

RockAn'Dron

A.FERNANDEZ & S.UNG

Projet STL
encadré par V.BOTBOL & G.ZIAT

23 mai 2017



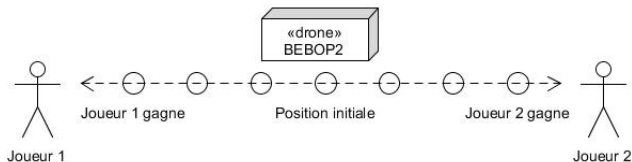
1 Introduction

2 Conception

3 Conclusion

Introduction

- 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
antenne Wi-Fi jusqu'à 300m
Application *Free Flight Pro*

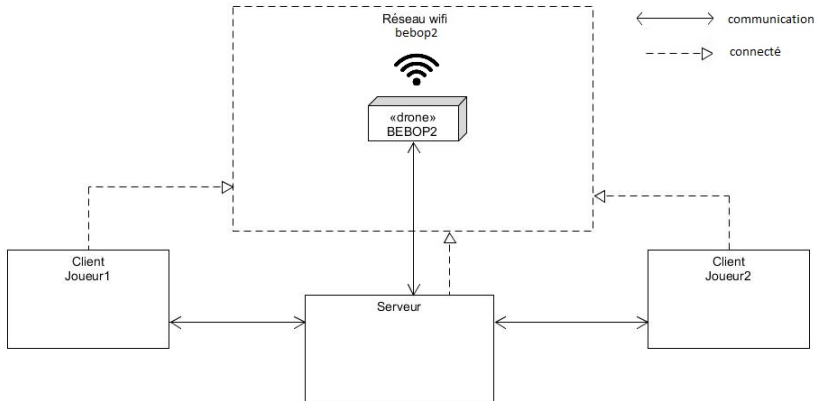
ANDROID SDK



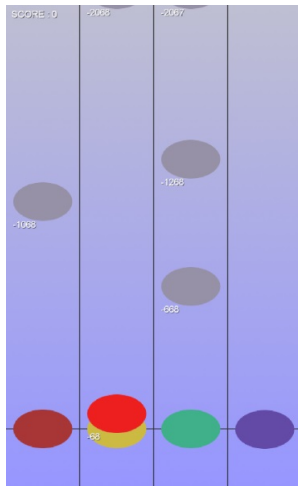
Java
framework *libGDX*

Conception

Réseau de communication



- Jeu de rythme
- Programme simple (≈ 1 semaine)
- 4 zones actives
- Toucher les notes en rythme
- Fenêtre de temps pour valider les points



- Protocole de communication textuel
 - ① Connexion
 - ② Initialisation et lancement de la partie
 - ③ Envoi périodique des scores
- Protocole UDP vs TCP
- Attente des joueurs dans un salon ou *lobby*
- Mise en place en $\approx 1 - 2$ semaine(s)

- Analyse et exploitation de la SDK (ARSDK Parrot)
- Programme C
- Mouvements opérationnels
- Composant inachevé ...

Exemples de primitives

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

- Problème du réseau Wi-Fi
- Faiblesses du protocole UDP
- Conditions de tests délicates
- Gestion propre de la position du drone complexe

Conclusion

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

- 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