

Manuel de déploiement

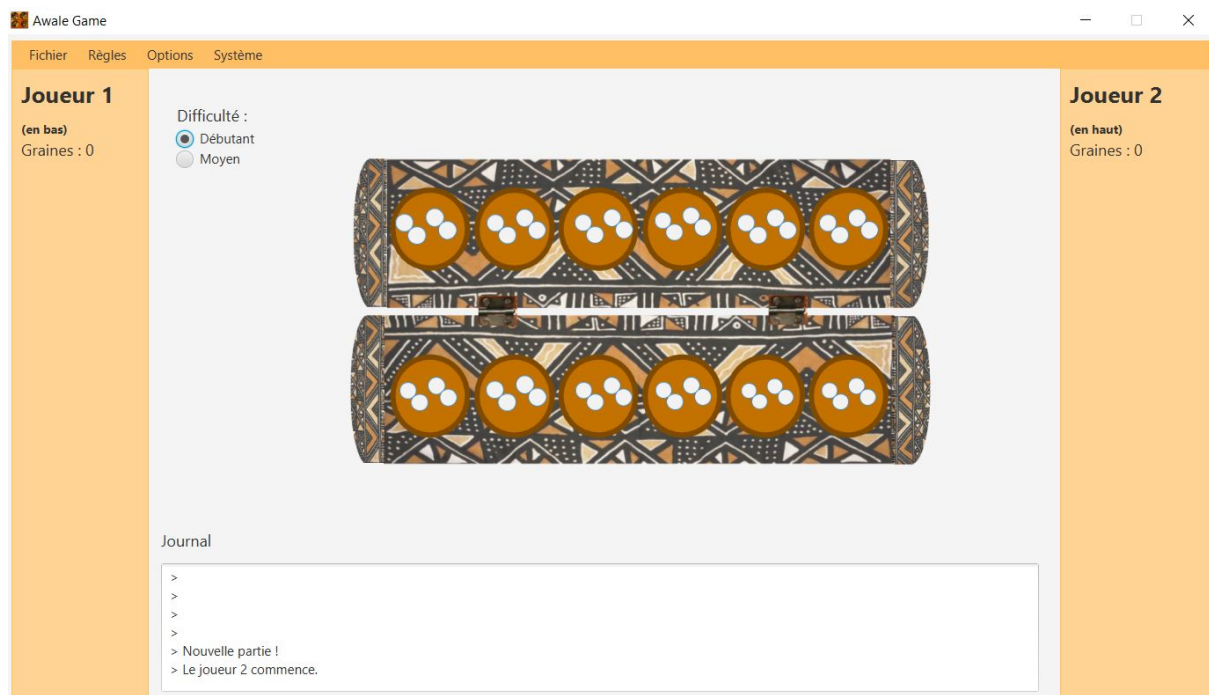
Pour jouer à notre jeu d'Awalé, vous pouvez simplement lancer l'exécutable fourni dans le rendu de cette manière :

```
> java -jar out\artifacts\Projet_JavaFX_AWALE_jar\Projet_JavaFX_AWALE.jar
```

Autrement, vous pouvez ajouter les sources dans un IDE gérant le javafx et lancer le projet ainsi. Pour cloner le repository GitHub :

```
> git clone https://github.com/BahiaSechi/Projet_JavaFX_AWALE.git
```

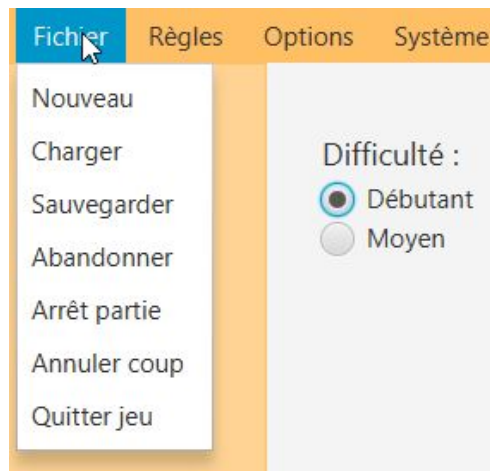
Manuel d'utilisation



Voici l'écran de jeu.

