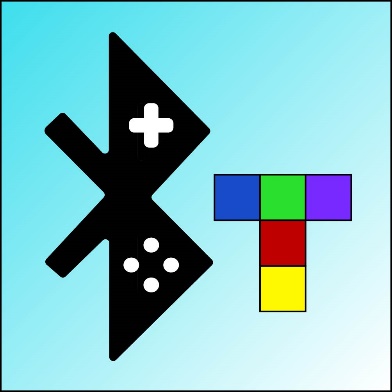
***Rapport Final BT-Games***



* **Cahier des charges**

Le but initial était de créer une console de jeux portable constituée d’un écran, d’un module BT et de la carte Arduino. Pour interagir avec la console l’objectif était d’utiliser le téléphone comme manette et donc éventuellement coder une application. On voulait à la base qu’il y ait au moins 3 jeux jouables sur la console et au moins 1 solo et 1 multi. De plus on voulait ajouter quelques autres fonctionnalités que simplement la manette sur le téléphone.

* **Console**

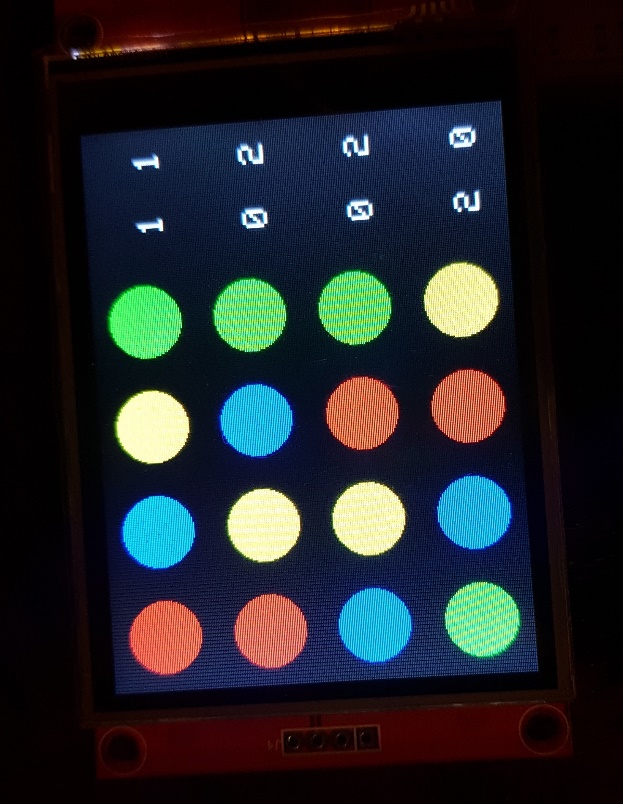
Au niveau de la console, il y a 7 jeux différents disponibles :

* Reflex (1 joueur)
* Simon (1 joueur)
* Mastermind (1 ou 2 joueurs)
* Blackjack (1 ou 2 joueurs)
* Yams (2 joueurs)
* Pendu (1 ou 2 joueurs)
* Morpion (2 joueurs)

Tous ces jeux ont été codé de zéro (sans prendre de code tout fait sur internet) d’abord dans une version CLI (Command Line Interface) et ensuite intégrer sur l’écran.

Les jeux les plus durs à programmer ont été le Mastermind, le Blackjack et le Yams. Le Mastermind a été difficile à intégrer à l’écran, la gestion des coups précédents et leurs affichages a rendu la tache plus ardue que prévu. Pour ce qui est du Blackjack la partie complexe a été la vérification du gagnant en particulier à 2 joueurs car il faut prendre en compte toutes les possibilités. Enfin le Yams les 2 difficultés rencontrés ont été l’affichage des dés sur l’écran (gestion des coordonnées) et le calcul des scores / reconnaissance des figures effectués.

