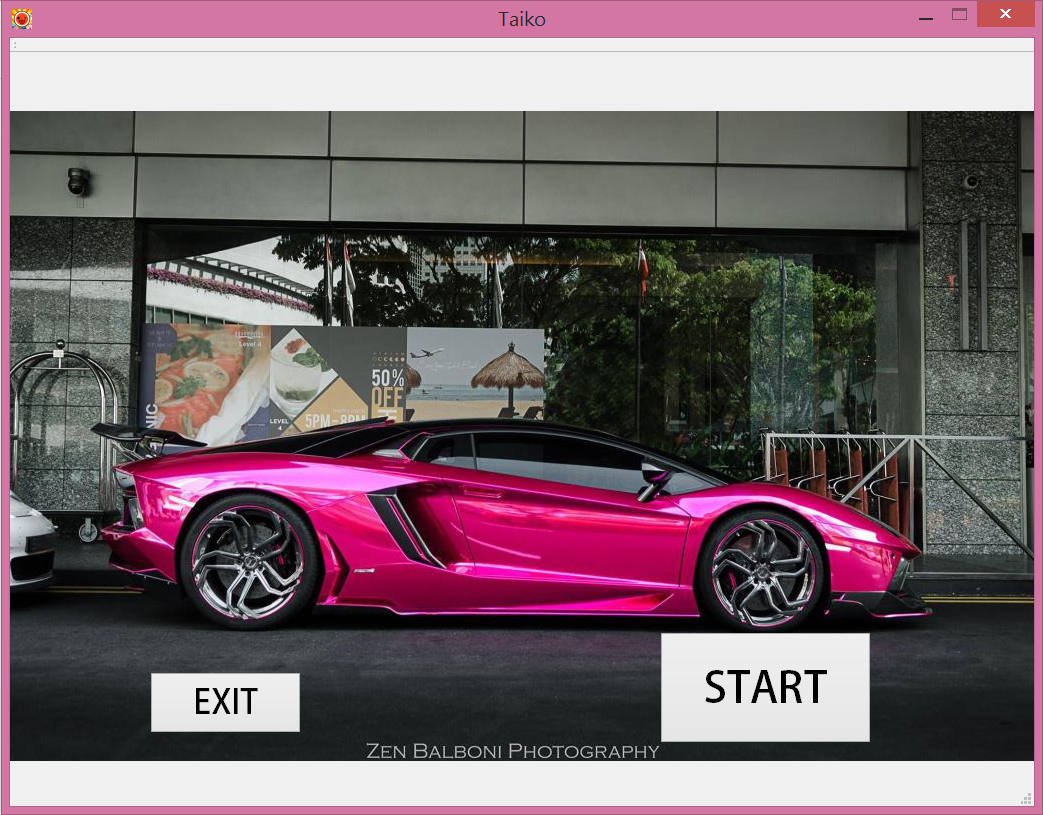
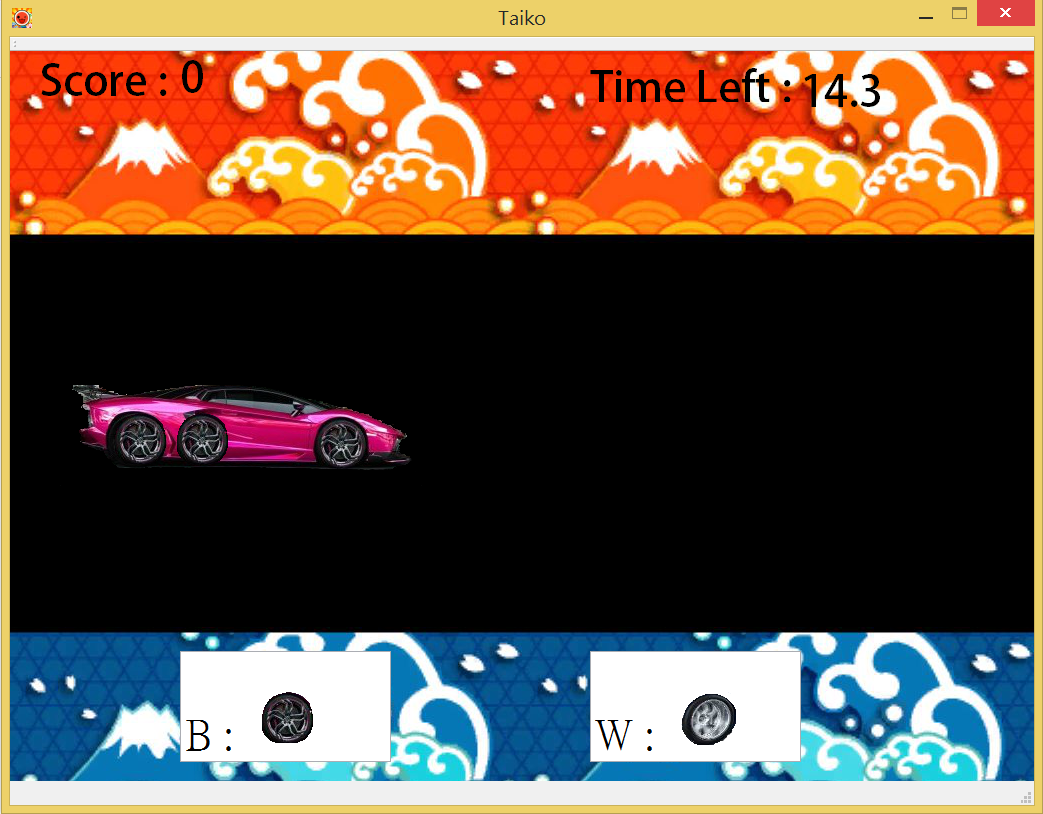
