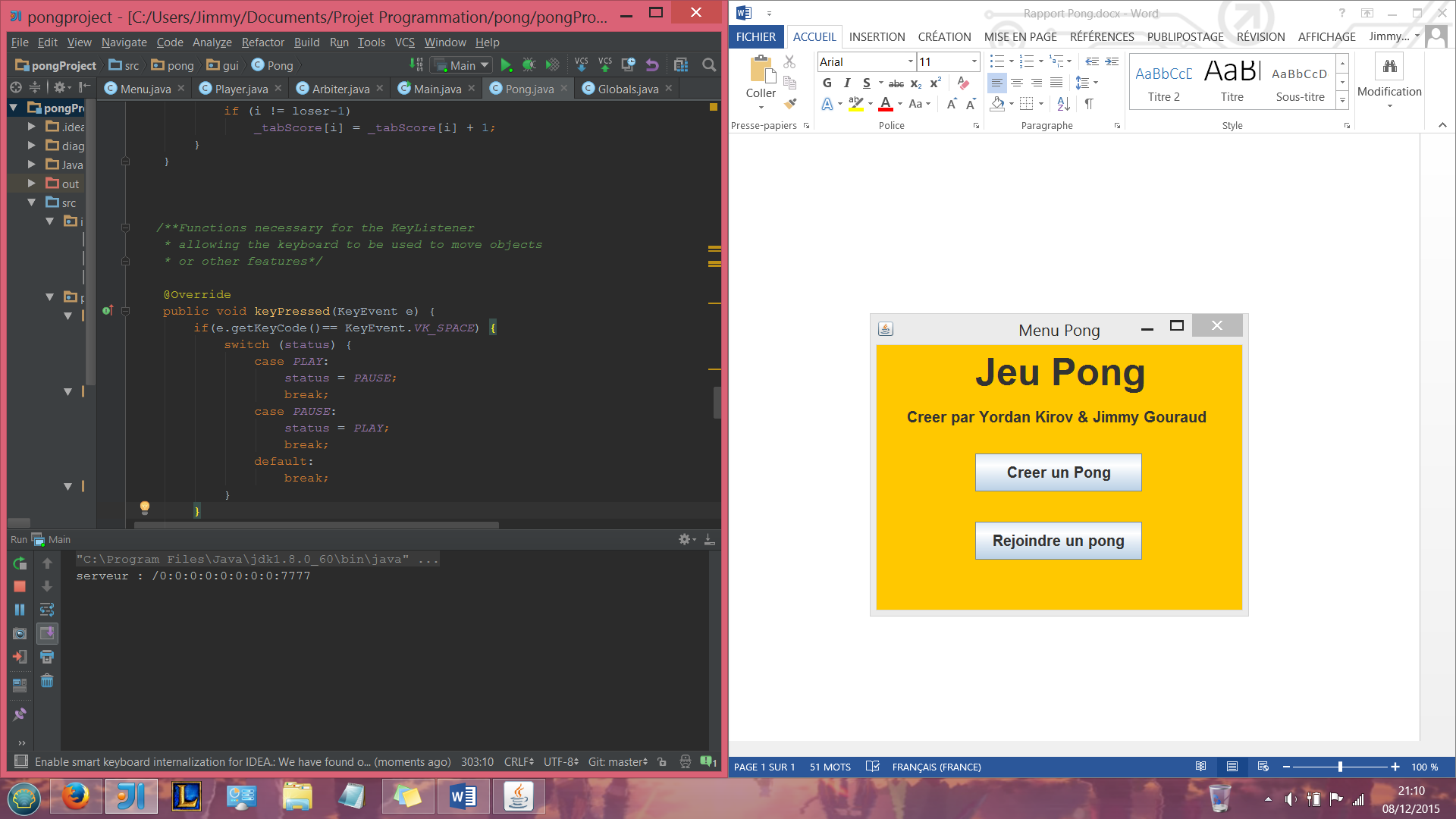
Rapport sur le Pong en réseau

Jimmy GOURAUD & Yordan KIROV

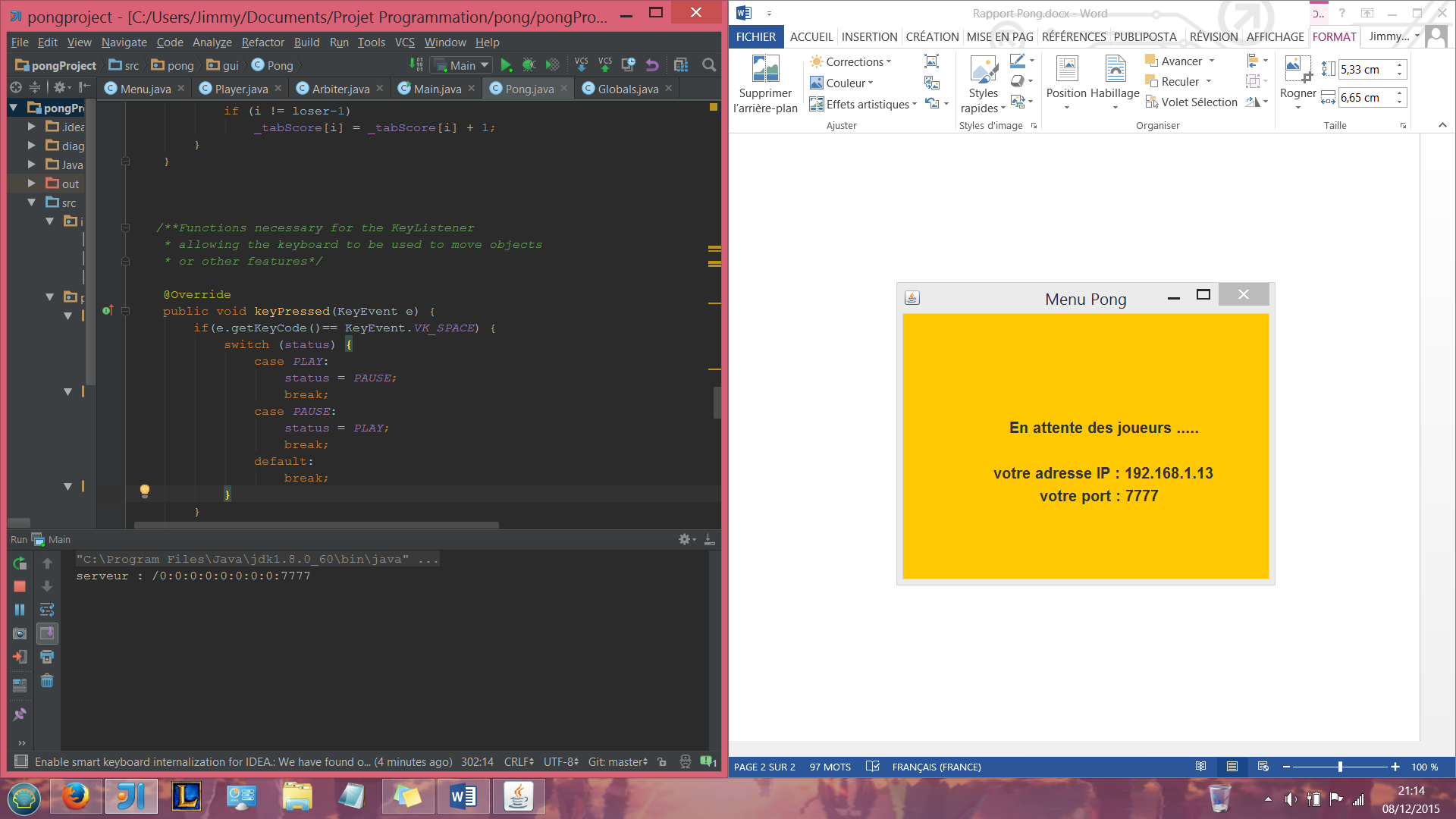
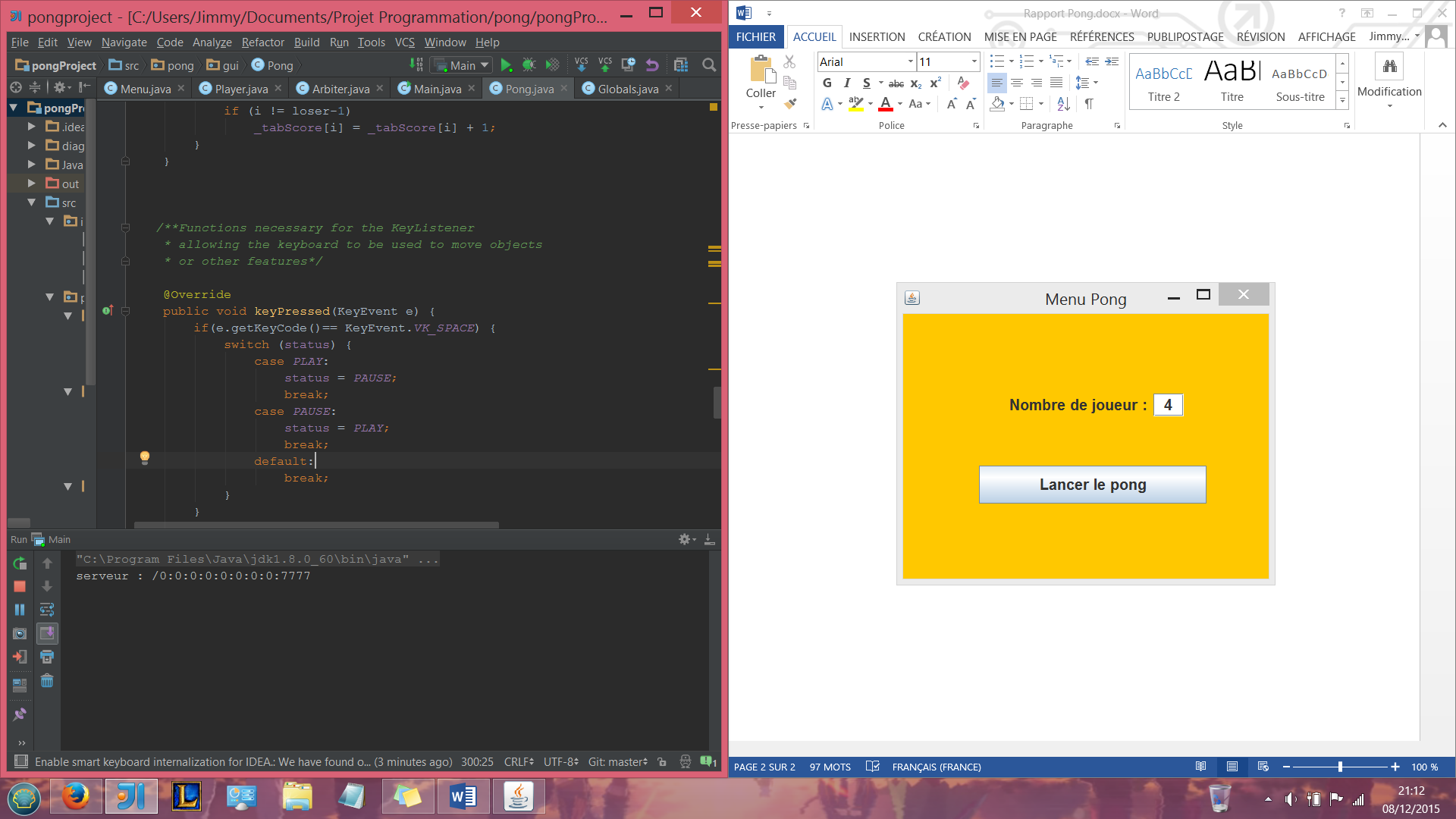
# I. Présentation des fonctionnalités

