

Bugs connus pour la version graphique

Chargement

Après du chargement, le tour du joueur lors de la sauvegarde n'est pas forcément celui qui jouera immédiatement après la sauvegarde. Cela n'impacte néanmoins que le premier tour après la sauvegarde.

Sauvegarde

La sauvegarde comporte un bug qui peut être qualifié de majeur si l'on souhaite jouer sur plusieurs ordinateurs, donc voici la description.

Les sauvegardes effectuées sur un ordinateur ne peuvent être restaurées que sur ce dernier, et non sur un autre ordinateur.

En effet, ayant travaillé sur deux versions de jdk différentes, nous nous sommes aperçus que cela pouvait être une source d'erreur potentielle.

La source ci-dessous indique que les JComponents ne peuvent être sérialisés d'une version d'un jdk à l'autre.

Au vu de la façon dont nous sérialisons notre bataille navale, c'est-à-dire en sérialisant les panels, et donc leur composants, il nous est impossible de corriger ce bug, à moins de repenser la façon dont nous sauvegardons nos données.

Cela n'a pu être fait car nous avons découvert ce bug la veille du rendu du projet, lors des tests finaux, et que nous n'avions donc plus le temps nécessaire pour repenser une méthode de sauvegarde. Nous avons donc choisi de le laisser tel quel, en sachant que cela limite l'intérêt de la sauvegarde, plutôt que de tenter de modifier la procédure de sauvegarde de façon précipitée, au risque de perdre des fonctionnalités.

Nous avons inclus une sauvegarde fournissant un jeu pré joué avec notre bataille navale, mais nous ne pouvons donc garantir qu'il vous sera possible de la restaurer. Par contre, si vous venez à effectuer une sauvegarde, puis à la restaurer sur le même ordinateur, elle sera fonctionnelle.

Source : <http://stackoverflow.com/a/14933323>

Nous avons également rencontré un autre bug durant notre sauvegarde, qui est plus une remarque qu'un bug. Nous avons comparé avec les tailles des fichiers de sauvegarde de certains groupes (qui en moyenne font entre 100 et 800 Ko) et avons constaté que nos sauvegardes étaient très lourdes (près de 2Mo) pour une application de ce genre, sans que nous ne trouvions d'explication.

Timer

Le bloc de code à la fin de la méthode actionPerformed du ListenerTirer est commenté pour éviter une impression de « lenteur » entre deux tirs. En effet, nous avons placé un Thread.sleep(800) ici, car il s'agit du seul moyen que nous avons trouvé pour éviter que les joueurs humains ne voient le plateau adverse immédiatement après avoir tiré. Néanmoins, nous l'avons commenté, car il se révèle très dérangent lors des phases de tests.