

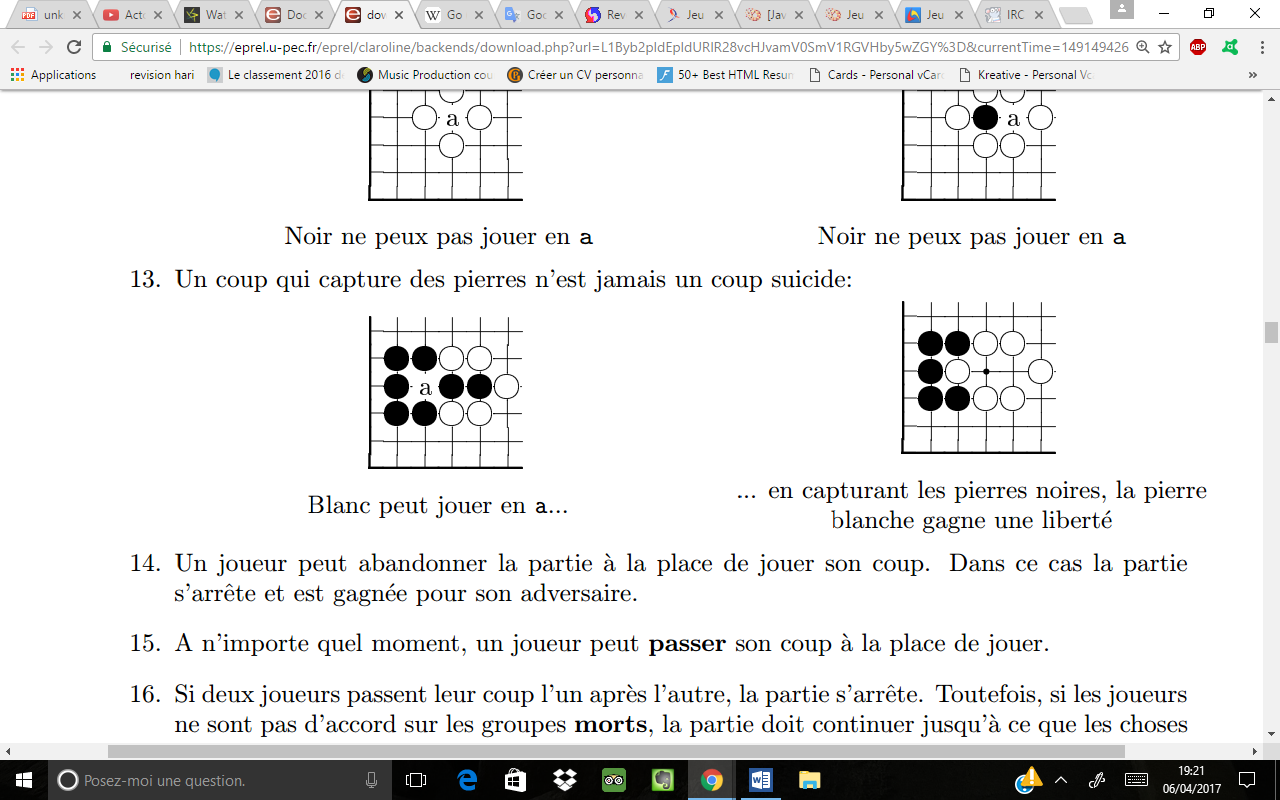
10. When a group has no freedom, it is captured by the opposing player.

*If white plays in a, the black stone will be captured Black stone is captured*

11. When a stone is captured, it is removed from the board and becomes a prisoner of the opponent.

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

*Black can not play in a Black can not play in a*

13. A move that captures stones is never a suicide move:

*White can play in a ... ... by capturing the black stones, the white stone gains a freedom*

14. A player may abandon the game instead of playing his turn. In this case the game stops and is won for its opponent.

15. At any time, a player may pass his turn instead of playing.

16. If two players pass their turns one after the other, the game ends. However, if the players do not agree on the dead groups, the game must continue until things are clarified. We will return to this point in the section on the subtleties of the game.

