

# *RockAn'Dron*

A.FERNANDEZ & S.UNG

Projet STL  
encadré par V.BOTBOL & G.ZIAT

25 mai 2017

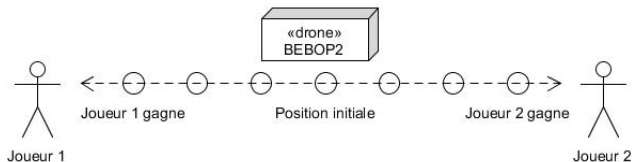


1 Introduction

2 Conception

3 Conclusion

- 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



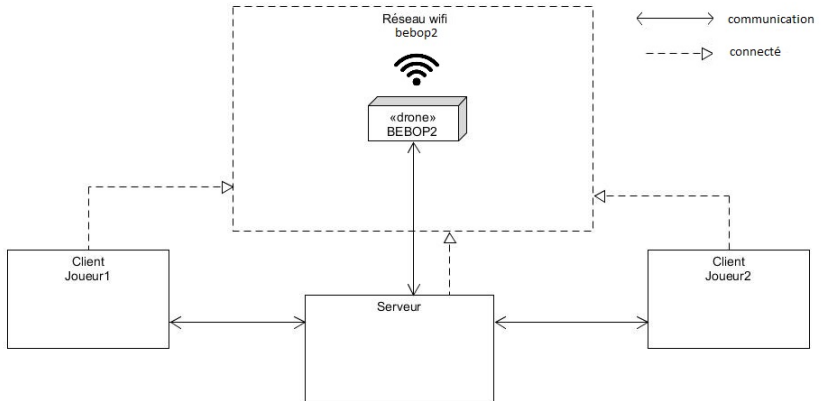
- Poids : 500g
- Autonomie : 25min
- Diffuse un réseau Wi-Fi sur 300m
- Application mobile *Free Flight Pro*

## ANDROID SDK

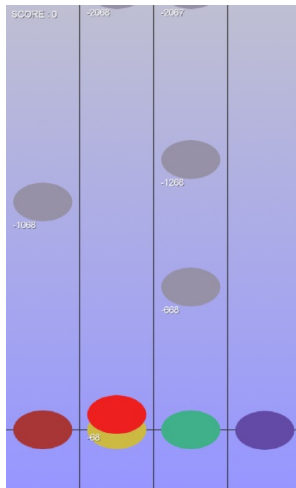


- Java
- Framework *libGDX*

# Réseau de communication



- Jeu de rythme
- Programme simple ( $\approx 1$  semaine)
- 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)
- Programme C
- Mouvements opérationnels

## Exemples de primitives

```
deviceController->aRDrone3->sendPilotingTakeOff(deviceController->aRDrone3);  
deviceController->aRDrone3->setPilotingPCMDFlag(deviceController->aRDrone3, 1);  
deviceController->aRDrone3->setPilotingPCMDPitch(deviceController->aRDrone3, 50);
```



- Conditions de tests délicates
- Gestion propre de la position du drone complexe

Ce qui a été fait ...

- Client jeu mobile fonctionnel ( $\approx 1400$  lignes de code)
- Serveur opérationnel ( $\approx 1000$  lignes de code)
- Programme de pilotage ( $\approx 1200$  lignes de code)

Ce qu'il reste à faire ...

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

- Comportement du drone durant la partie
- Assurer la position du drone

Merci de votre attention