## 1. Le Menu

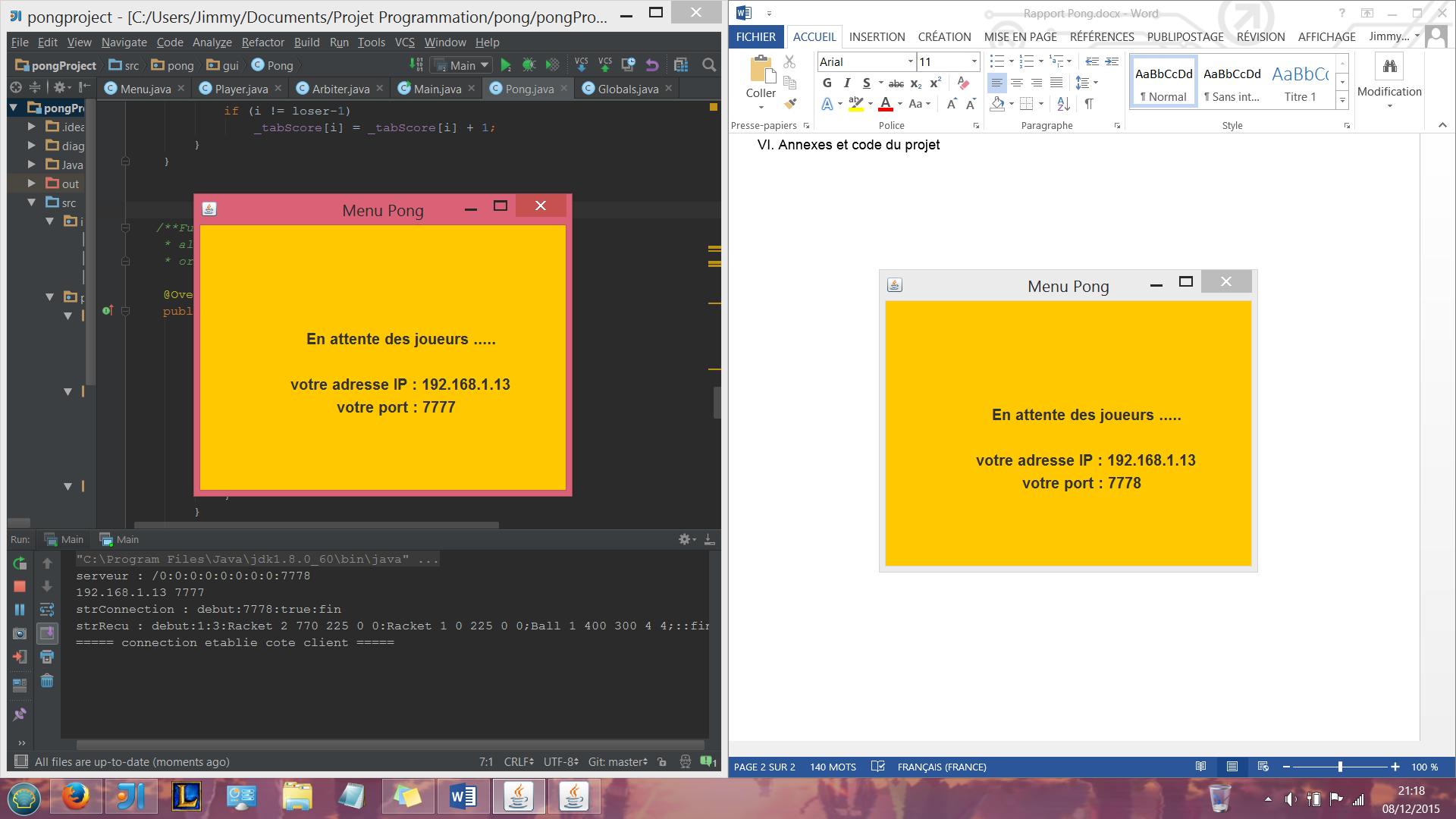
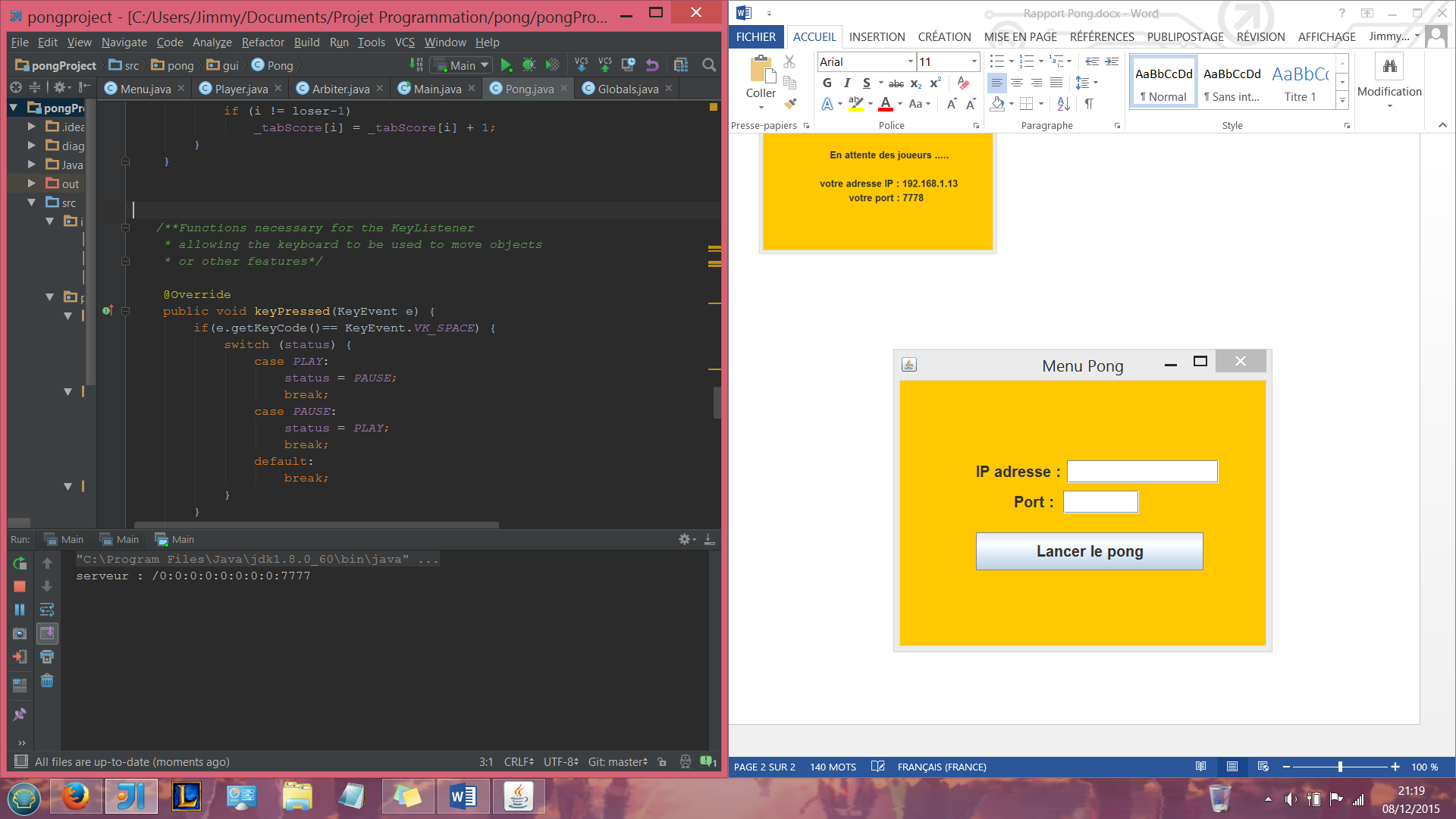
Nous avons un Pong en réseau qui peut accueillir jusqu’à 4 joueurs différents. Lors du lancement du Pong, une première interface nous invite à choisir entre la création du Pong (si nous sommes le 1er joueur) ou bien à rejoindre un Pong existant :



Si l’on choisit de créer un Pong, nous devons saisir le nombre de joueur et ensuite une fenêtre « d’attente » apparait :