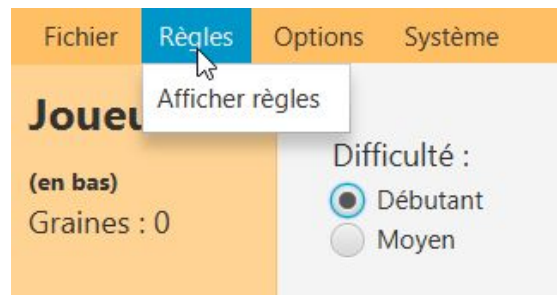
Le plateau du joueur 1 est celui du bas, le plateau du joueur 2 est celui du haut. Les scores des deux joueurs s'affichent à gauche et à droite de l'écran. Le journal, en bas de l'écran, contient plus d'informations sur les derniers coups joués.

Pour jouer un coup, cliquez sur une case du plateau. Vous pouvez régler la difficulté avec dans la partie "Difficulté" à gauche du plateau. Vous pouvez régler la difficulté sur "Débutant" ou "Moyen".

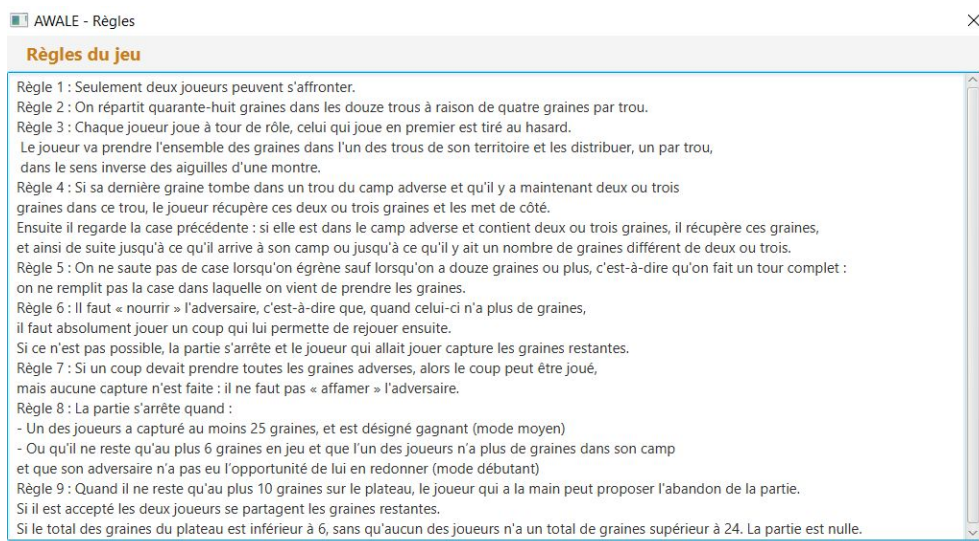


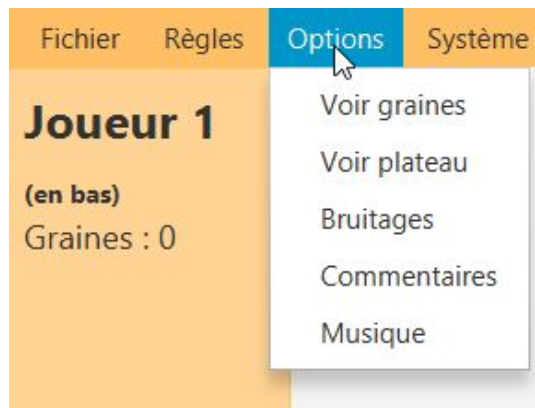
Voici le détail des boutons du menu “Fichiers” :

- “Nouveau” permet de relancer une partie.
- “Charger” permet de charger une partie à partir d’un fichier.
- “Sauvegarder” permet de sauvegarder votre partie dans un fichier.
- “Abandonner” permet d’abandonner la partie.
- “Arrêt partie” permet de proposer l’arrêt de la partie lorsqu’il reste moins de 10 graines sur le plateau.
- “Annuler coup” permet de revenir un coup en arrière.
- “Quitter jeu” permet de quitter le jeu.