Vous pouvez regarder les tests des jeux via la playlist YouTube : <https://www.youtube.com/playlist?list=PL39GM1mJVgYZjbCEZOsycZlSEwhUoDfaK>  
Et le code Arduino est disponible sur notre GitHub : <https://github.com/AndreaSamuel/BT-Games/blob/master/RapportsSeance/Code/BT-Games.ino>

*Exemple du jeu Mastermind à l’écran :  
*

* **BT-Controller**

Passons maintenant à l’application « BT-Controller », elle a été réaliser en utilisant MIT App Inventor 2 ( <https://appinventor.mit.edu/> ) qui nous a permis de gagner énormément de temps mais qui en contre partie nous a quelques fois restreint.

Sur « BT-Controller » il y a :

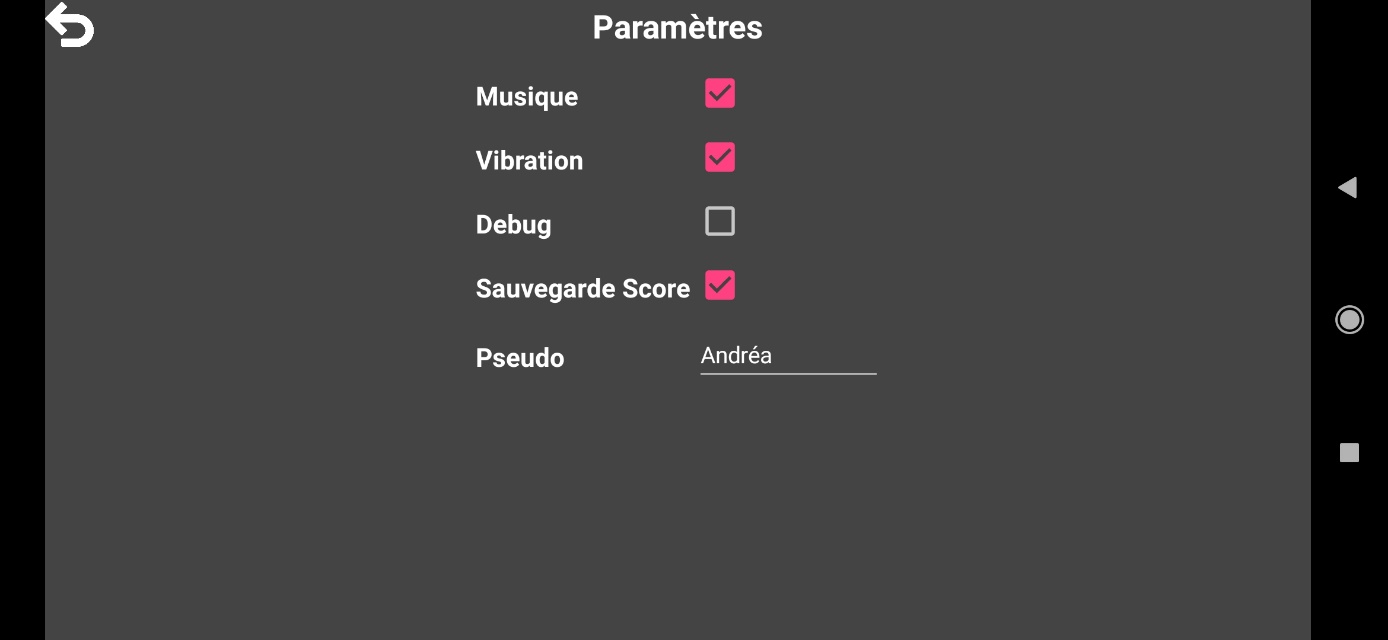
* Le menu principal
* Les 6 manettes différentes
* Un onglet « règles »
* Un onglet « Paramètres »
* Un onglet « Tchat »
* Un panel administrateur

Le menu principal permet de lancer les jeux et d’accéder au tableau des scores. Les 3 onglets « règles », « Paramètres » et « Tchat » sont disponible tous le temps (même en jeu). Les paramètres sont sauvegardés dans un fichier et donc sont enregistrer au redémarrage de l’application.