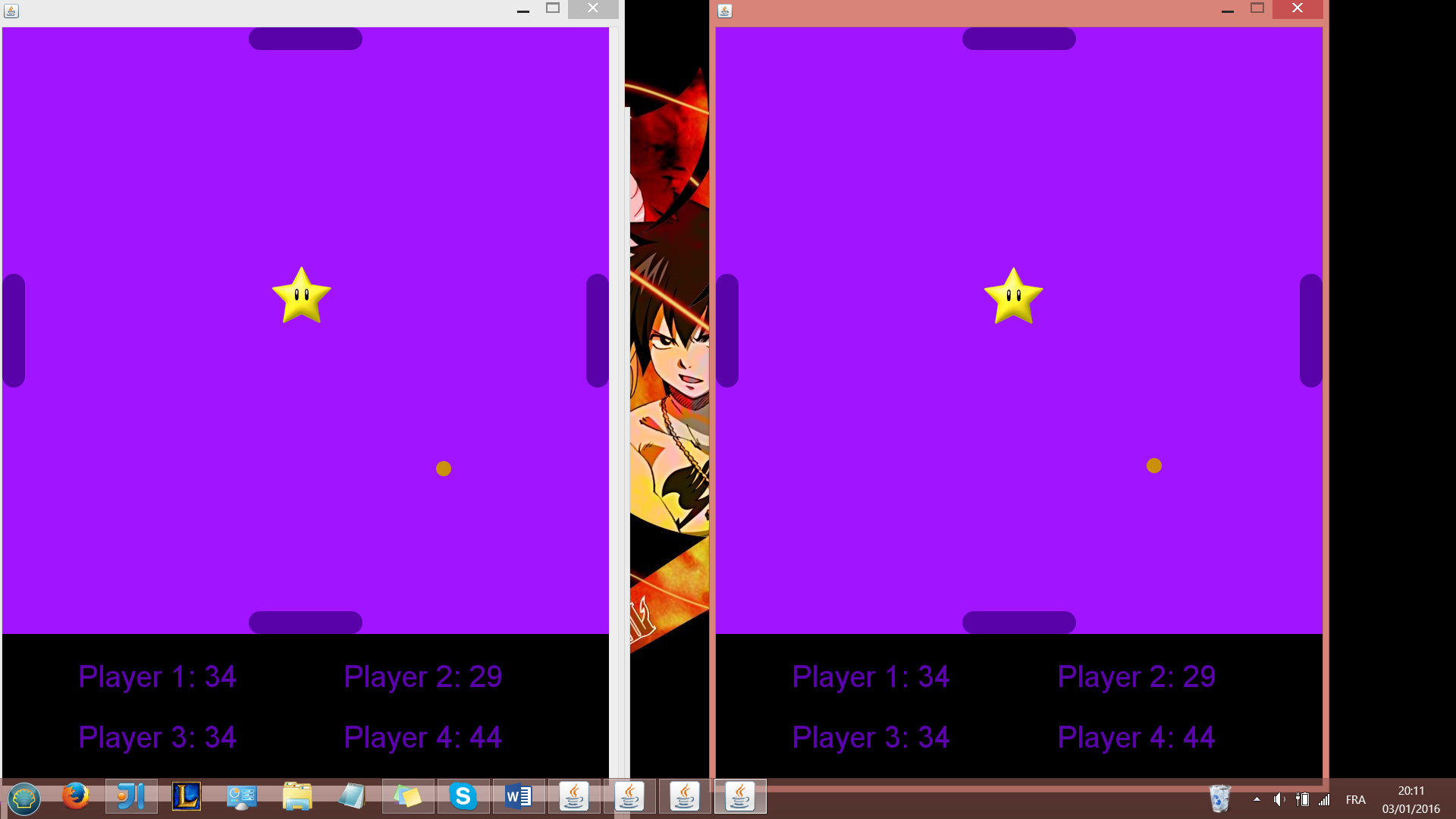
Si par contre on choisit de rejoindre un Pong existant, une fenêtre demandant l’adresse IP, ainsi que le port, du Pong sur lequel on veut se connecter, apparait, entrainant ensuite une fenêtre « d’attente » si le nombre de joueur n’est pas atteint.



Lorsque le nombre de joueur est atteint, le jeu se lance.