Depuis le bouton “Afficher règles” vous pouvez afficher les règles de cette manière :



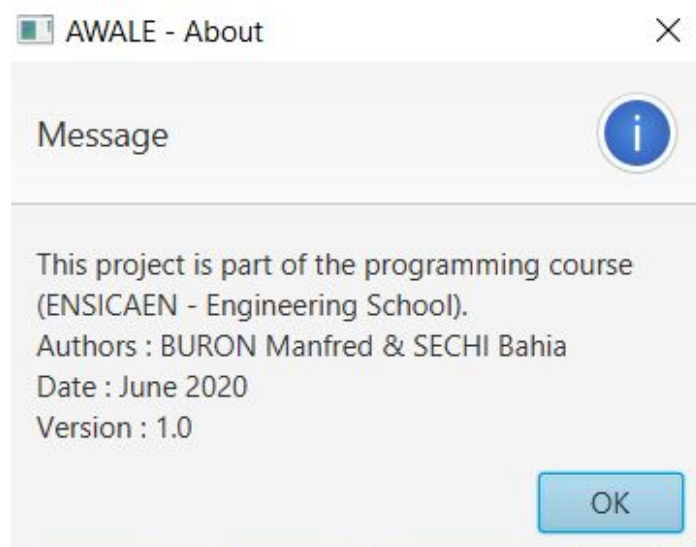


Voici le détail des boutons du menu “Options” :

- “Voir graines” permet d’afficher le nombre de graines dans une case lorsque l’on passe la souris dessus.
- “Voir plateau” permet d’afficher le nombre de graines dans chaque case du plateau. Le jeu n’affiche pas plus de 10 graines par case, il est donc conseillé de jouer avec cette option activée pour savoir exactement combien il y a de graines dans les cases.
- “Bruitages” permet d’activer le bruit de bille à chaque coup.
- “Commentaires” permet d’activer des commentaires que nous pouvons entendre en fonction de différentes situations possiblement rencontrées dans le jeu.
- “Musique” permet de lancer une musique lorsque l’option est activée. La musique choisie est l’hiver des 4 saisons de Vivaldi, c’est normal si le son est faible au début.



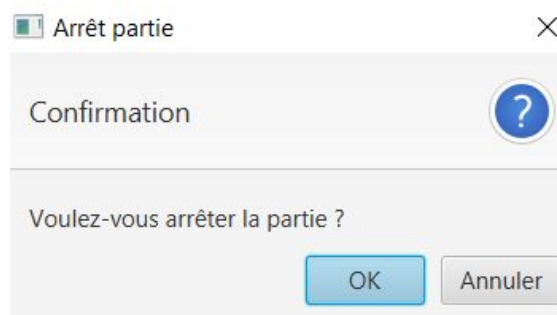
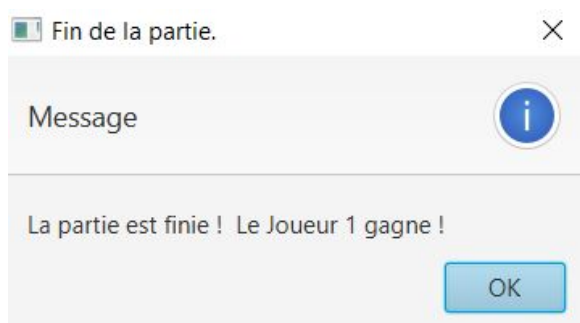
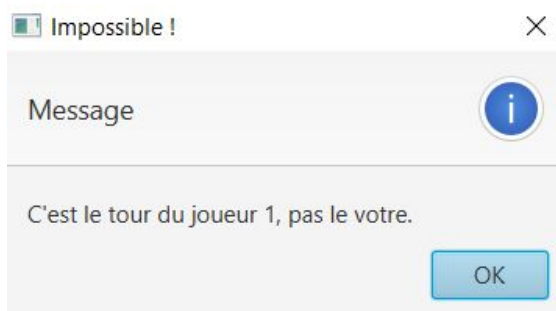
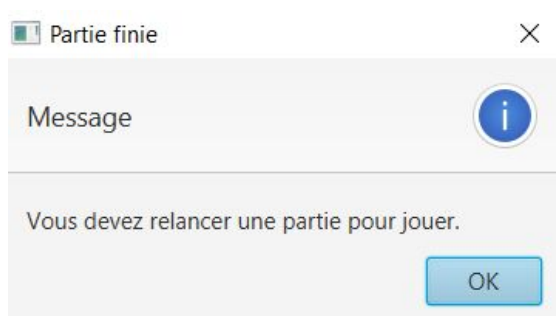
Depuis le bouton “A propos” vous pouvez afficher les informations sur le jeu de cette manière :



