Project 2 Report – Taiko no Datsujin

F74046234

資訊一甲 108

吳貞頤

# UML class diagram:

dialog_howto
-Ui::ui:dialog_howto*
<<constructor>>+explicit dialog_howto(parent:QWidget* = 0) +~dialog_howto() -void on_pushButton_clicked():void

dialog_score
-Ui::ui:dialog_score*
<<constructor>>+explicit dialog_score(parent:QWidget* = 0) +~dialog_score() -on_pushButton_clicked():void

dialogaboutris
-Ui::ui:DialogAboutris*
<<constructor>>+explicit DialogAboutris(parent:QWidget* = 0); +~DialogAboutris() -on_pushButton_clicked():void

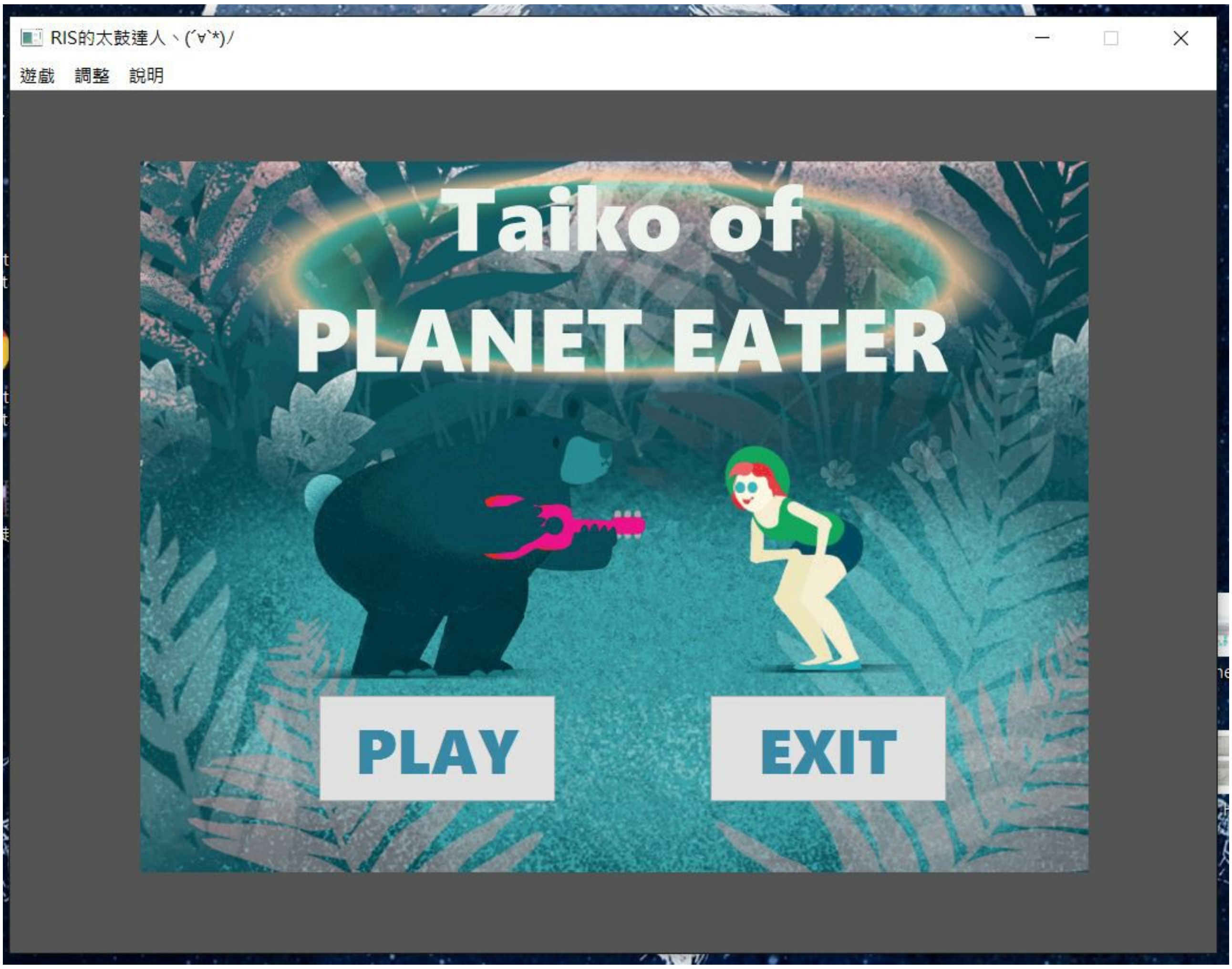
drums
-whichdrum:int -s:QMediaPlayer*
<<constructor>>+drums() +keyPressEvent(event:QKeyEvent*):void +move():void