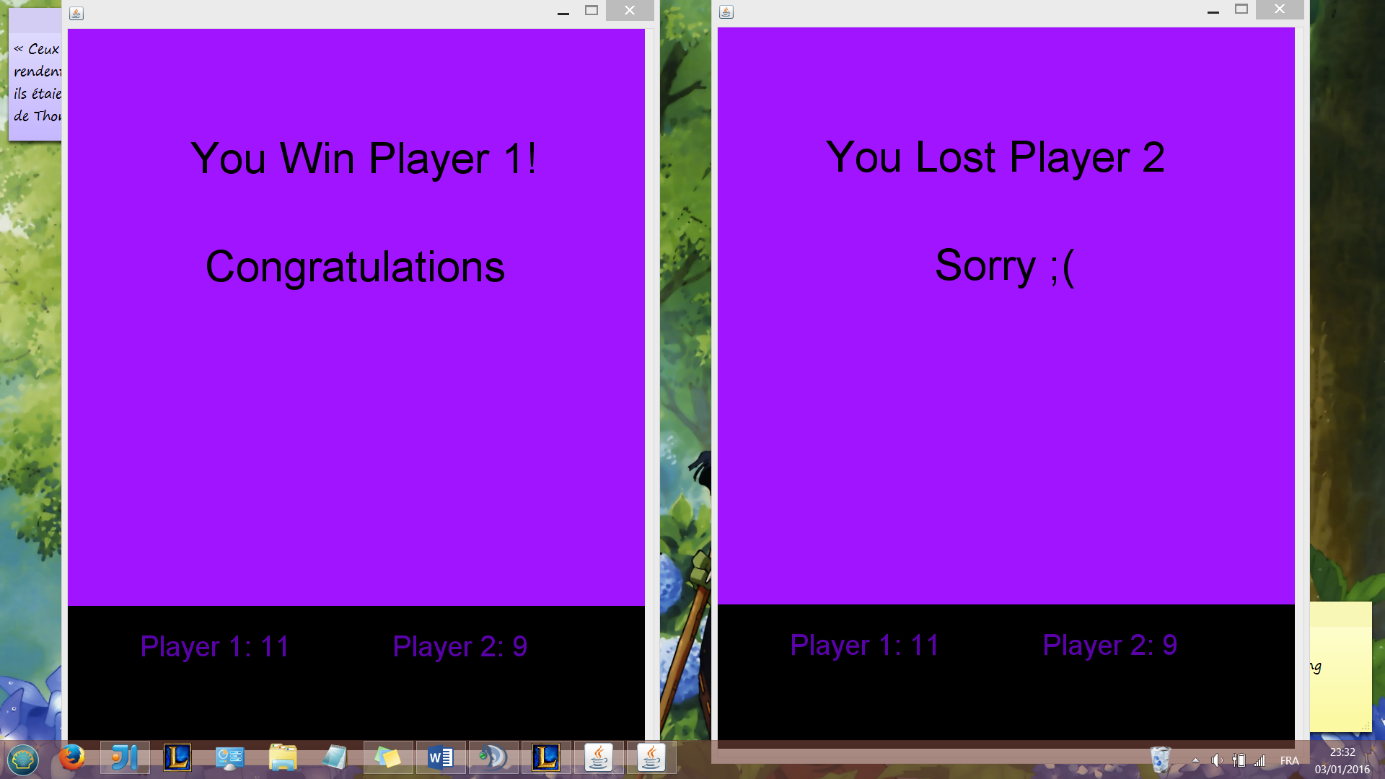
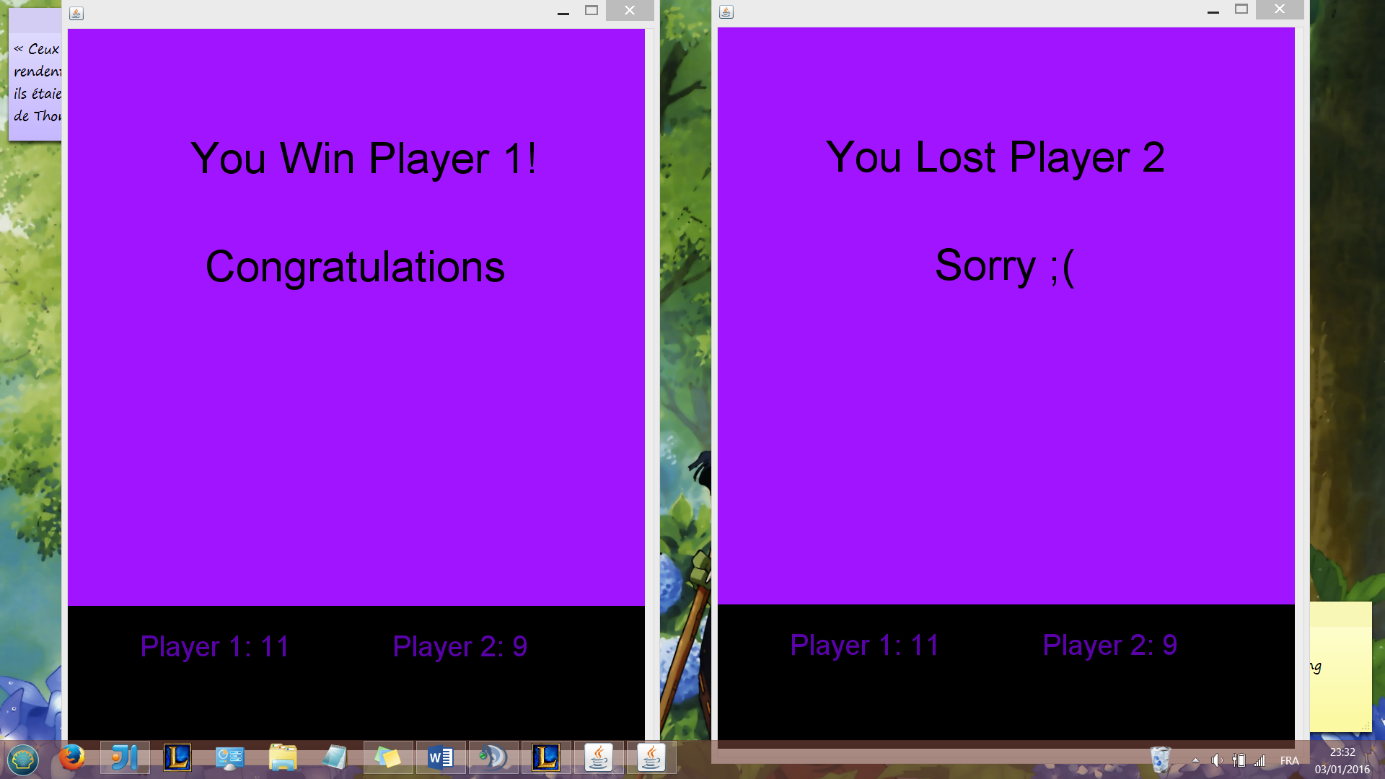
## 2. Le jeu

Voici le rendu final du jeu (4 joueurs) :



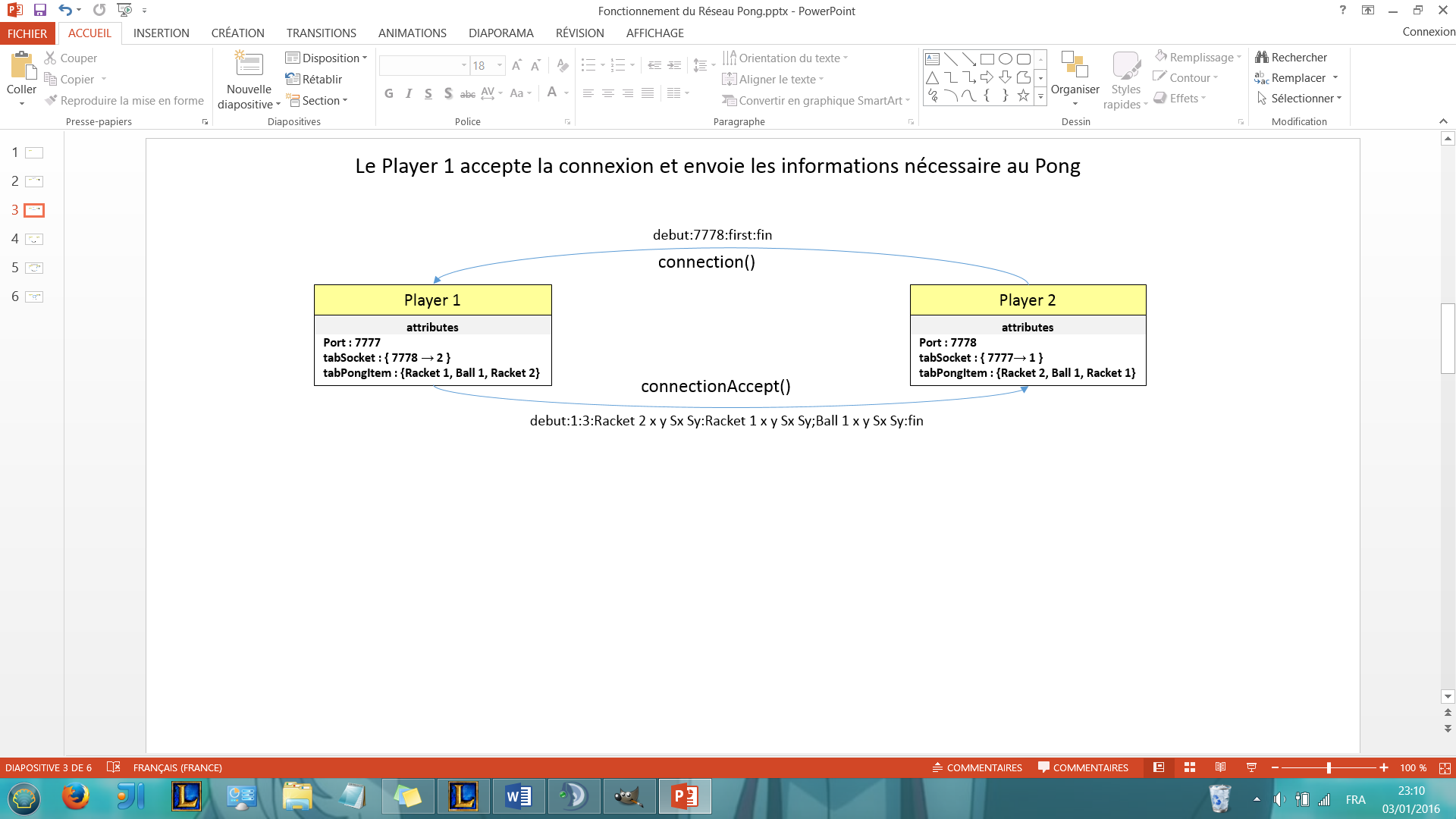
Le jeu Pong se joue ainsi entre 1 et 4 joueurs, et possède un Bonus (qui est aussi parfois un malus) au milieu. La trajectoire du Bonus est un cercle avec pour centre, le milieu de l’écran. Pour le débloquer, il faut réussir à le toucher avec la balle. C’est le dernier joueur à avoir touché la balle entrant en collision avec le Bonus, qui le récupère. Il y a seulement 4 bonus/malus : augmentation/diminution de la raquette/vitesse. Le bonus apparait au bout de 2-3 points marqués. Le bonus après avoir été frappé par une balle touchée au préalable par un joueur, se dirige vers le joueur. Pour récupérer le bonus, le joueur doit ensuite « l’attraper » et obtient ainsi les effets du Bonus (qui sont généré aléatoirement lors de la création de celui-ci. Le bonus agit ensuite sur la raquette du joueur pendant 5s, puis les effets du bonus s’exemptent et 5s plus tard, le bonus est réinitialisé au milieu.