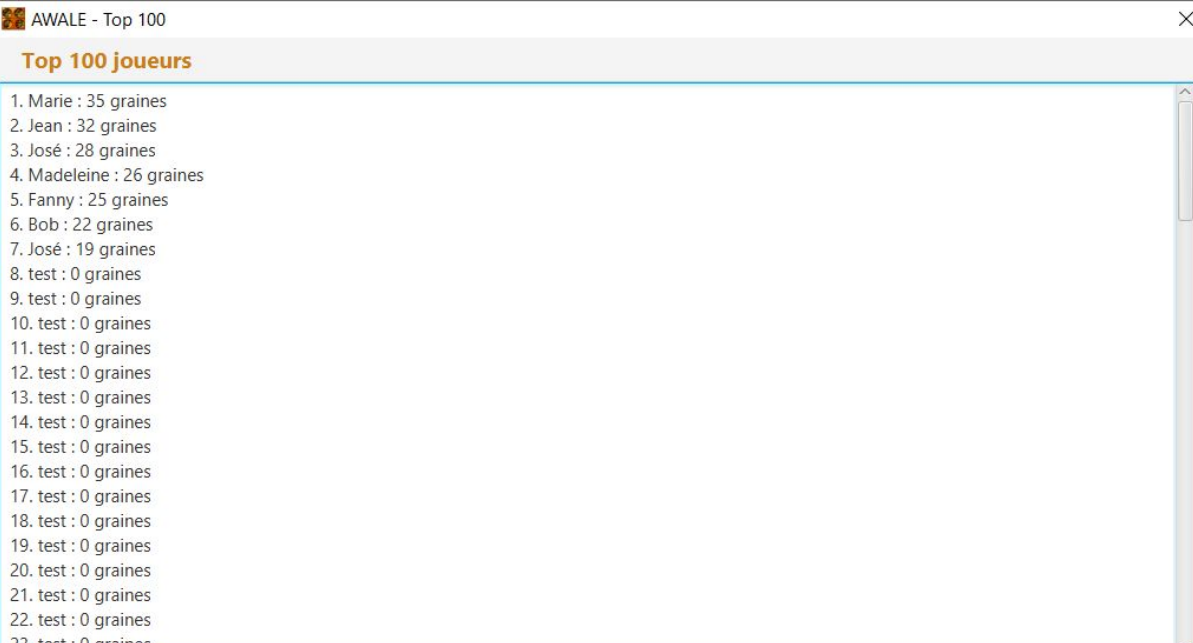
Voici le visuel du plateau lorsque les options “Voir graines” et “Voir plateau” sont cochées.



Quelques pop-ups pouvant apparaître au cours de la partie :



La fenêtre affichant les 100 meilleurs scores :



Top 100 joueurs	
1. Marie	: 35 graines
2. Jean	: 32 graines
3. José	: 28 graines
4. Madeleine	: 26 graines
5. Fanny	: 25 graines
6. Bob	: 22 graines
7. José	: 19 graines
8. test	: 0 graines
9. test	: 0 graines
10. test	: 0 graines
11. test	: 0 graines
12. test	: 0 graines
13. test	: 0 graines
14. test	: 0 graines
15. test	: 0 graines
16. test	: 0 graines
17. test	: 0 graines
18. test	: 0 graines
19. test	: 0 graines
20. test	: 0 graines
21. test	: 0 graines
22. test	: 0 graines
23. test	: 0 graines

Maintenant, à votre tour de jouer et de découvrir toutes les possibilités que renferme le jeu !

Gestion de projet

Conception :

Nous avons d'abord étudié le cahier des charges puis établi les besoins fonctionnels de notre projet. Le jeu étant assez simple, cette première partie fut assez rapide.

Ensuite nous avons fait le plateau de jeu et les images de billes afin d'avoir un support à partir duquel nous avons pu faire le reste du design de la fenêtre de jeu.

Documentation :

Nous avons généré la javadoc permettant de visualiser aisément les fonctions de notre projet. Vous pouvez y accéder dans le **dossier doc**, et ouvrir le fichier **index.html**.

