

RockAn'Dron

A.FERNANDEZ & S.UNG

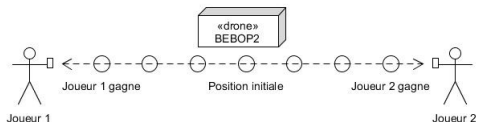
Projet STL
encadré par V.BOTBOL & G.ZIAT

27 mai 2017



Architecture Client-Serveur avec commande d'un drone :

- Programmation sur un drone
- Réalisation d'un jeu mobile
- Architecture Client-Serveur



BUT : Contrôle à distance du drone

⇒ Démonstration publique à la *Fête de la Science*

DRONE PARROT BEBOP 2



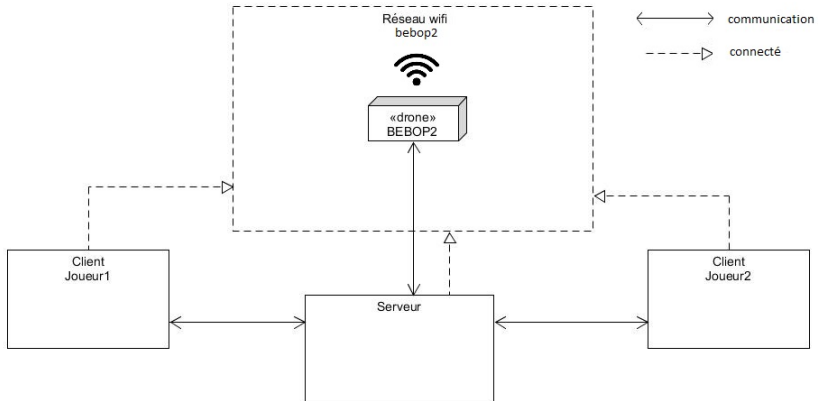
- Poids : 500g
- Autonomie : 25min
- Diffuse un réseau Wi-Fi sur 300m
- Application mobile *Free Flight Pro*

ANDROID SDK

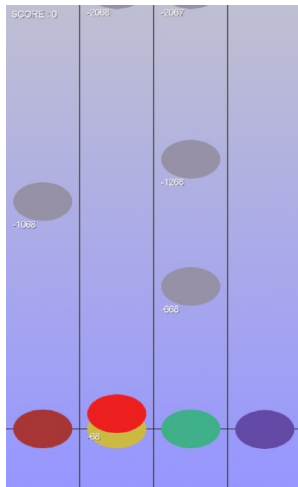


- Java
- Framework *libGDX*

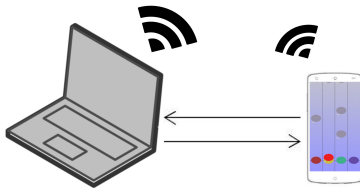
Réseau de communication



- Jeu de rythme
- Prototype du client sous forme d'une application *Android*
- 4 zones actives
- Attente des joueurs dans un salon



- Protocole de communication textuel
 - ① Connexion
 - ② Initialisation et lancement de la partie
 - ③ Envoi périodique des scores
- Protocole UDP vs TCP



- Analyse et exploitation de la SDK (ARSDK Parrot)
- Utilisation de l'API C du SDK
- Mouvements opérationnels

Exemples de primitives

```
//Décollage
```

```
deviceController->aRDrone3->sendPilotingTakeOff(deviceController->aRDrone3);
```

```
//Prise de photo
```

```
deviceController->aRDrone3->sendMediaRecordPicture(deviceController->aRDrone3, 0);
```

```
//Angle du drone
```

```
deviceController->aRDrone3->setPilotingPCMDPitch(deviceController->aRDrone3, 50);
```


- Conditions de tests délicates
 - Disponibilité du drone
 - Sécurité
- Gestion propre de la position du drone complexe
 - Facteurs extérieurs
 - Limites matérielles

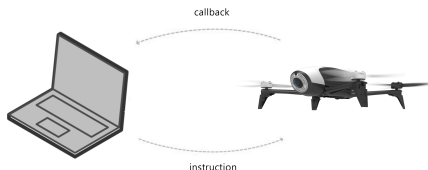
Ce qui a été fait . . .

- Client jeu mobile fonctionnel
- Serveur opérationnel
- Programme de pilotage

. . . ce qu'il reste à faire :

- Assembler le serveur et le programme de pilotage
- Génération de niveau à partir d'un fichier audio

- Assurer la position du drone
 - Système d'asservissement



- Comportement du drone durant la partie
 - Contrôler les allers-retours
 - Plus de 2 joueurs ...

Merci de votre attention