mainwindow
-Ui::ui:MainWindow* -movie:QMovie*
<<constructor>>+explicit MainWindow(parent:QWidget* = 0); +~MainWindow(); -on_pushbutton_play_clicked():void -on_pushbutton_exit_clicked():void -on_action_quit_triggered():void -on_action_aboutris_triggered():void -on_action_howto_triggered():void

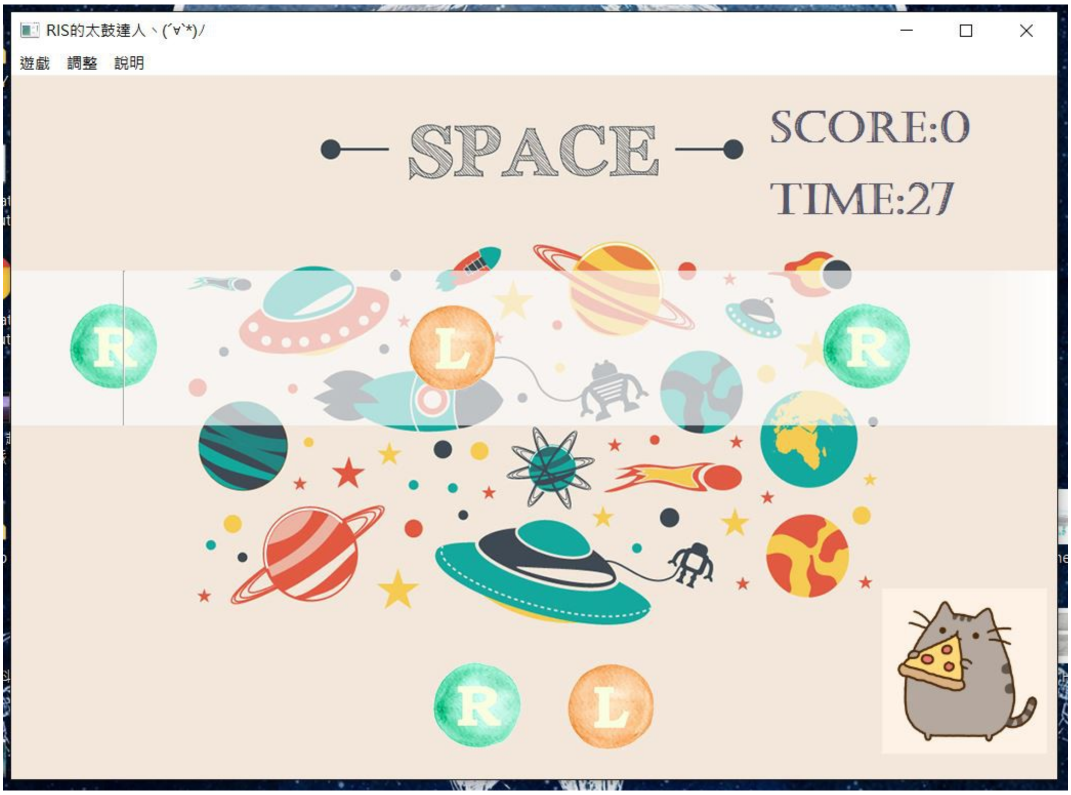
playwindow
-Ui::ui:PlayWindow* -scene:QGraphicsScene* -view:QGraphicsView -alltimetext:QString -timetext:QString -count:QTime -many:int -scoretext:QString -transcore:QString
<<constructor>>+explicit PlayWindow(parent:QWidget* = 0) +~PlayWindow() +showScore(acc:int = 0):void +displayscore:int +displaytime:int +generate_item():void +delay():void +game_end():void -showTime():void -on_action_howto_triggered():void -on_action_quit_triggered():void -on_action_ris_triggered():void