**The score**

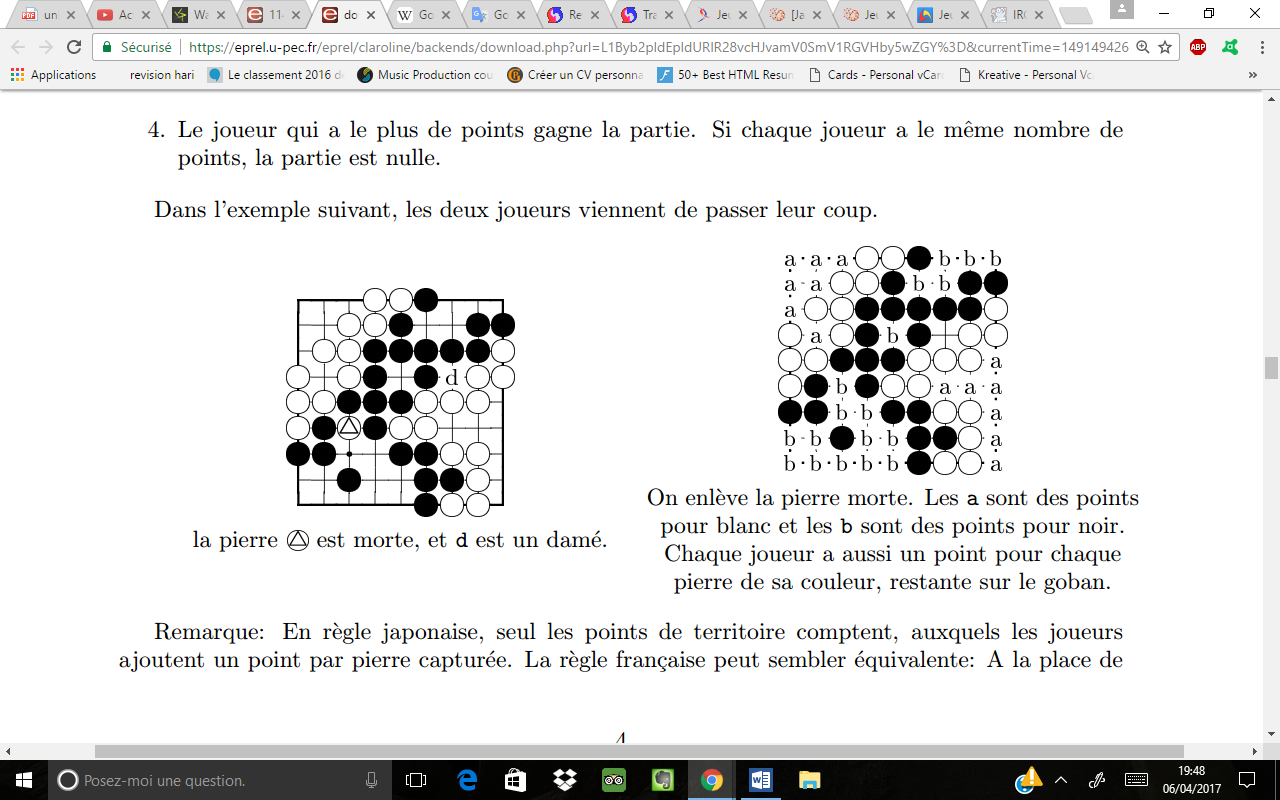
Once the part is completed otherwise than by abandonment, the points are counted as follows:

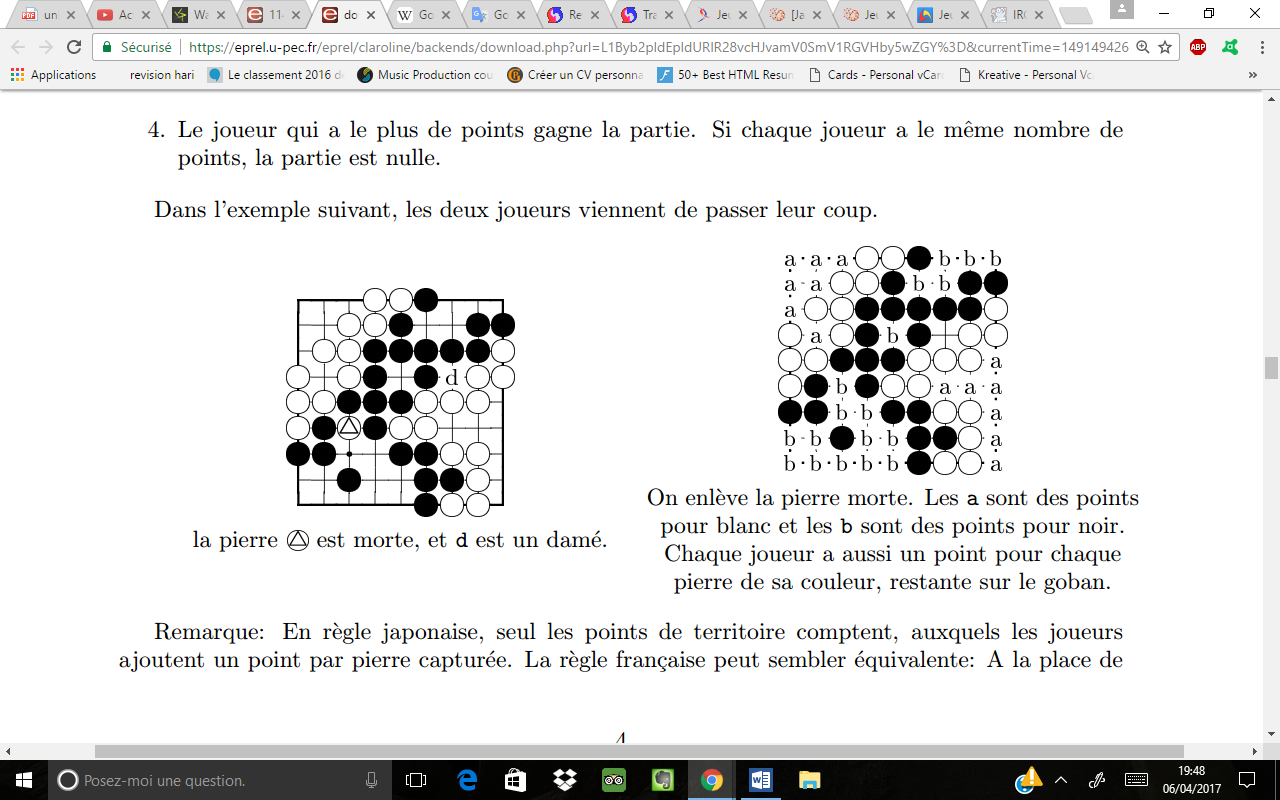
1. Players begin by removing dead stones by taking them as prisoners.

