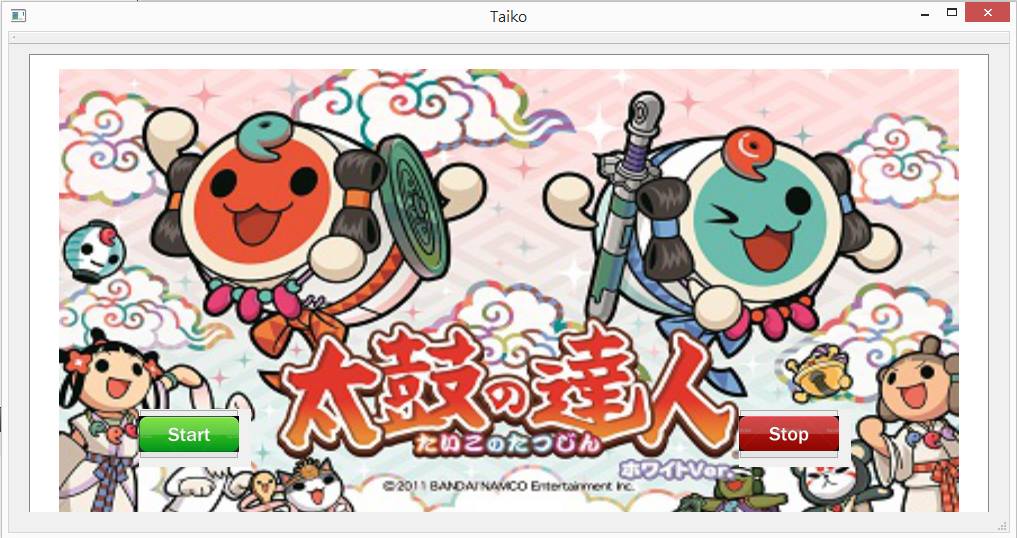
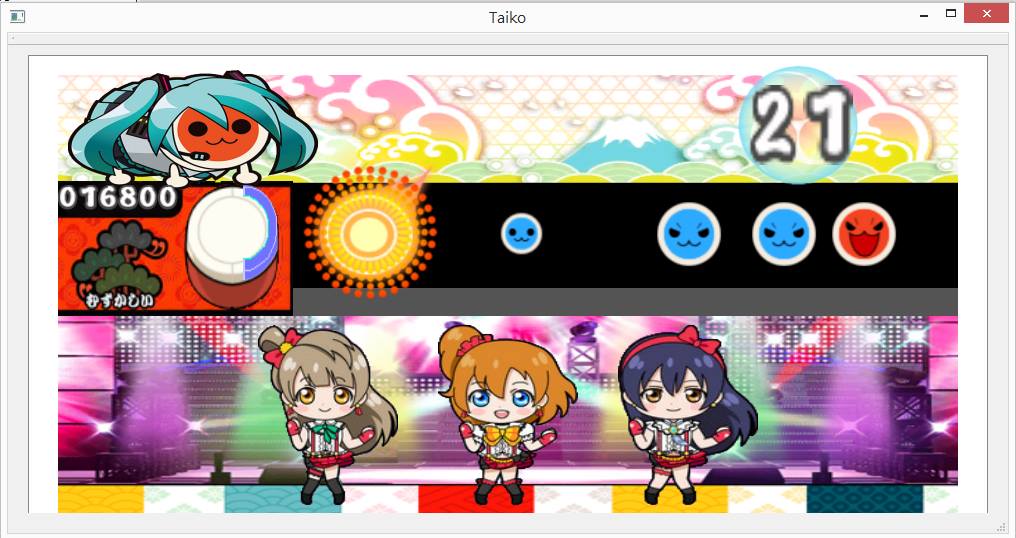
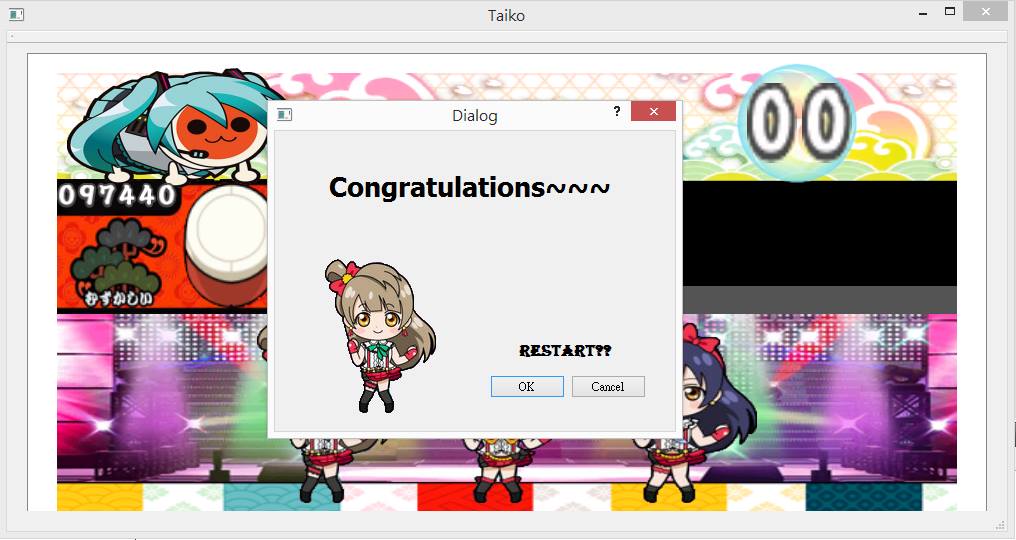
Taiko master Report