Répartition du travail :

Nous n'avions pas fait de planning à l'avance. On toujours travaillé ensemble en vocal et jamais seuls de notre côté. Donc, on se tient au courant des avancées sur le code en temps réel en s'aidant au besoin.

Au final le travail s'est réparti de cette manière :

Bahia :

- Création des éléments graphiques.
- Gestion de la musique et des bruitages.
- Création du front end.
- Gestion des fichiers (sauvegarder, charger, top scores).
- Création de l'encart avec les logs et fonction d'envoi d'alerte.

Manfred :

- Créations des fonctionnalités permettant de jouer :
 - déplacer les billes
 - ramasser les billes gagnées
 - déterminer quand un coup est jouable
 - s'assurer de ne pas affamer son adversaire
- Gestion des différentes conditions de victoires.

Bien entendu, nous avons tout au long du projet travaillé sur GitHub. Voici un bref aperçu de nos 80 commits :

Merge branch 'master' of https://github.com/BahiaSechi/Projet_JavaFX_... ManfredBuron committed 4 days ago	3db2352	<>
resize pictures ManfredBuron committed 4 days ago	d63d304	<>
Modif graines J1 et J2, pas dans l'arraylist mais en int séparés, plu... BahiaSechi committed 4 days ago ...s simple	95ab8c9	<>
Commits on Jun 15, 2020		
fonction newGame et update view ManfredBuron committed 5 days ago	cf9d2d3	<>
Vue imageviews BahiaSechi committed 5 days ago	b591279	<>
ya une arraylist ManfredBuron committed 5 days ago	2fd7045	<>
ya une arraylist ManfredBuron committed 5 days ago	88f3d00	<>
Modifications : ... BahiaSechi committed 5 days ago - Vue resize / Couleur pas très belle mais on verra	abe81d9	<>
test ManfredBuron committed 5 days ago	b125635	<>
Modifications : ... BahiaSechi committed 5 days ago - Location images - Quelques fonctions de menu	cce980a	<>

Activer V
Accédez aux

Fonctionnalités implémentées :

Il est possible de jouer à deux joueurs en suivant toutes les règles précisées dans l'énoncé en partie locale sur le même écran, en tour par tour. Toutes les options de menu spécifié dans l'énoncé sont implémentées également, à savoir, toutes ces dernières :

Menu Awalé

- o Nouveau: commence une nouvelle partie
- o Charger : recharge en mémoire une partie sauvegardée.
- o Sauvegarder : sauvegarder l'état de la partie.
- o Abandonner : termine la partie immédiatement (l'adversaire capture les graines restantes)
- o Arrêt Partie: arrête la partie en cours.
- o Annuler dernier coup : Revient en arrière d'un tour de jeu

Menu Règles

- o Voir la règle : Visualise la règle.

Menu Options

- o Voir nombre de graines : Si cette option est active, le nombre de graines présentes dans le trou que vous pointez avec la souris s'affiche.
 - o Voir état plateau : Si cette option est active, le nombre de graines contenues dans chaque trou est affiché.
- o Bruitages : Si cette option est active, le jeu est bruité.
- o Commentaires: *Si cette option est active, le jeu est commenté. (bonus)*
- o Musiques : Si cette option est active, une musique se fait entendre durant le jeu.

Menu système

o Ouvre la boîte d'information donnant la version d'Awalé et les noms des réalisateurs.

