## Synthèse personnelle

Depuis le commencement de l'activité sur la programmation d'une voiture, mon groupe (le groupe 2), composé de Gustave Migdal, Apollinaire Kaufman, Clara Caselli, Denise Bonasegla, Marcello Franzosi, et moi-même, avons réussi à modifier de nombreuses choses sur la voiture (par exemple : faire en sorte qu'elle puisse tourner et avancer en même temps, la faire s'arrêter quand elle est à moins de 40cm de quelque chose...).

J'étais (et je le suis toujours) en charge de programmer l'interface utilisée pour télécommander la voiture avec mon coéquipier, Apollinaire Kaufman. Pour cela, nous avons utilisé App Inventor, dont nous avions déjà la maîtrise grâce à notre « entraînement » l'année dernière.

Nous avons profité de nos heures de permanence commune pour aller en salle de technologie afin de réfléchir à deux sur les différents problèmes rencontrés durant la période. Quand nous n'arrivions pas à finir à temps, nous finissions le travail à la maison, le soir même, en communiquant par Skype. Nous avons aussi aidé notre camarade Gustave Migdal sur Mblock.

Ensemble, nous donc avons réussi à :

- faire en sorte que la voiture avance lorsqu'on appuie sur le bouton « avancer » et s'arrête lorsqu'on enlève son doigt (plus besoin du bouton stop comme dans la version de base)

- ajouter des boutons permettant à la voiture d'avancer (ou de reculer) et de tourner en même temps

- ajouter la possibilité de se déconnecter puis se reconnecter de la voiture

- modifier le programme pour faire en sorte que lorsque l'on ne fait rien pendant plus de 30s, la voiture se déconnecte

- nous avons aussi modifié l'interface pour la rendre pour uniforme, plus esthétique :



Pour cela, j'ai modifié le programme initial. Mon programme complet ressemble à ça :



Donc pour commencer, la partie suivante est activée dès que « Screen1 » (l'écran) est mis en route, donc dès le lancement de l'application permettant de contrôler la voiture. Elle a pour but de désactiver les boutons.

quand	Screen1 .Initialise
faire	mettre [btnForward •]. Activé • à [faux •]
	mettre [btnReverse · ]. Activé · ] à [ [faux · ]
	mettre [btnRight ]. Activé ] à [faux ]
	mettre [btnLeft]. Activé] à [faux]
	mettre reculer_droite . Activé à faux
	mettre reculer_gauche . Activé a faux
	mettre avancer_droite . Activé a faux
	mettre [avancer_gauche •] . Activé •) à 🔰 faux •)
	mettre [btnDisconnect ] . Visible ] à [faux ]

Elle permet aussi de cacher le bouton « Disconnect » (dernière partie en vert foncé), car celui-ci ne doit apparaître qu'à partir du moment où on est connecté à la voiture afin de nous donner la possibilité de nous déconnecter (au lancement de l'application, la voiture est obligatoirement déconnectée).

Ensuite, la partie suivante permet de relier « IstPairedBluetoothDevices » (qui est le bouton qui permet de se connecter à la voiture) à la liste des voitures dont le Bluetooth est disponible sur cette tablette (chaque voiture étant reliée à une tablette, nous n'avons qu'une seule possibilité).



Pour se connecter, il suffit d'appuyer se le bouton « Connect to Robot » (ou IstPairedBluetoothDevices). Après que l'on ait cliqué sur le bouton, le programme :



On peut alors télécommander la voiture ! En effet, quand on appuie sur un bouton (ici le bouton pour aller à droite),

L	1.	
la valeur correspondant à		d btnRight . Enfoncé
cette direction est envoyée à		appeler BluetoothClient1 . Envoyer texte
la voiture (ici « 3 ») —		texte 📜 " var=3\n "
et le chronomètre est		mettre Clock1 x IntervalleChronomètre x à (30000)
mis en route.		

Quand on retire son doigt, la valeur correspondant à l'arrêt (5) est envoyée à la voiture et celle-ci s'arrête.



Enfin, pour se déconnecter, il suffit d'appuyer sur le bouton « disconnect to Robot » :



Le chronomètre, lui, est activé quand on ne fait rien pendant plus de 30 seconde. En effet, après chaque appui sur un bouton, un compte à rebours est déclenché (30 000 ms) et la voiture se déconnecte grâce au même programme que pour le bouton disconnect :