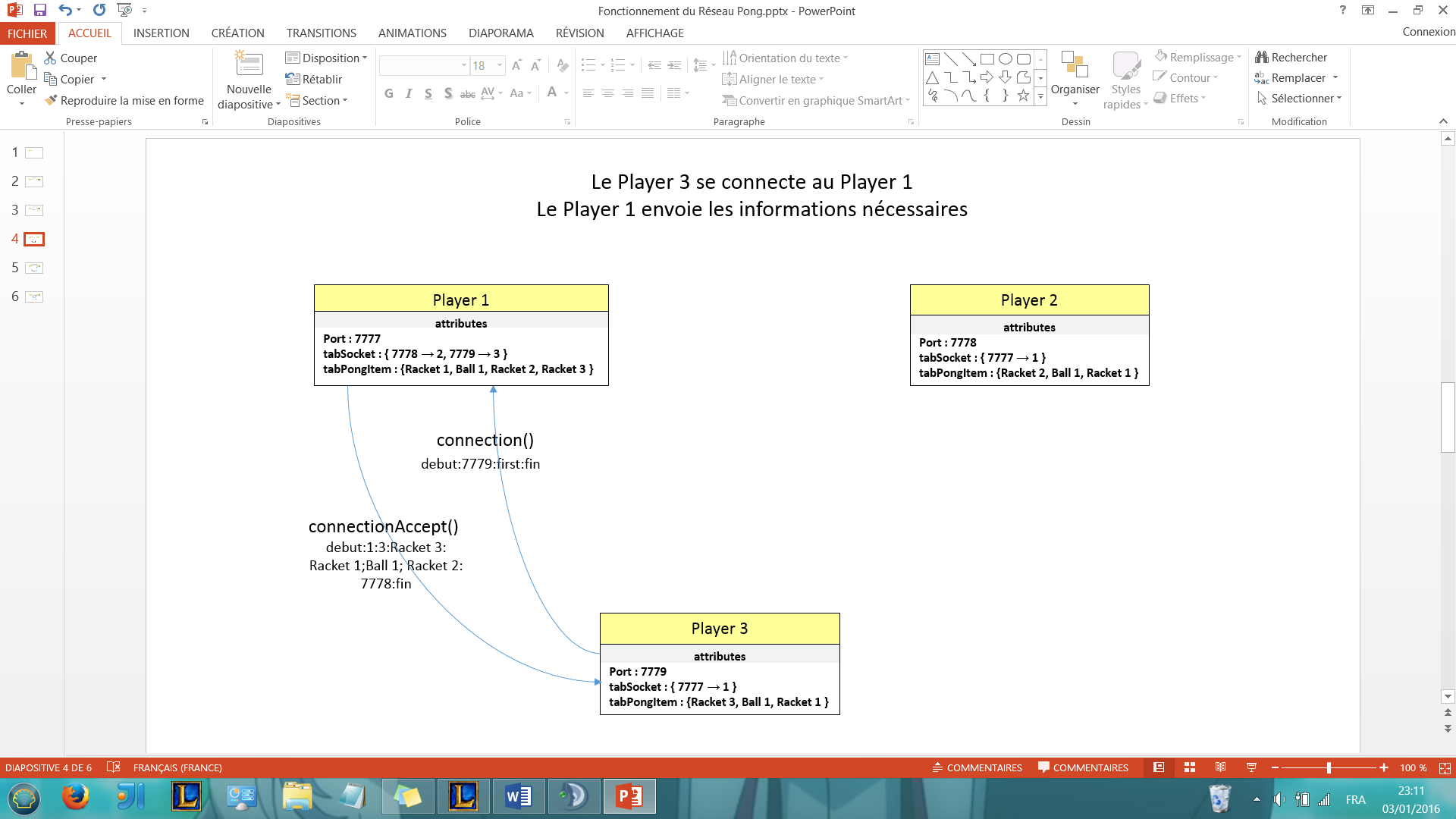
Lorsque la partie ce termine (au bout de 11points marqués par un joueur), un écran personnalisé apparait à chaque joueurs (différents selon si le joueur à gagner ou perdu.