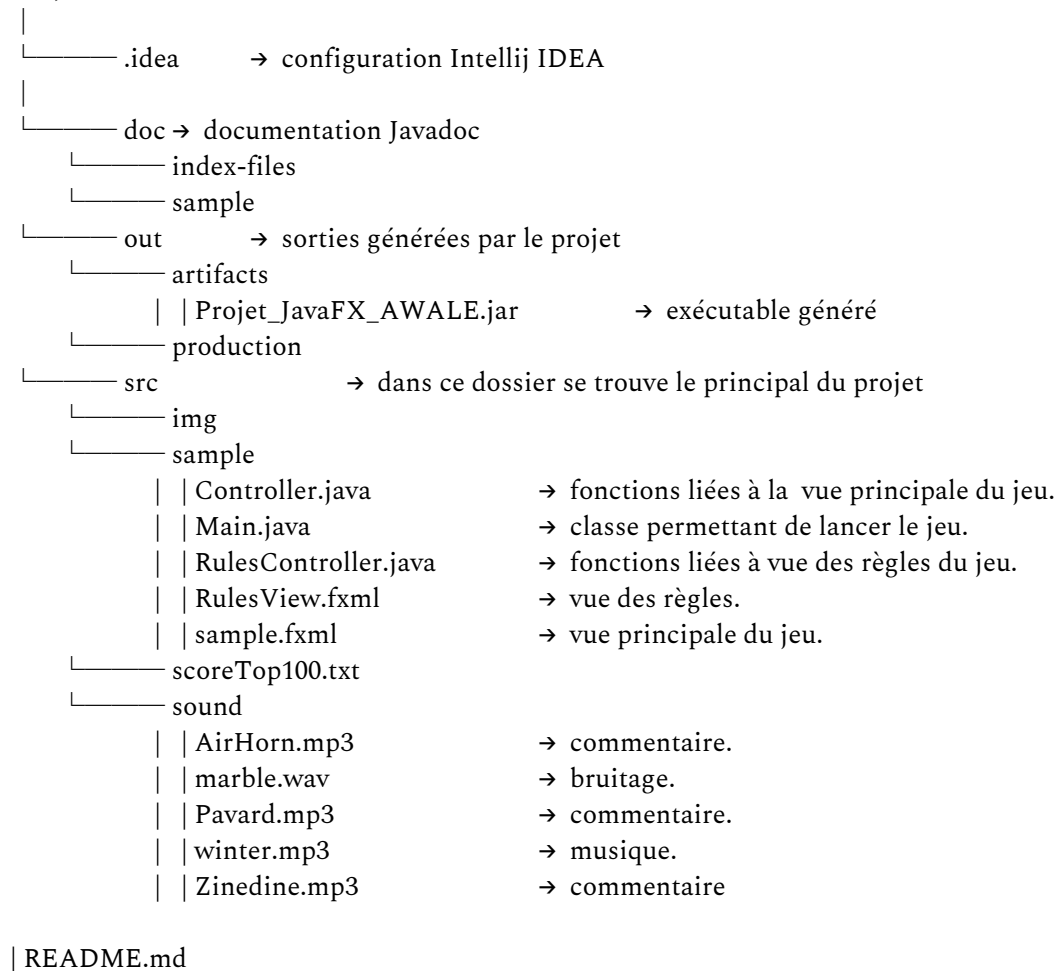
Nous reviendrons sur les fonctionnalités bonus que nous avons implémentées plus bas.

Fonctionnalités non-implémentées :

L'unique fonctionnalité que nous n'avons pas implémentée est le système de serveur qui héberge une partie pour permettre à deux joueurs de jouer ensemble en ayant chacun une instance d'un client. Nous reviendrons sur nos raisons dans les remarques supplémentaires.

Structure du projet :

Projet_JavaFX_AWALE



Particularité de notre version :

Nous avons également implémenté un système de journal (logs) donnant des informations pour informer les joueurs de ce qu'il se passe pendant la partie.

Enfin, pour nous amuser et apporter une touche d'originalité, nous avons également ajouté des commentaires, comme on peut en avoir pour le sport, qui se déclenchent lors de certaines situations précises. Cette fonctionnalité est activable dans les options, de la même manière que la musique et le bruitage. Voici les déclencheurs des commentaires :

- Lorsque les scores des deux joueurs sont au dessus de 10 et que l'un d'eux égalise.

- Lorsque un joueur ne marque pas de point et qu'il a au moins 15 points de de retard avec son adversaire.
- Lorsque la partie est finie.

Évidemment on vous laisse découvrir par vous même ce que les commentateurs ont à dire du match que vous jouez. Précisons également que chaque commentaire ne se joue qu'une fois par partie.

