Université Pierre et Marie Curie Faculté d'Ingénierie

Rapport de Projet STL

Alexandre Fernandez
Sylvain Ung

Encadré par
Vincent Botbol
Ghiles Ziat

Le 27 février 2017

Table des matières

Première partie

Introduction

Chapitre 1

Sujet

Dans le cadre des activités robotiques au sein du M2 STL, nous aurions besoin d'explorer la programmation sur les drones. A l'heure actuelle, toutes les réalisations robotiques du M2 STL se basent sur des plateformes mobiles terrestres. Il est souhaitable d'explorer d'autres possibilités de support. Nous proposons de nous attaquer à ce projet sous un angle "Kid games", en récoltant des données de jeux multi-joueurs et en envoyant des commandes de déplacement selon ces données sur un drone baptisé "RockAn'Dron".

Le premier de tels jeux serait un Tambour-Hero où de multiples joueurs jouent en parallèle un jeu de rythme sur des smartphones Android en mode "client". Les résultats de chaque joueur sont évalués dynamiquement sur chaque support "client" avant d'être synchronisés sur un smartphone Android en mode "serveur". Les évaluations de chaque joueur doivent refléter sa capacité à imiter un batteur dans un morceau de musique folle. A chaque synchronisation sur le "serveur"(de l'ordre de la seconde par exemple), le smartphone "serveur" décide le gagnant actuel et envoie le drone faire un pas vers le gagnant en question. Le jeu termine quand un des joueurs arrive à attirer vers lui le drone, porteur de gros paquets de bonbons, ou de nouilles, voire d'autres choses selon des envies du moment.

Ce PSTL s'adresse à des étudiants en M1 STL curieux des réalisations robotiques ouvertes au grand publique de type "tout en kit" [1]. Dépendant du niveau de chaque participant au PSTL, on commencerait avec une partie de programmation purement Android, puis, on réfléchirait à l'intégrer dans les actionneurs du drone ; ou l'inverse ; voire on ferait des choses complètement imprévues. Une réalisation, cependant, est attendue au terme du PSTL, avec poster de présentation et montage de vidéo-clip.

1. Les étudiants curieux des réalisations robotiques à partir de (quasiment) rien pourraient s'intéresser au PSTL "Prototype de robot mobile".

L'avantage de ce PSTL est qu'il permet d'acquérir des compétences à la fois en programmation Android (partie jeu multi-joueurs) et en programmation sur les drone (partie RockAn'Dron). Le PSTL introduit également aux activités robotiques au sein du M2 STL. Le point délicat de ce PSTL est qu'un drone autonome est toujours très dangereux, parfois même assez suicidaire, à l'instar de ceux envoyés sur Mars...

Chapitre 2

Enjeu

Il va principalement s'agir dans ce projet de faire des expérimentations sur un drone. En ce sens, bien qu'on ait défini des objectifs, le produit final a été constamment redéfini selon les possibilités qui s'offrent à nous ; un poster de présentation et vidéo clip de ce qui aura été réalisé est toutefois prévu. Le gros enjeu du projet sera de pouvoir manipuler le drone à l'aide du kit de développement logiciel [1] fourni par le constructeur ; le modèle du drone utilisé est Parrot BEBOP 2 [2].

Nous avons découpé la phase de développement en 3 grandes parties que nous présenterons plus en détails dans ce rapport :

1. Développement du jeu (client)
2. Communication client - serveur
3. Communication serveur - drone

Sommairement, nous allons développer un environnement dit client-serveur. En effet, les joueurs, à partir de leur application mobile, représentent les clients et se connecteront à un serveur hébergé sur un ordinateur dont le rôle sera de centraliser les scores des joueurs. A partir de ces données, le serveur communiquera avec le drone pour le piloter vers l'un des joueurs.

1. SDK : Software Development Kit
2. `https://www.parrot.com/fr/Drones/Parrot-bebop-2`

Deuxième partie

Développement

Chapitre 1

Client - Serveur

La première chose qu'il fallait faire était de mettre en place le système de communication client-serveur et d'établir un protocole. Pour des raisons de simplicité nous avons opté pour un protocole textuel, dont la définition est la suivante :

Pour identifier d'où proviens les paquets tout les message (sauf le premier message "CONNECT pseudo") seront structurés par le client comme ceci :

```
Paquet = [idClient] [MESSAGE]
```

- **Connection** :
  Client envoie :
  - `CONNECT pseudo:string`
    Premier message envoyé au serveur.

  Serveur envoie :
  - `CONNECTOK`
    S'il reçoit ce message ce cas Client envoie : READYRECEIVE Cela permet au serveur d'attendre que le client ait bien initialisé sa socket (s'il ne reçoit pas ce message il renvoie le CONNECTOK et au bout de 4 tentatives il supprime le client)
  - CONNECTBAD Renvoyé si l'indentifiant est déjà pris.
- **Initialisation/gestion de la partie** :
  Client envoie :
  - `AULIST`
    Demande au serveur la liste des utilisateurs

  Serveur envoie :
  - `ULIST nom1 nom2 ...`
    Renvoie la liste des utilisateurs au client

  Client envoie :
  - `STARTGAMEOK`

  Serveur envoie :
  - `STARTGAME nbsec:int`
    Envoyé quand tout les clients de la partie ont envoyé STARTGAMEOK. La partie commence pour le client et le serveur au bout des nbSec

— **Partie Jeu** :
Client envoie :
— `SCORETICK score:byte`
Par exemple toutes les 500 ms. Met à jour le score du joueur côté serveur, et leur avancement relatif par rapport a la piste (si jeu musical)
— **Partie Fin** :
Serveur envoie :
— `GAMEEND pseudoWinner:string`
Envoyé lorsqu'un gagnant a été décidé, plusieurs idées pour définir les règles : Plus on approche de la fin de la piste, plus le drone va avancer vers le jouer au meilleur score. Ou bien, sans prendre en compte la fin de la piste (peut finir avant, ou reboucler sur la piste mais plus vite) demande plus d'équilibrage sur le gameplay. Déconnection
Client envoie :
— `DISCONNECT`
Si jamais le client n'est pas déconnecté, nettoyer les clients qui ne jouent plus depuis un certain temps.

Chapitre 2

Jeu mobile

Chapitre 3

Serveur - Drone

Troisième partie

Conclusion

Quatrième partie

Bibliographie