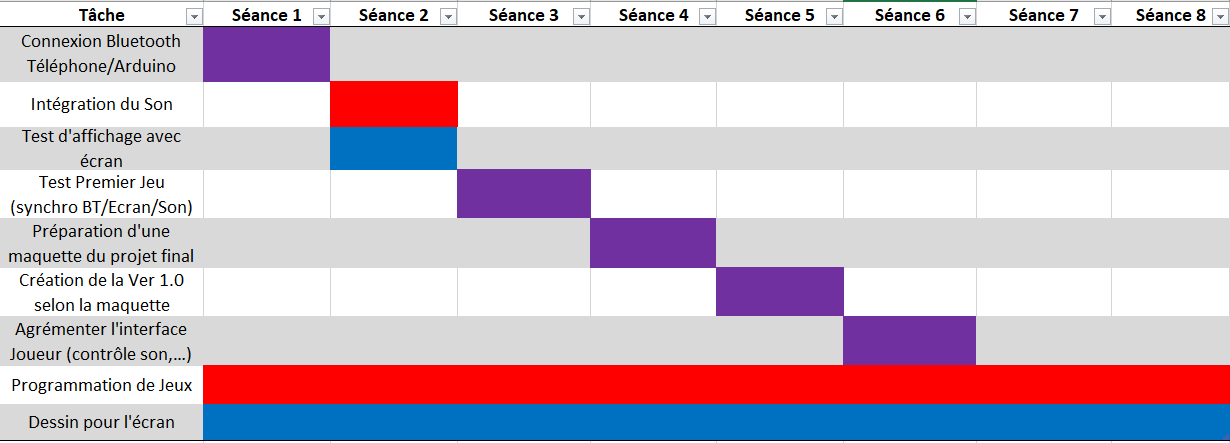
Les 3 plus grandes difficultés rencontrés ont été :

* La sauvegarde des paramètres dans un fichier qui n’est pas évident à faire avec App Inventor.
* L’affichage et la disparition des éléments à l’écran étant donné que pratiquement toute l’application est codée sur un seul écran.
* La communication BT et la synchronisation avec la console.

Vous pouvez récupérer l’application BT-Controller sur notre GitHub : <https://github.com/AndreaSamuel/BT-Games/blob/master/RapportsSeance/Code/BT-Controller.apk>