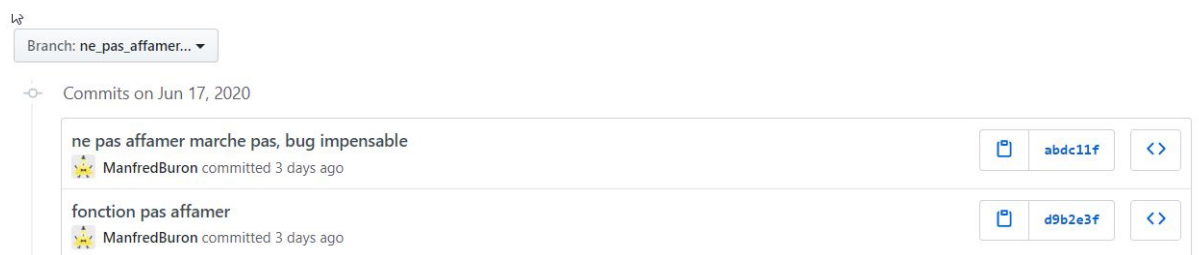
Choix ergonomiques :

- Les logs sont une manière d'afficher des informations utiles sans surcharger le joueur de pop-ups, ce qui peut être très désagréable en pleine partie ; nous voulions minimiser ces dernières au strict nécessaire.

Problèmes rencontrés :

Nous avons rencontrés plusieurs problèmes lors du développement et nous avons réussi à résoudre la plupart de ces derniers. Voici quelques exemples de problèmes rencontrés :

- Calcul de valeur tampon pour savoir si un coup va affamer notre adversaire ou non. Pour cette fonctionnalité nous avons eu quelques soucis d'attribution des valeurs. Nous croyions, à un moment, que nous n'implémenterions pas cette fonctionnalité à cause d'un bug. Nous avons quand même gardé le code dans une branche à part du git au cas où (plus de 2 heures de travail). Finalement, c'est en travaillant sur une autre fonction le lendemain que la solution nous est apparue et nous avons pu reprendre le travail sur la branche du git pour le corriger et donc implémenter cette règle au jeu.



Et sur la branche master :



- Une musique qui fonctionne. Lors du développement de cette fonction, Bahia n'entendait jamais la musique même après avoir tout essayé et alors qu'aucun message d'erreur n'apparaissait. Dans le doute, on a fini par tester sur la machine de Manfred et cela a fonctionné. Pour une raison que nous ignorons, l'ordinateur de Bahia ne détecte pas les sons du jeu et cela nous a fait perdre une heure ou deux mais encore une fois, avec un peu de persistance, nous avons triomphé.
- Un autre problème lié à l'audio concerne les bruitages. Pour une raison qui nous est inconnue encore une fois, les bruitages ne se jouent qu'un coup sur deux. Cette fonctionnalité n'étant pas déterminante dans le jeu, nous ne nous sommes pas attardés dessus pour résoudre ce soucis.

Il nous est également parvenu quelques moments de doutes lors du développement mais cela est inhérent au développement informatique et on ne peut pas réellement parler de “Problèmes” dans la mesure où tout se règle en l’espace de quelques minutes.

Ce que nous aurions aimé faire :

Nous aurions aimé rajouter plus de commentaires audio pour mettre les joueurs dans l’ambiance mais cela demande du temps de trouver les meilleurs passages.

Améliorations possibles :

Nous pourrions organiser un peu mieux notre code (un peu de répartition en plusieurs fichiers et un peu de factorisation) car de beaux commentaires ne suffisent pas toujours pour s’y retrouver.

Nous mettre d’accord sur une langue pour écrire les fonctions et surtout la documentation. Pour l’instant, mieux vaut être bilingue franglais.

Remarques supplémentaire :

La raison pour laquelle notre jeu ne fonctionne pas avec un serveur est que la question n’est pas mentionné dans le sujet, il y a juste un lien vers les sockets. De plus nous n’avons pas parlé de cette technologie lors des cours. Cela ne nous a donc pas semblé être le plus important pour le projet. Nous nous sommes donc d’abord focalisés sur l’implémentation des fonctionnalités permettant de jouer. Lorsque ceci fut fait, nous nous sommes intéressés aux sockets et nous sommes rendu compte qu’il aurait fallu complètement revoir notre code alors que le temps manquait. Nous aurions dû prendre en compte cette partie dès la conception si nous avions voulu l’implémenter.

Nous savons que ce manque de clarté dans les consignes est principalement dû au fait que les cours ont eu lieu à distance. Nous espérons que vous ne nous en tiendrez pas rigueur compte-tenu de la qualité du travail fourni pour tout le reste du projet.

Très bonne fin d’année à vous et à l’année prochaine. :)