# **Screen shot:**

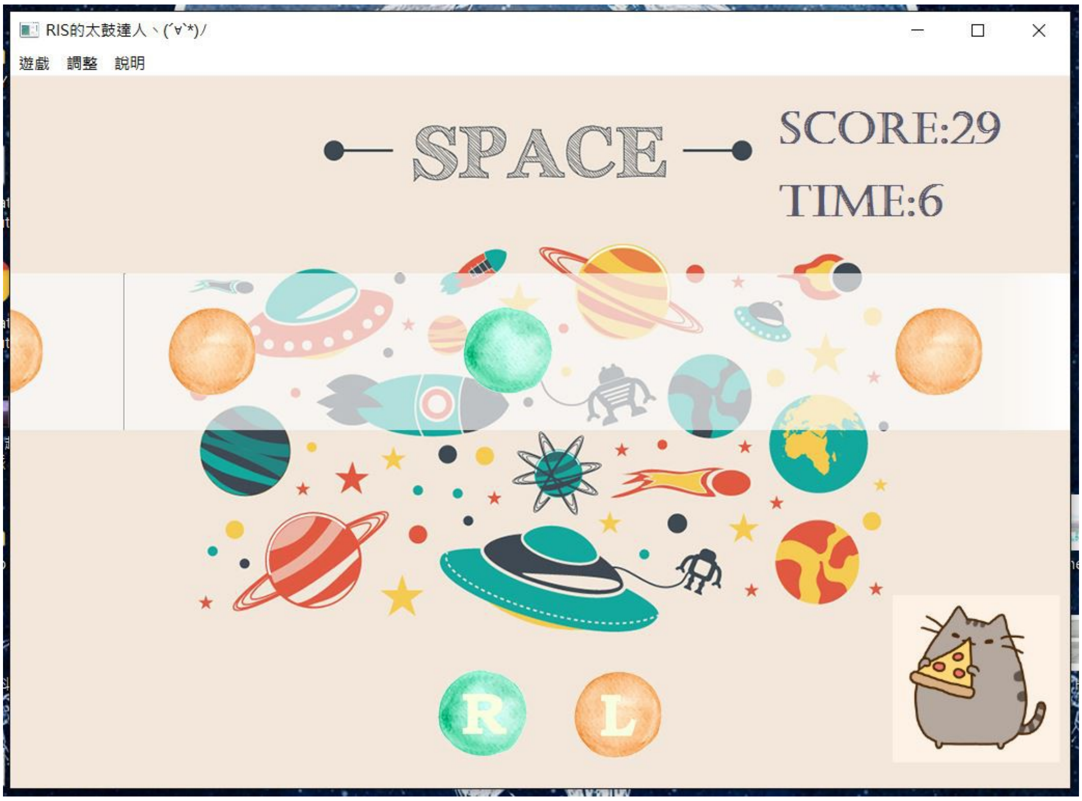
(the mainwindow)



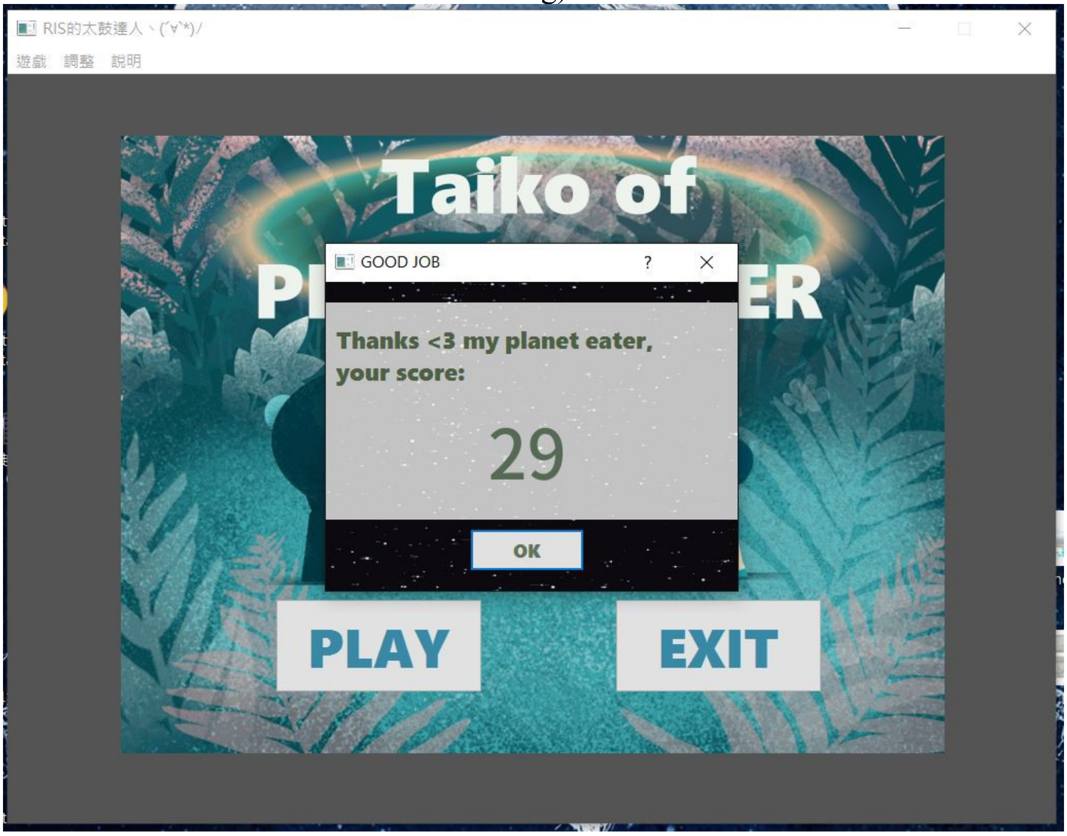
(the playwindow)(The first 5 seconds of the game, there are drums with words to give a hint of how to play the game)