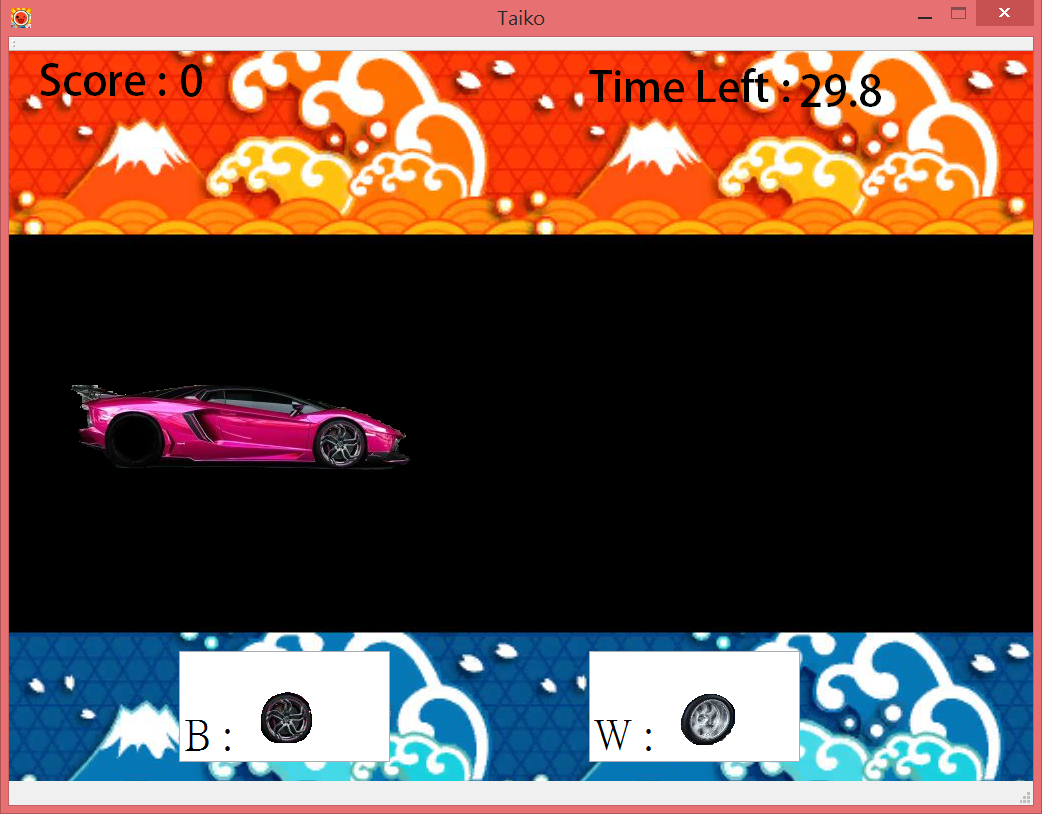
UML Class Diagram :