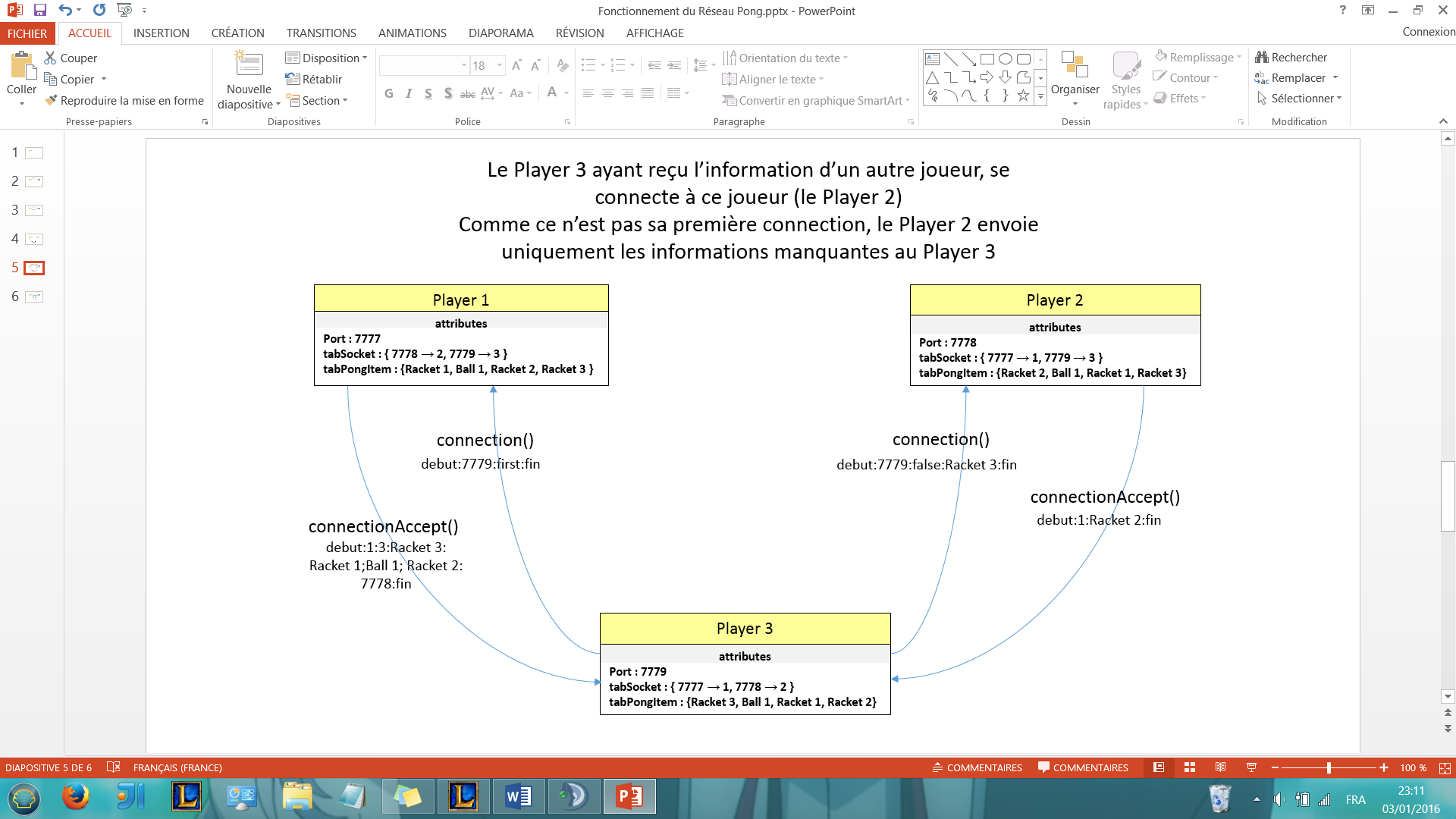


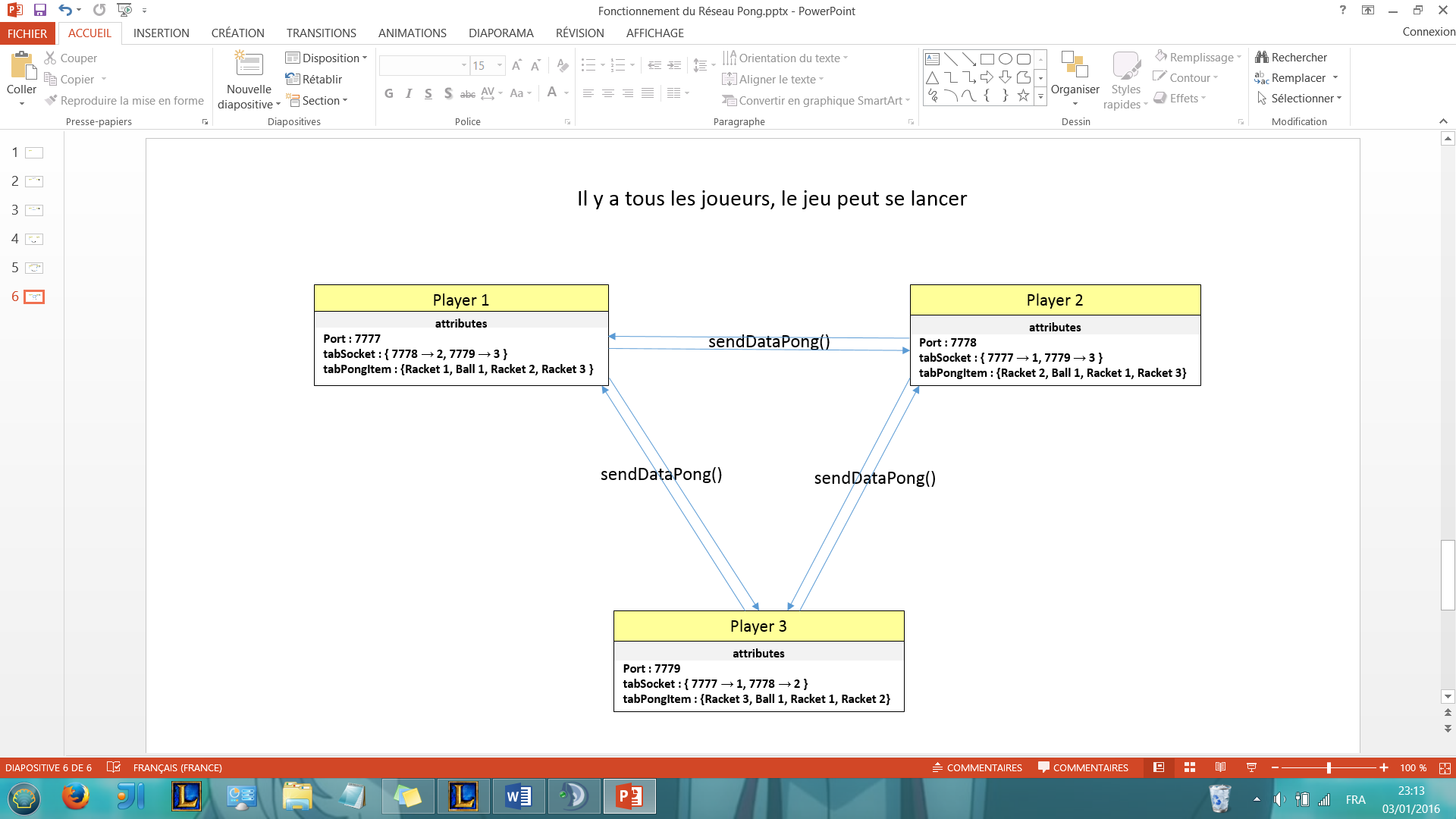
## 3. Fonctionnement de la connexion à 3 joueur











# II. Valorisation du travail réalisé

Concernant le Pong de base :

* La balle a une vitesse aléatoire à chaque initialisation !
* La balle rebondie avec un angle différent selon l’endroit de la raquette qu’elle a touché

Concernant le Pong en réseau :