We will come back to this important point at the end of the game.

2. Each stone on the goban brings a point to the corresponding player.

3. Each intersection of the goban surrounded by stones of the same color yields a point to the corresponding player. These intersections are said to be part of the player's territory. The intersections that are not surrounded by stones of the same color are called **« damés » (crowned)**.

4. The player with the most points wins the game. If each player has the same number of points, the game is a draw.

In the following example, both players have just passed their move

*The dead stone is removed. The a are points for white and the b are points for black. Each player also has a point for each stone of its color, remaining on the goban.*

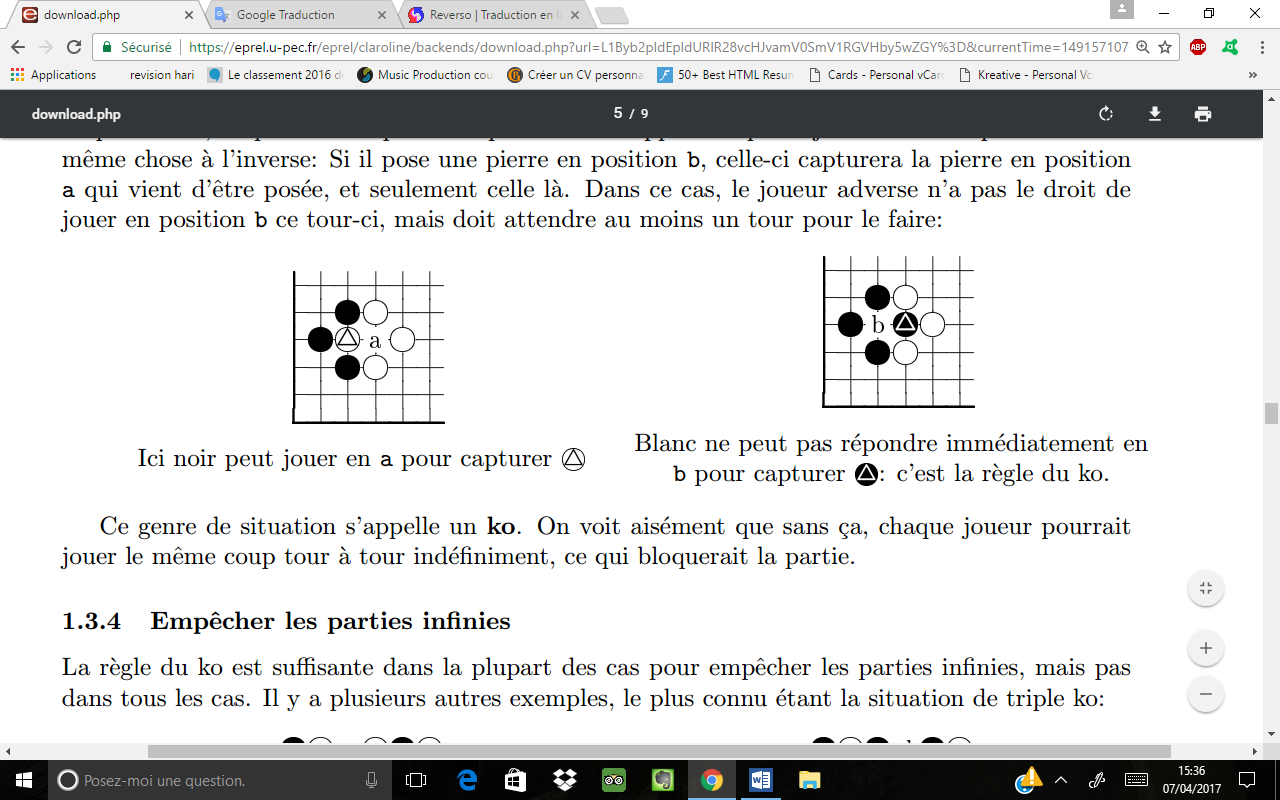
*The stone is dead, and d is a crowned*

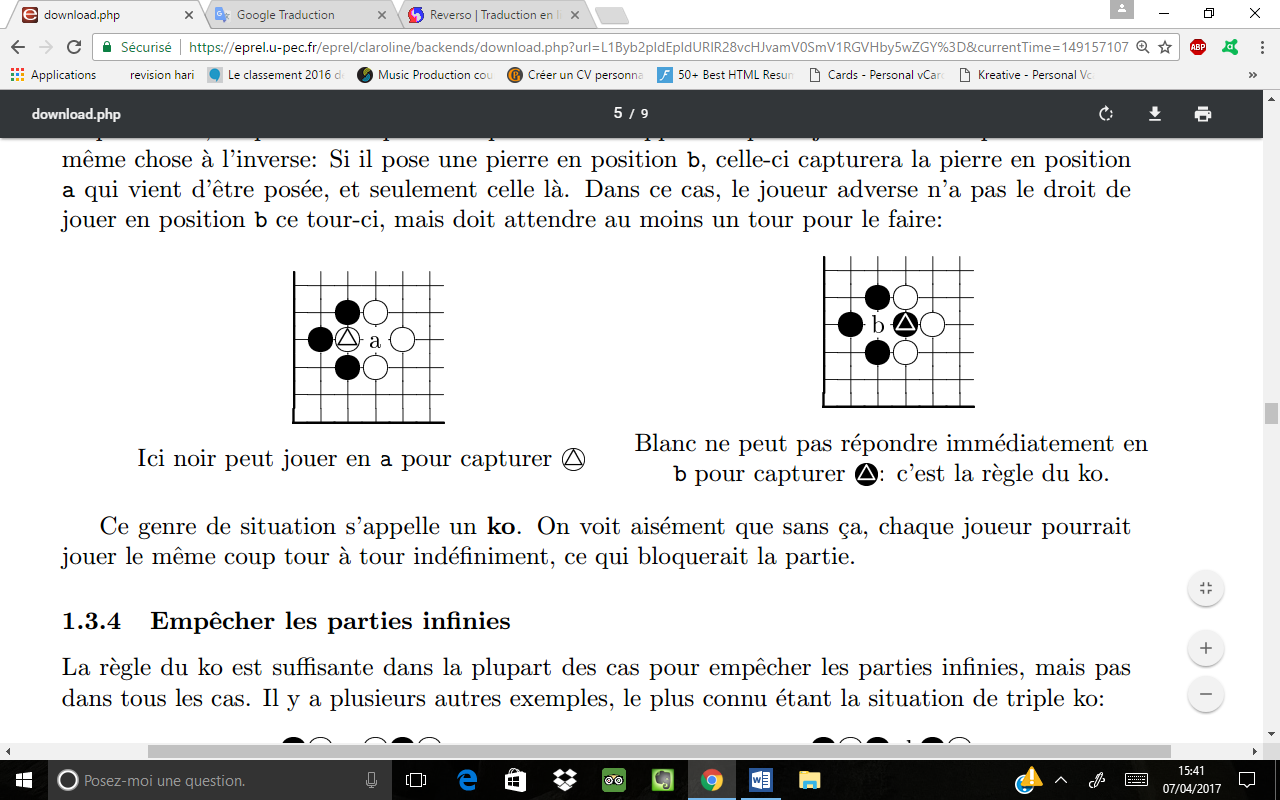
Note: In Japanese rule, only territory points count, to which players add one point per captured stone (one stone captured = one point added). The French rule may seem equivalent: Instead of counting the prisoners like point to add for us, they are count as equally less points when the opponent counts his remaining stones on the game board. In practice, there are subtle differences between these two ways of counting: the French rule is more stronger in case of dispute. We will return to this in the section on the subtleties of the game.

**1.3.2 The komi**

The player who begins has an advantage on the one who does not begin. To balance this, a number of points are given to the player who does not start the game (the white player). This number, called komi is currently 7.5 in the French rules (which prevents in particular the draws).

**1.3.3 The ko**

This rule is intended to prevent the possibility of making infinite parts. Suppose that we are in a situation where a player has the possibility to capture exactly one opposing stone in position b, putting a stone in position a. Let us suppose that the opposing player can do the same thing in reverse: If he puts a stone in position b, it will capture the stone in position a just placed, and only that one. In this case, the opposing player does not have the right to play in position b this turn, but must wait at least one turn to do it:

