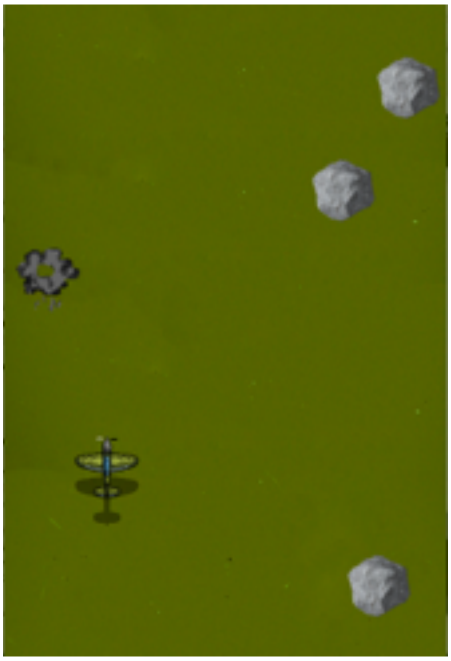
PANG BO

REPORT OF PROJECT OF CREATIVE CODING

INTRODUCTION

This project's name is MoPlane.



In this project I used processing to implement a plane game which could be controlled by user's voice. The user need to sound “da” to fire a bullet and “qiu~~” to launch a missile to destroy the aerolite.

BACKGROUND

At the first course after I watching the professor's demo of processing, I was shocked by using such a easy way to combine image, audio, control function together to implement a program without considering about if it is MCV, if it uses a good engine, etc the problems we always think about before when we want to make a software. Here we don't need to think about any deep programming problem, what we need is only our imagination.

So this time I decide to combine image , sound together. And also I want to make it interesting. During the professor's demo, he used voice control to “shock “ those point. It inspired me to think maybe I could do a voice-control game by maybe more like a real game.

So finally the idea making a plane game came into my mind.

WHY DID I MAKE THIS CHOICE ?

Traditionally, in most games designed for smartphones, people only need to use their fingers to control game.In my game, you can use both your hands to control the plane, but also your voice if you want to shoot: I thought it would allow the users to have a new experience with game, and have more interactions with it.
Plus, it also allows people to get familiar with some essential Chinese vocabulary words, such as “Da” (meaning Bullet) or “Qiu” (meaning Missile)
There aren’t many games that use such a concept, and basing my game on such a concept is very creative in my opinion.

TECHNICAL ISSUES AND HOW TO SOLVE THEM

Classes: I make each object in the game as a class including their position, speed, acceleration, and their behaviours(like plane, it can fire a bullet, launch a missile ,etc).

Voice Control: I used ddf class to analyse the sound and java.util to calculate time. For example, when the input sound higher than 0.001, it means there is an effective sound and it will trigger a timer to start. When it’s lower than 0.001, it means the sound is over. After that, the program calculate the duration of the sound and if it’s between 0.05s to 0.7s, the program will treat this as a fire command and tell plane to fire.

Collision detection: The program will run all the time to check if a missile or bullet is close enough to a target.

Missile control: The missile could find their nearest target, lock to it and destroy it automatically. Every time a new missile was launched, it will try to find their nearest target, calculate their distance, and add a certain acceleration to this target until it has destroyed.

Multiple treads: Since the program is alway refreshing its image, it’s not efficient to put all its functions together. Here here I used Java’s multiple treads to run these functions in different treads, like voice control, collision detection.

CONCLUSION

To sum it up, this project was extremely fulfilling: I discovered another way of coding that changed a lot compared to my habits.

Thanks to the tools I used, I now know that there are many simple ways to create very cool applications and I might not have thought of them all on my own.

This project also helped me understand more deeply how vocal recognition worked : it will without a doubt be useful to me, as I intend to work in a somewhat related field later on.