*Exemples de rendu de l’application :*

* **Comparatif planning prévisionnel et réel**



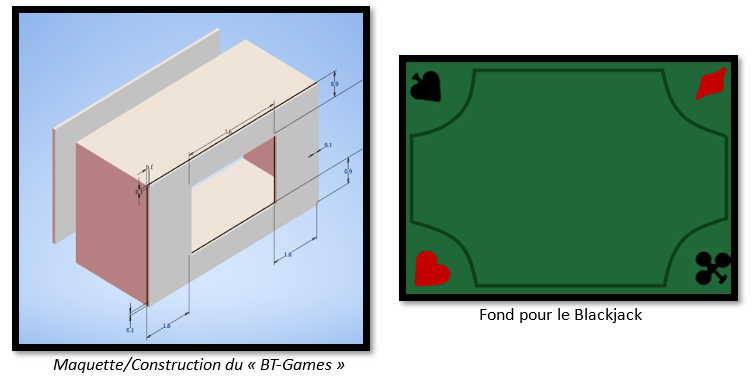
**Andréa  
Samuel**  
**Les Deux**

* Lors de la première séance, nous avons bien respecté ce qui était prévu. Nous avons choisi quel module Bluetooth on allait utiliser (HC-06), et effectué quelques tests de programmes basiques pour s’assurer que nos téléphones pouvaient bien se connecter à la carte via le module. L’application Android a commencée à être développé, et quelques tests de son ont été réalisés.
* Pour la 2e séance, nous n’avions pas encore l’écran. Nous avons donc pu intégrer la musique car c’était sur l’application Android, mais nous n’avons pas pu tester l’écran comme prévu. On a donc réalisé les taches dont on pouvait s’occuper sans l’écran : création du logo, et des images de fond (les « décors »). 2 jeux ont été implémentés dans le code et dans l’application.
* Pour la 3e séance, l’écran n’était toujours pas arrivé. Nous avons commencé la programmation de l’affichage des images de fond et certaines fonctions basiques, sans pouvoir les tester.
* Pendant la 4e séance, nous avons ajouté 2 nouveaux jeux. Nous avons aussi pu tester (enfin) sur l’écran d’afficher les premiers jeux. Nous avions prévu, au début du projet, de réaliser un test de maquette pour y insérer les composants. L’écran étant à peine arrivé, nous avons préféré nous concentrer sur l’affichage des jeux ainsi que la création de nouveaux jeux. Nous voulions un projet avec plusieurs jeux aboutis, plutôt qu’un beau support pour nos composants.
* Ayant choisi de ne pas faire de maquette, nous avons continué le développement des jeux et leur affichage pendant cette 5e séance. Nous avons changé de bibliothèque, pour une nouvelle, plus performante et complète que celle que nous utilisions avant. Certains jeux ont été optimisés, et nous avons plusieurs jeux totalement fonctionnels (solo/multi, affichage sur l’écran, communication Bluetooth grâce à la manette).
* Pour cette 6e séance, l’interface de l’application a été améliorée, et il y a maintenant un tableau des meilleurs scores et un tchat. Nous avons aussi implémenté d’autres jeux sur l’écran. Certaines personnes ont pu tester nos jeux, et leurs avis nous ont permis d’améliorer la compréhension de l’interface ainsi que l’affichage de certains jeux.
* Dans notre planning initial, nous n’avions pas prévu de tache spécifique à réaliser lors des 2 dernières séances, car nous savions que ce qu’il allait rester à faire serait de terminer les jeux et l’affichage. Nous avons ainsi apporté des modifications à l’application Android, ajouté des musiques, vibrations et pseudos. Nous avons ajouté un nouveau jeu et réalisé son affichage sur l’écran, et commencé l’affichage des derniers jeux.
* Lors de la 8e et dernière séance, nous avons finalisé le projet. Les derniers bugs ont été corrigés, certains paramètres de l’application sont mieux gérés. Nous avons terminé l’affichage de tous les jeux, et apporté quelques modifications aux jeux existants. Nous avons aussi commencé les plans d’une maquette qui pourrait servir de support pour l’écran et le reste des composants.
* **Conclusion**

Ce projet nous a appris les contraintes extérieures que nous pouvions avoir, la principale étant le retard de livraison de l’écran, qui a décalé notre planning et nous a forcé à nous adapter. Nous avons aussi eu des problèmes de mémoire et performance liés à notre carte Arduino, nous avons donc dû opter pour une Méga. Nous avons aussi découvert que l’environnement de travail avait un impact sur l’efficacité. Par exemple, nous avions régulièrement des problèmes/déconnexions Bluetooth car il y avait beaucoup d’appareils dans la pièce qui interféraient dans le transfert des données et nous ralentissait. Travailler avec du bruit autour (projet avec un moteur à 1m de nous) a aussi un impact sur notre concentration et notre efficacité. Nous nous sommes adaptés en réalisant le plus gros parti du travail chez nous.

Avec l’expérience que nous avons acquise, si nous devions recommencer le projet, nous pourrions mieux optimiser le code et l’affichage des jeux sur l’écran. Le contenu, quant à lui, serait assez similaire. Nous pourrions cependant rajouter des fonctionnalités aux jeux au moment de leur création. Avec plus de temps, nous pourrions développer l’application à l’aide d’Android Studio et non MIT App Inventor, ce dernier ayant restreint l’application en termes de quantité de musiques et fonctionnalités.

Le principal ajout au projet serait une carte SD. Nous pourrions, grâce à des images, rendre nos jeux plus détaillés, plus explicites pour le joueur, et apporter une touche « artistique ». Nous aimerions aussi créer un support pour l’écran, pour ainsi camoufler les câbles et vraiment avoir cet aspect de console. Mais notre volonté première étant de créer une console portable avec un téléphone comme manette, le fait d’avoir un support pour l’écran pourrait rendre plus difficile le transport et l’utilisation de la console.

*Exemples d’éléments d’améliorations commencés : *