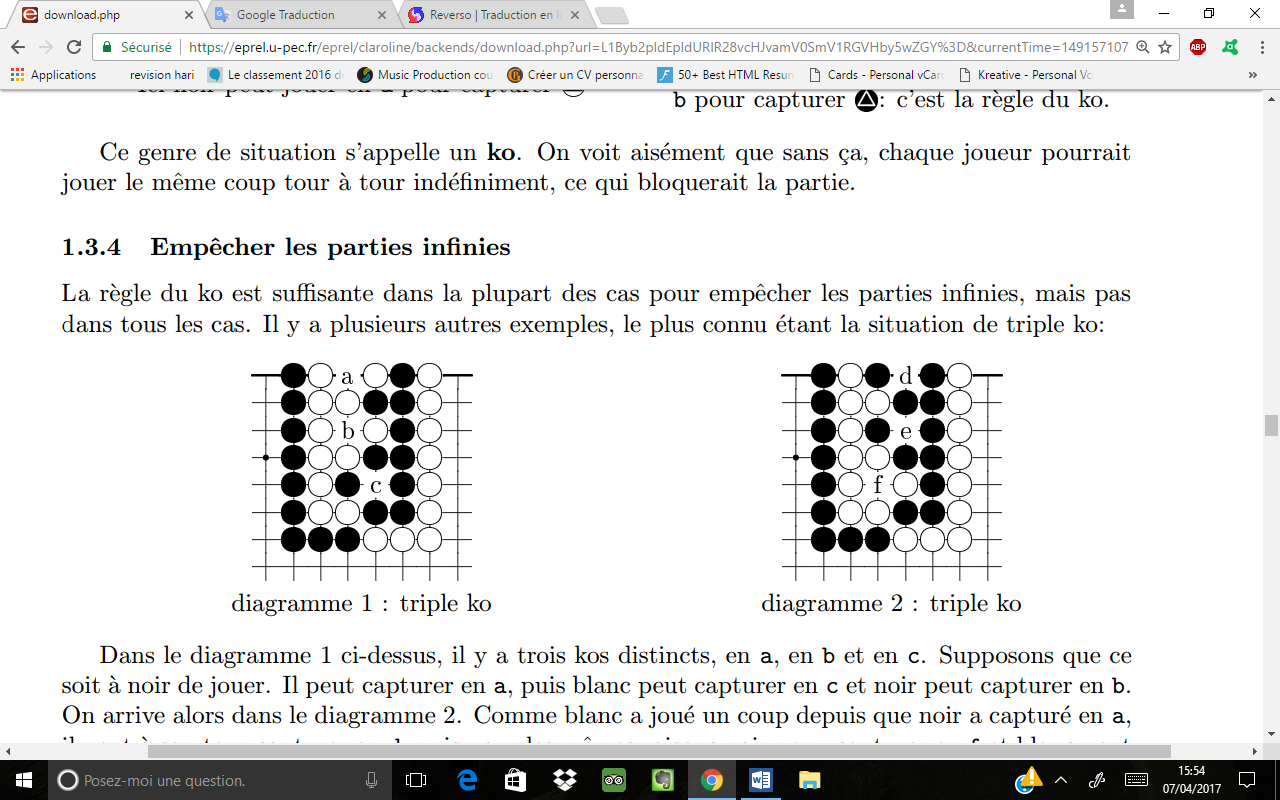


*White can not answer immediately in b to capture it is the rule of the ko.*

*Here black can play in a to capture *

This kind of situation is called a ko. It is easy to see that without this, each player could play the same move in turn indefinitely, which would block the game.

**1.3.4 Prevent Infinite Parts**

The rule of the ko is sufficient in most of the cases to prevent infinite parts, but not in all cases. There are several other examples, the best known being the situation of triple ko:

*Diagram 1: triple ko Diagram 2: triple ko*

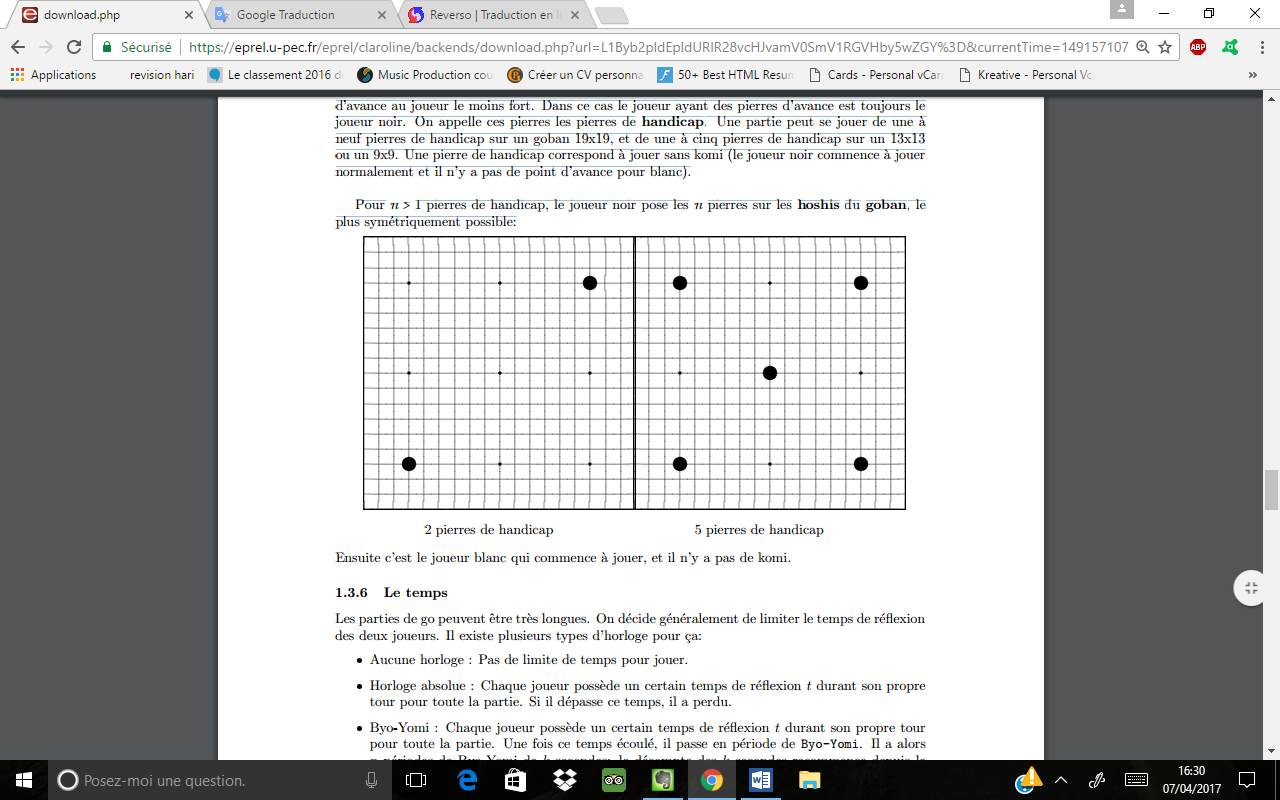
In diagram 1 above, there are three distinct kos, in a, b and c. Suppose it is black to play. It can capture in a, then white can capture in c and black can capture in b. Then we arrive in diagram 2. As white has made a move since black has captured in a, it can in his turn capture in d then for the same reasons black can capture in f and white can capture in e. We then return to diagram 1, and we can start playing this way indefinitely again.

To prevent definitively all the infinite parts, we introduce the following rule: if by making a move, the game board arrives in a configuration already met for the third time, then this move is forbidden.

Note: In Japanese rule, if the same configuration happens three times in succession, the game is a draw. In practice, this kind of situation is extremely rare.