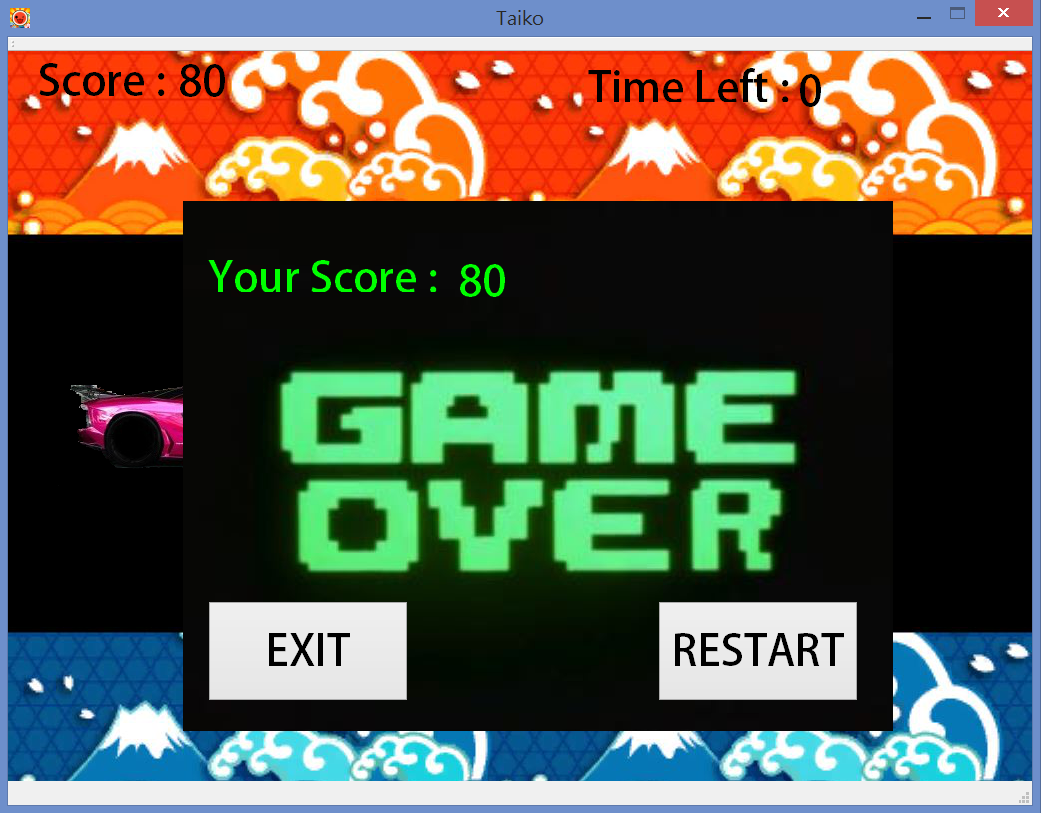
|  |
| --- |
| MainWindow |
| +QTimer \* : timer  +QTimer \* : time\_call  +QTimer \* : restarttime  +QTimer \* : time\_call\_2  +t : double  +C : int  +tire1[15] : int  +tire2[15] : int  +score : int  +finalscore : int  +MARK : int  +combo : int  +test: int  -Ui::MainWindow \* : ui  -QMediaPlayer \* : clicksong  -QMediaPlayer \* : backsong |
| <<constructor>>+ MainWindow  + Remake():void  - Time():void  - on\_START\_clicked():void  - on\_EXIT\_clicked():void  - time\_x():void  - init\_pos():void  - position():void  - *keyPressEvent*(press : QKeyEvent \*): void  - *keyReleaseEvent*(release : QKeyEvent \*): void |

Screen shot :









How to play ?

First, click the START button, and click B to eliminate the Black wheels, click W to eliminate the White wheels .When time’s up, you can click the RESTART button to restart the game, or click the EXIT button to end the game.

Program architecture :

First, set the labels and buttons. Show what you want to see at first screen. Second, after clicking START, hide the first screen, and show the game. Third, use the array to set the wheels, and make them move(use a function to call it repeatly), and then, set the range to eliminate the wheels. Last, count the scores and time. Last, show your scores and the screen that you can choose to RESTART or EXIT the game.