C14036227 王晉鋒

**Program execution snapshot:**







**UML class diagram:**

(僅列Scene這個class,其他的有resultBoard(QDialog), Taiko(QMainWindow))

|  |
| --- |
| Scene |
| +start: QPushButton\*  +stop:QPushButton\*  +pix:QPixmap  +background:QLabel\*  +drum\_main:QLabel\*  +core:QLabel\*  +bottom:QLabel\*  +character:QLabel\*  +dancer1:QLabel\*  +dancer2:QLabel\*  +dancer3:QLabel\*  +countdown:QLabel\*  +left\_red:QLabel\*  +right\_red:QLabel\*  +left\_blue:QLabel\*  +right\_blue:QLabel\*  +numone:QLabel\*  +numten:QLabel\*  +numhen:QLabel\*  +numtho:QLabel\*  +numttn:QLabel\*  +numtop:QLabel\*  +num\_count\_ten:QLabel\*  +num\_count\_one:QLabel\*  +drums\_sign[10000]:Drums\_sign   |  | | --- | | Drums\_sign | | +drums\_array:QLabel\* | | +pattern:int |   +drums:QLabel\*  +explosion:QLabel\*  +block:QLabel\*  +clk1:QTimer\*  +clk2:QTimer\*  +clk3:QTimer\*  +clk4:QTimer\*  +clk5:QTimer\*  +clk6:QTimer\*  +clk7:QTimer\*  +clk8:QTimer\*  +mode:int  +timecount:int  +timeincre:int  +timeexplosion:int  +randdrum:int  +randtime:int  +init:int  +b:int  +yourScore:int  +tryagain:int |
| +Entry()  +bgChange()  +bgBack()  +changeTimeNum()  +selectDrums(int b)  +judgeaccr\_red()  +judgeaccr\_blue()  +addBigScore()  +addSmallScore()  +keyPressEvent(QKeyEvent\* event)  Slots:  -changeBg()  -move()  -countingDown()  -generateDrums()  -incre0()  -incre1()  -incre2()  -incre3()  -incre\_explosion() |

**README: (instructions, program structure)**

1. press start to play, press stop to exit
2. press“A”,”S” if the red drums runs through the circle area, press”Q”,”E” if the blue drums runs through the circle area

when succeed, you get points and big size drums can give you double points.

(special functions: the explosion image will show if you successfully hit, and when you press the key, the red or blue light will show on the main drum)

1. Be careful!! The speed might change during the play, enjoy☺
2. After the countdown to zero, it will ask you to restart or not, if yes, restart the game; if no, you stay at the result scene.

Briefly explain the program structure:

I use a structure of array to save a integer(the type of drums picture)and a QLabel(to generate new drums at a random time)

Move function occurs when the timer starts, and move all the items of the array to move(setGeometry).

For restart I use a QDialog and to clear the original class and reset all the variables and connections.