**1.3.5 The handicap**

When there is a difference in levels between two players, it is possible to give stones in advance to the least strong player. In this case the player with stones ahead is always the black player. These stones are called stones of handicap. A game can be played from one to nine handicap stones on a 19x19 goban, and from one to five handicap stones on a 13x13 or a 9x9. A handicap stone corresponds to play without komi (the black player starts to play normally and there is no point of advance for white).

For n > 1 handicap stones, the black player places the n stones on the hoshis of the goban, as symmetrically as possible:

*2 stones of handicap 5 stones of handicap*

Then the white player starts to play, and there is no komi.

**1.3.6 Time**

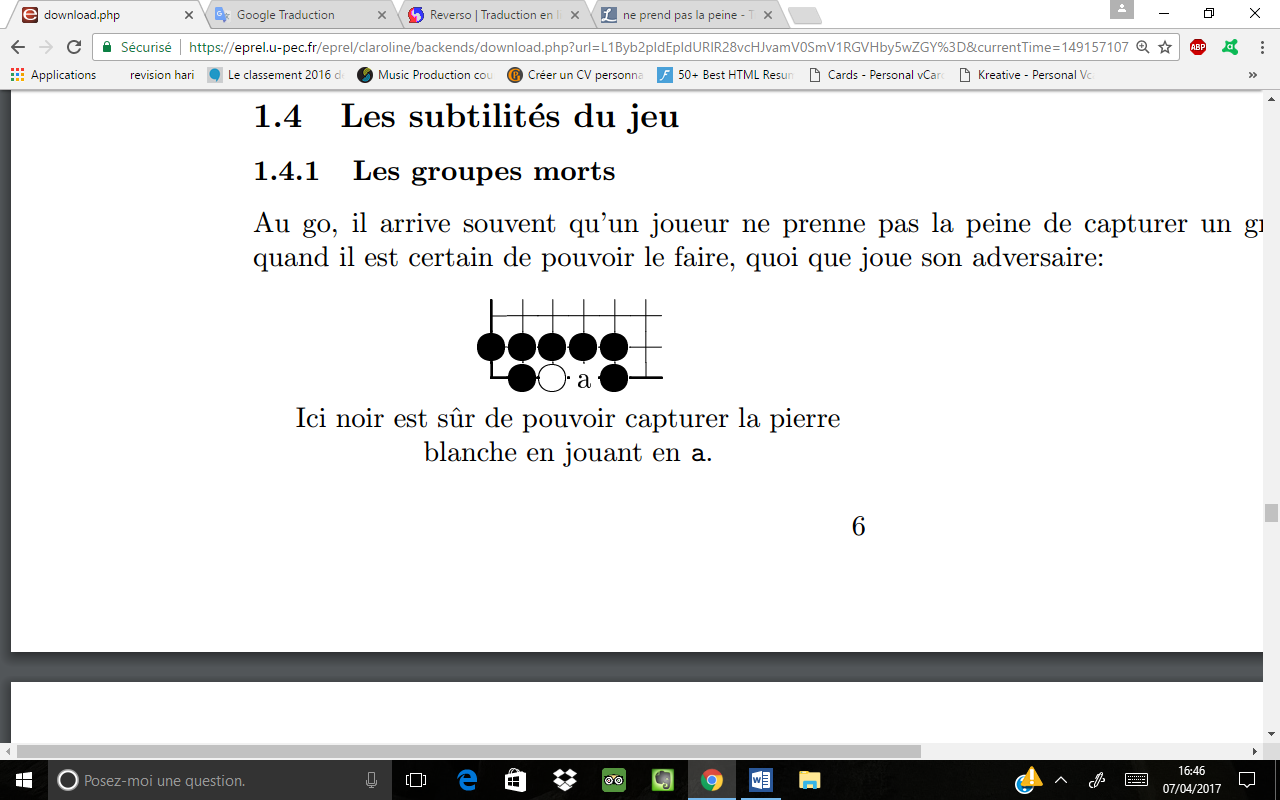
The go parts can be very long. It is generally decided to limit the time of reflection of the two players. There are several types of clock for this:

* No clock: No time limit for playing.
* Absolute Clock: Each player has a certain time of reflection *t* during their own turn for the whole game. If it exceeds that time, it has lost.
* Byo-Yomi: Each player has a certain time of reflection *t* during his own turn for the whole game. Once this time has passed, he passes through the Byo-Yomi period. It then has *n* Byo-Yomi periods of *k* seconds: the *k* seconds count starts from the beginning of each turn as long as the player plays in less than *k* seconds. If the player exceeds *k* seconds during his reflection time, a Byo-Yomi period is removed (if it exceeds 2*k* seconds, 2 are removed, etc.). If the player reaches zero Byo-Yomi period, he has lost the game.