* Il y a une fonction searchPort() qui permet de trouver un port libre, et ainsi on ne doit plus en chercher un manuellement.
* J’ai créé une fonction « String strProtocol (String... messages) », qui prend en paramètre un nombre illimité d’arguments (d’où le « String … » et qui les transforme de façon à obtenir une chaine de caractères conforme au protocole et prête à être envoyé.
* J’ai créé une classe SocketPlayer qui me permet de stocker plus d’informations qu’il y en a sur un simple socket (par exemple le numéro du joueur auquel appartient le socket)
* J’ai découpé au maximum mes fonctions afin d’obtenir des fonctions plus facilement compréhensible, maintenable et d’éviter les redondances.
* Pour les bonus, j’ai choisi de ne pas faire de serveur. Simplement car je trouvais que c’était en contradiction avec l’ensemble de notre projet ! En effet, depuis le début on a tout fait en sorte qu’il n’y ait pas de serveur, et que le jeu soit en Peer-to-Peer. Et soudainement pour le bonus, pour simplifier sûrement les choses, on nous demande de créer un serveur. J’ai donc décidé de poursuivre sur l’objectif de base qui était de réaliser ce jeu de Pong en Peer-to-Peer. J’ai donc dû trouver un moyen pour qu’il n’y ait pas de triche en réseau malgré l’absent de serveur. Je suis tout d’abord partie sur la même optique que pour les balles, c’est-à-dire une position de base au centre, puis une vitesse aléatoire. Le problème de cette solution, c’est que si un joueur décide de tricher et d’envoyé une vitesse à son avantage (que le bonus aille vers lui uniquement), les autres joueurs n’ont aucun moyen de vérifier qu’il n’a pas triché, puisque c’est une vitesse généré « aléatoire » et donc qu’une vitesse se dirigeant vers le joueur qui a lancé le bonus est tout à fait probable. J’ai donc pensé à une solution où tous les joueurs envoient une vitesse et faire ensuite la moyenne de tous les vitesses. Mais une fois de plus, un joueur pouvait décider de tricher et d’envoyer une vitesse le favorisant. Bien sûr, la vitesse serait quand même une moyenne, mais la probabilité que le bonus aille vers le joueur tricheur serait tout de même plus importante que pour les autres joueurs. Après maintes réflexions et prise de tête à n’en plus finir, j’ai exposé mon problème à des amis (non-informaticien), pour qu’ils aient une approche complétement différentes de la mienne. Après leur avoir expliqué le challenge, et les différentes contraintes, ils ont pensé à un bonus qui aurait une position et une vitesse/trajectoire prédéfinis, et que ce serait la personne qui arrive à toucher le bonus qui le gagnerait ! Mon problème était résolu, il fallait ensuite simplement que je le mette en place, en modifiant mon Pong de base (pour que les balles est un « propriétaire » par exemple). Je suis assez content du résultat car il est en adéquation avec mes conventions de faire du Peer-to-Peer jusqu’au bout, et d’éviter la triche (qui est facilement repérable avec un bonus prédéfinis).
* On peut supprimer un joueur sans que cela pose de problème. Le joueur est tout simplement enlevé !

# III. Diagramme des classes

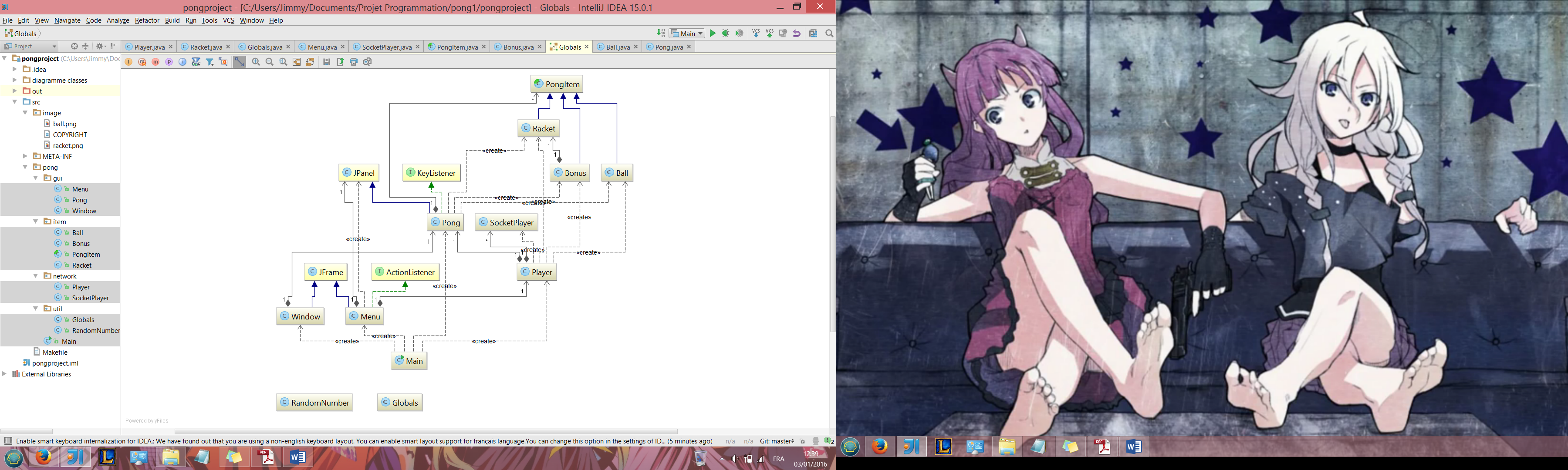


Figure 1 - Diagramme des Classes du Projet Pong

# IV. problèmes

Nous avons eu quelques soucis pour le réseau au début, mais une fois que le réseau fut maitriser (réceptions et envois des données), le projet de Pong fut très intéressant en terme des raisonnements. En effets, des problèmes que nous n’avions jamais rencontré se sont posé (comment faire pour que les données s’échange simultanément, comment faire quand une personne qui le jeu, comment faire un Peer-to-Peer, comment éviter certains lags qui sont apparus au début, comment faire lorsqu’un client se connecte pour qu’il sache où se connecter ensuite, comment faire pour que les informations ne soient pas envoyé de façon redondante, etc…). Pleins de façons de penser complétement différentes de problèmes « habituelles » sont venus et nous ont appris à penser différemment, et cela fut très intéressant intellectuellement !

Nous avons eu aussi pas mal de soucis par rapport à la « triche » surtout venant du fait que nous avons décidé de faire une balle avec une vitesse aléatoire, et donc non prévisible par les autres joueurs qui criés à la triche ! Nous avons aussi dû faire face au problème du Bonus qui permettait une triche facile, …

Mais tous ses problèmes sont plus ou moins derrière nous, mises à part la triche qui demanderait à être amélioré encore !

# V. Bibliographie commentée

Internet en général. J’ai lu beaucoup de sites différents, des tutoriels, etc. Voici une liste non exhaustive des sites que j’ai visionnés :

* https://openclassrooms.com/
* http://java.developpez.com/faq/java/?page=reseau#RESEAU\_ip
* http://docs.oracle.com/javase/6/docs/api/
* …