1. UML class diagram

|  |
| --- |
| BlueDrum |
| - pressSound:QMediaPlayer\* |
| <constructor>+BlueDrum()  +keyPressEvent(event: QKeyEvent \*)  +move() |

|  |
| --- |
| RedDrum |
| - pressSound:QMediaPlayer\* |
| <constructor>+RedDrum()  +keyPressEvent(event:QKeyEvent\*)  +move() |

|  |
| --- |
| Button |
| - text:QGraphicsTextItem\* |
| <constructor>+Button(name:QString, parent:QGraphicsItem\*)  +mousePressEvent(event: QGraphicsSceneMouseEvent\*)  + hoverEnterEvent(event: QGraphicsSceneHoverEvent\*)  + hoverLeaveEvent(event: QGraphicsSceneHoverEvent\*); |

|  |
| --- |
| MyRect |
|  |
| <constructor>+MyRect()  +spawn() |

|  |
| --- |
| MyTimer |
| +mytime:Integer |
| <constructor>+MyTimer()  +decrease() |

|  |
| --- |
| Game |
| + scene: QGraphicsScene \*  +player: MyRect \*  + score: Score \*  + target: Target \*  + mytimer: MyTimer \*  +music: QMediaPlayer\*  - timer: QTimer \*  - countdown\_timer: QTimer \* |
| <constructor>+Game()  +displayMainMenu()  +gameover()  + displayGameOverWindow()  + stopSpawn()  + stopCountdown()  +start()  +restartGame()  - drawPanel(x:Integer, y:Integer, width:Integer, height:Integer, color:Qcolor, opacity:Double) |

|  |
| --- |
| Score |
| +score:Integer |
| <constructor>+Score(parent:QGraphicsItem\*)  +increase()  +getScore():Integer |

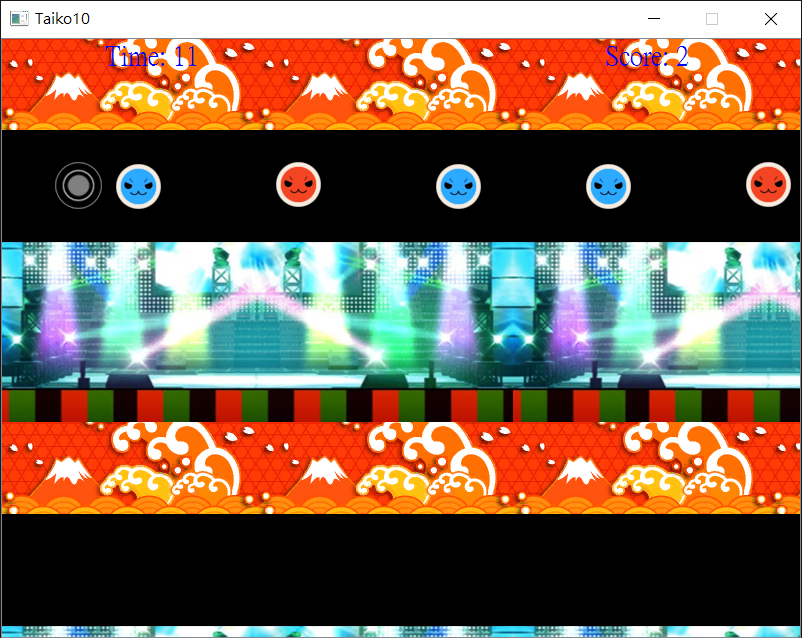
|  |
| --- |
| Target |
|  |
| <constructor>+Target() |

1. Screen shot

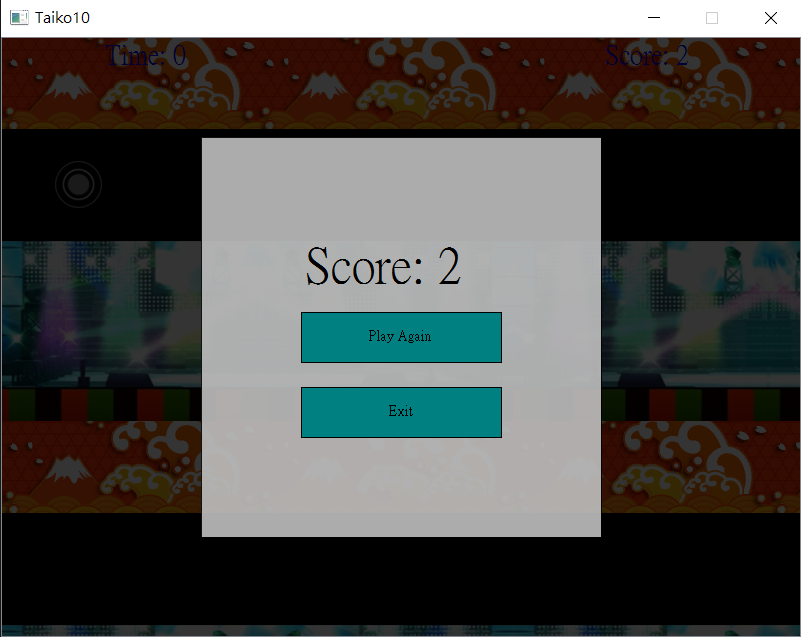
(1)start



(2)play



(3)gameover



1. How to play

(1)Press J to eliminate red drum.(2)Press K to eliminate blue drum.(3)You can eliminate the drum by pressing J and K only when the drum near the target.

1. Bonus

Music:background and eliminate drums

1. Program architecture

link

Taiko

compile

main.o

target.o

score.o

mytimer.o

game.o

button.o

red\_drum.o

blue\_drum.o

myrect.o

myrect.cpp

Preprocess(#include)

red\_drum.h

game.h

game.cpp

mytimer.cpp

score.cpp

target.cpp

main.cpp

button.cpp

red\_drum.cpp

blue\_drum.cpp

target.h

score.h

mytimer.h

button.h

blue\_drum.h

myrect.h