**1.4 The subtleties of the game**

**1.4.1 The dead groups**

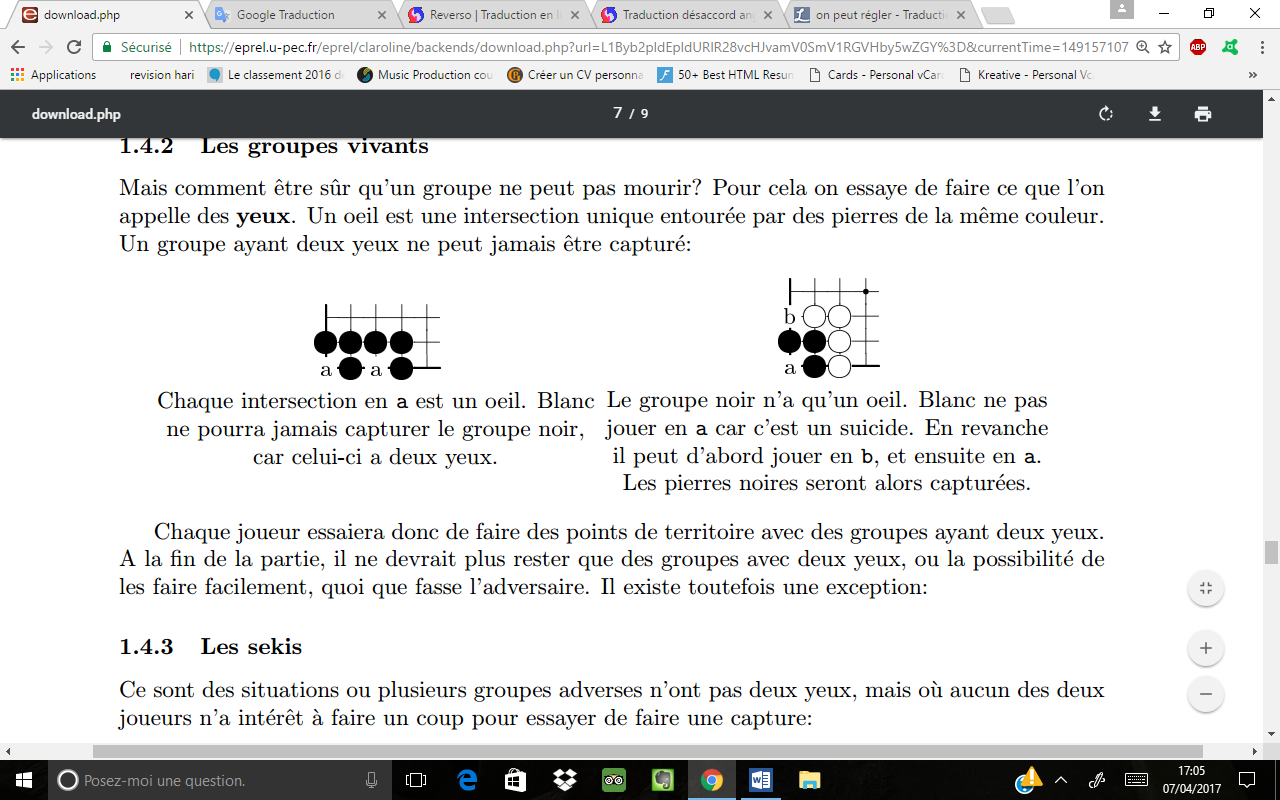
In the go, it often happens that a player does not take the effort to capture a group of stones when he is certain to be able to do it, whatever his opponent plays:

*Here black is sure to be able to capture the white stone by playing in a.*

In the example above, black will not normally play in a. Thus, at the end of the game, the white stone will be removed which will be a prisoner of black, and will count a point: for the final score, during the counting of the white stones remaining on the goban, white will have one less. The black group will then report three points of territory, one for each empty intersection, which makes a total of four points. Groups of stones that we are sure will be captured no matter what happens are dead groups.

The above case is simple to determine, but there are more complicated cases. In particular, if the two players do not agree on the status of a group: dead or alive, it is the player who thinks the group has died to prove it, by capturing it. We see here the advantage of the French rule on the Japanese rule: always in the example above, if black plays in a to capture the white stone, it will have only three points of territory, because a previously empty intersection Will be taken by one of its stones. He would have a point less in Japanese rule, but not in French rule, because the stone he just put gives him a point: We can thus simply resolve the cases of dispute.

**1.4.2 Living groups**

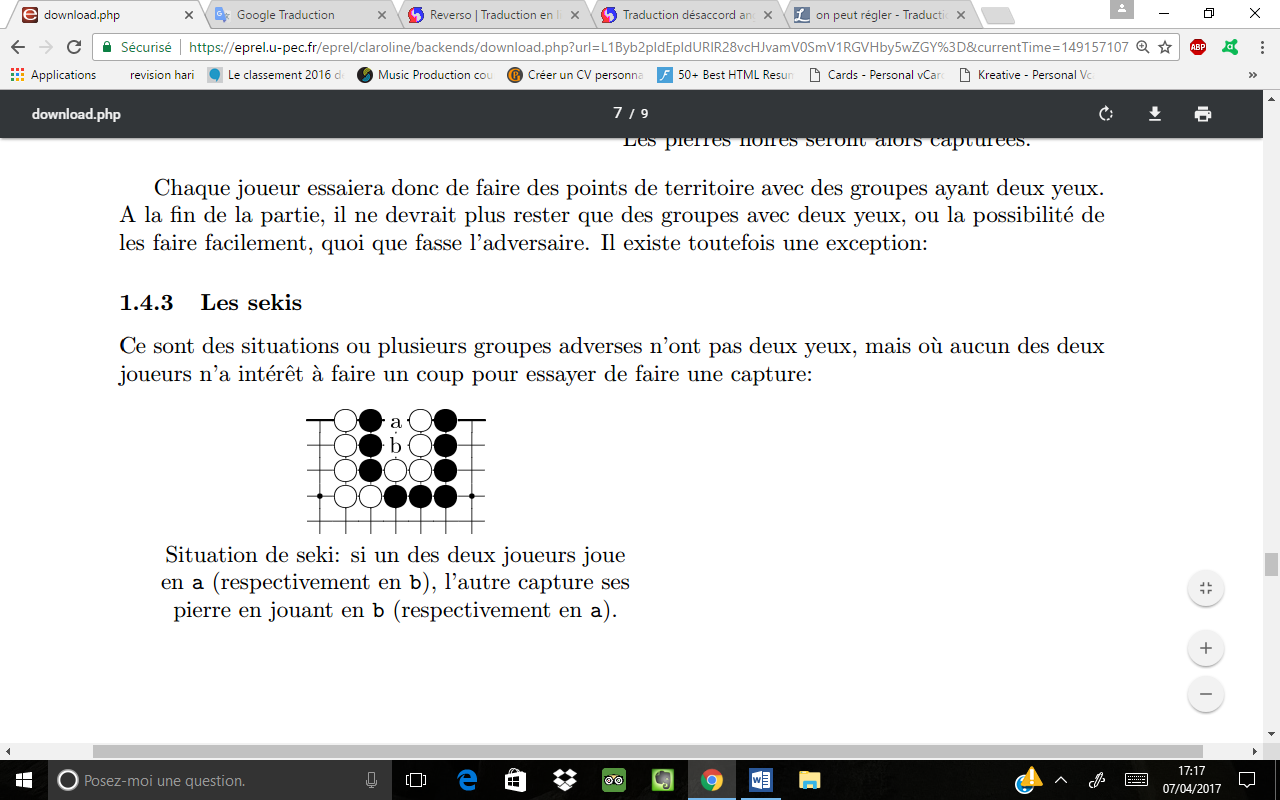
But how be sure that a group can not die? For that we try to do what we call the eyes. An eye is a unique intersection surrounded by stones of the same color. A group with two eyes can never be captured:

