

Guide de test du jeu XCube.

A l'arrivée sur l'écran d'accueil. On distingue trois modes de jeu, "Local", "IA" et "Multijoueur".

Les trois modes de jeux se basent sur la même classe pour ses mécaniques principales, il n'est ainsi pas utile de toutes les tester individuellement. Procédons donc dans l'ordre.

Test basiques en local :

Il s'agit dans un premier temps de tester le jeu en lui-même. On doit donc désactiver le DLC dans l'écran d'accueil puis cliquer sur "local". Le jeu se lance alors et le premier joueur peut commencer.

On peut rapidement réaliser 4 tests :

- On fait gagner les bleues en alignant 3 carrés bleues en ligne.
- On fait gagner les bleues en alignant trois carrés bleues en diagonale.
- On fait gagner les rouges en alignant trois carrés rouges comme on le souhaite.
- On attend sans jouer la fin du chronomètre.

Ainsi, on a pu voir que les côtés des carrés rouges et bleues se posent bien comme ils le devraient, que toutes les conditions de victoires possibles sont couvertes (3 carrés alignés diagonalement/verticalement/horizontalement). Et enfin que la condition de victoire ne dépend pas de la couleur et affiche la bonne couleur gagnante dans tous les cas.

On peut à présent activer le "DLC" qui est un simple ajout de carrés sur le terrain et remarquer que ça fonctionne encore correctement en testant de gagner en utilisant les nouveaux carrés.

On peut enfin cliquer sur le bouton adéquat sur le menu pour modifier la durée et constater que celle-ci est bien changée en jeu.

Test de L'IA :

Jouer une vraie partie contre celle-ci devrait suffire à se rendre compte qu'elle est opérationnelle.

Test du multijoueur :

Il va falloir lancer plusieurs clients pour tester le multijoueur. Pour ce faire, si on utilise android studio il faut d'abord lancer avec l'option de debug, puis de façon normale. Mais le mieux reste d'importer le projet gradle sur un autre IDE, Eclipse par exemple et de lancer la classe Desktop, ou bien d'utiliser le .jar.

Attention ! Dans le cas de l'utilisation du .jar particulièrement (et de façon générale). Il ne faut pas quitter le multi en cours de partie sous peine de faire bugger la déconnexion et d'entraîner des problèmes au niveau du serveur. Rendant tout autre test impossible*. C'est particulièrement vrai en utilisant les .jar (les IDE pouvant permettre de forcer la fermeture des clients, le problème se corrige facilement).

- Test de connection : Connecter les deux clients en cliquant pour chacun sur "multijoueur" dans le menu. Le premier client connecté hérite de la couleur "bleue". Le second de la couleur "rouge". Jouer deux coups avec le bleu puis essayer d'en jouer un troisième. Si rien ne se passe, alors c'est que contrairement au local on ne peut pas jouer les coups de l'adversaire sur son propre client, ce qui est rassurant. Finissez ensuite la partie normalement. Les autres tests n'étant pas nécessaire puisqu'ils ont été réalisés en local et que le multijoueur reprend très largement le code local.
- Test de bouclage : Une fois la partie précédente terminée, cliquez jusqu'à retourner au menu. Recliquez alors sur multijoueur pour les deux clients dans l'ordre précédemment établi. Une nouvelle partie devrait alors pouvoir se dérouler sans problème. Recommencez alors mais cette fois-ci en changeant l'ordre de connection (l'ancien bleu devient rouge, le rouge devient bleu). Vérifiez alors que cela fonctionne.
- Test du Matchmaking : Pour ce faire, on est obligé de passer par un autre IDE ou par un .jar pour lancer 4 clients en même temps. Connecter les deux premiers. Puis un troisième en cliquant sur "multijoueur". Pour les 2 premiers, une partie est normalement lancée. Le troisième doit être sur un écran d'attente de joueur. Connectez alors le 4e joueur. La partie devrait se lancer pour ces deux là. Vérifier que les deux parties sont bien décorrélées en jouant quelques coups.

*Par défaut, l'application se connecte à un serveur minet qui héberge le serveur du jeu. Si vous rencontrez un problème. Installer nodeJS et faire tourner le serveur (dossier server/src/index.js dans le projet) en local à l'aide de la commande node [chemin du serveur]. Au niveau du jeu, il faudra alors remplacer l'ip du serveur minet

par "localhost" ou "127.0.0.1" dans la ligne de connection "<http://ip:port>" de la classe "Multi".