*The black group has only one eye. White could not play in a because it is a suicide. On the other hand, he can first play in b, and then in a. The black stones will then be captured.*

*Each intersection in a is an eye. White can never capture the black group because it has two eyes.*

Each player will try to make points of territory with groups having two eyes. At the end of the game, there should only remain groups with two eyes, or the ability to do them easily, whatever the opponent does. There is, however, one exception:

**1.4.3 The sekis**

They are situations where several opposite groups do not have two eyes, but where none of the two players should rather make a move to try to make a capture:

*Situation of seki: if one of the two players plays at a (respectively in b), the other captures its stone by playing in b (respectively in a).*

**2 The project**

You have to make a java program, allowing two players to do go games. It is forbidden to use packages other than standard Java packages (javax.swing, java.util, etc ...) except when explicitly specified in the following instructions.

The CGoban software, available free on the Internet, can give you a good idea of what your project might look like (the software is programmed in Java and must start with "Java Web Start").

**2.1 Rendering mode**

Your submission must consist of a .tar file containing your source files. You should also include an AUTHORS file (with your first and last name) and a README file that briefly explains the contents of each of your source files, as well as instructions for starting the program (in which file is the entry point , Etc ...), and in case you think it necessary, instructions explaining how to use your program.

**2.2 The internal representation**

You must create a structure representing your go game, with methods to make moves.

1. A move must automatically capture the stones of the goban if necessary.

2. A forbidden move must not be played (you must also handle the rule of the ko and the rule that prevents infinite parts).

3. Once a game is completed by two players who have taken turns, you must have a function that tells you the winner. We do not ask to automatically detect dead groups: the function must take this information as a parameter.

**2.3 History**

You must store the history of all played moves. You must make methods to navigate the history as desired: Allow the possibility to cancel as many moves as you want, but also "establish" them. On the other hand, if you cancel a move and play something else, it erases the move in the history that you canceled.

**2.4 The** [**graphical interface**](http://www.linguee.fr/anglais-francais/traduction/graphical+interface.html)

You have to make a [graphical interface](http://www.linguee.fr/anglais-francais/traduction/graphical+interface.html) for your go game. In particular, you must have:

1. A menu allowing you to choose the game parameters and start a game.

2. The representation of the game, with the goban, information about the two players (time remaining, number of prisoners, etc ...)

3. the ability to navigate through the history of the moves.

4. the ability to resize your game window.

**2.5 The Game**

All this together must allow two players to play a game according to the French rules set out above. You must also manage the choice of handicap stones, the size of the goban, and the type of clock. Once a game is over, you must automatically allow players to select dead groups, and your program must count down the points.

**3 Possible bonuses**

**3.1 The score**

Add the ability to estimate the score of a current game. Also add the ability to automatically remove dead groups when a game is finished.

**3.2 The SGF files**

You can manage the reading and the saving of files in sgf format. You can use an already existing library for this if you wish.

**3.3 The Go text protocol**

You must allow one of the two players (or both players) in your program to be an AI. To do this there is no need to do AI: your program must be able to interface with an existing AI using the Go text protocol. You can use an already existing library for this if you wish. In addition, you can test various open source AI, such as Gnu Go or Fuego.

**3.4 AI**

Make an Artificial Intelligence (AI) for your game of go ...

**4 Ratings**

There will be a presentation for your project. The evaluation will be based on the following criteria:

1. Respect the rules of the game (all rules).

2. The graphical interface (game and menu).

3. The good management of the parties (manage the handicap, the time, the calculation of the score, etc ...).

4. Proper history management.

5. The quality of the code (modularity, cleanliness, etc ...), as well as the clarity of rendering (no unnecessary files, README file well explained, etc ...)

6. Finishing (Is your program easy and pleasant to use).

7. The quality of the presentation.