(the playwindow)(After 5 seconds, there's no hint on the drums. But there are still hints at the bottom of the window)



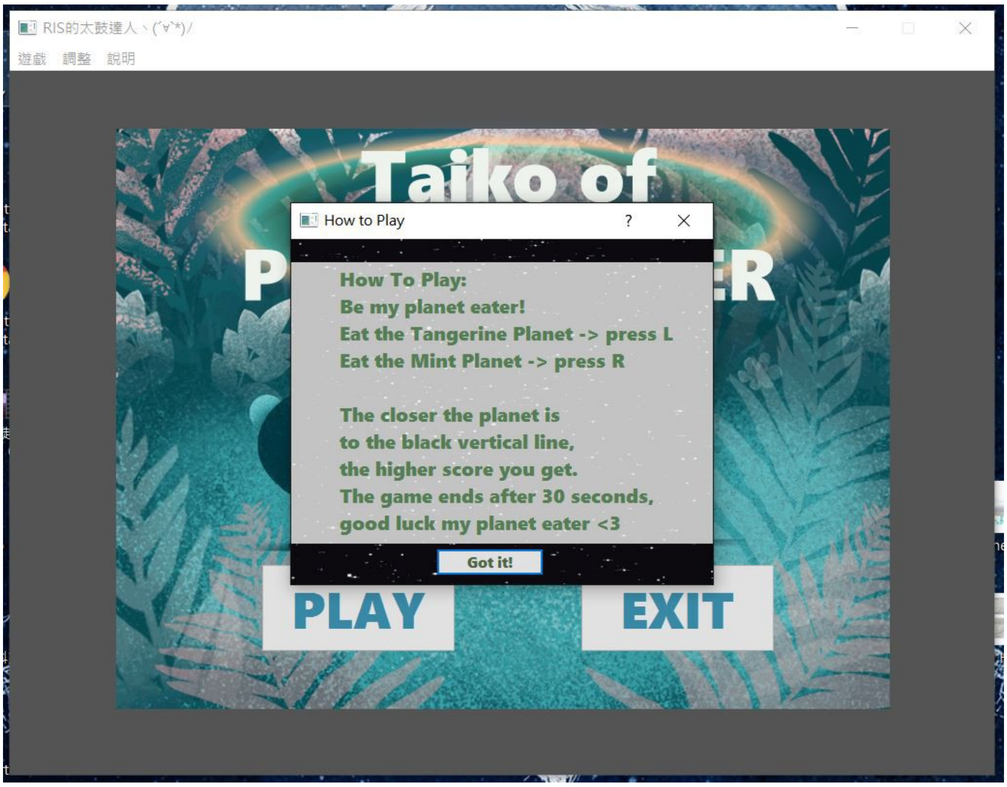
(back to the mainwindow and show the score dialog)



Thanks <3 my planet eater,
your score:

29

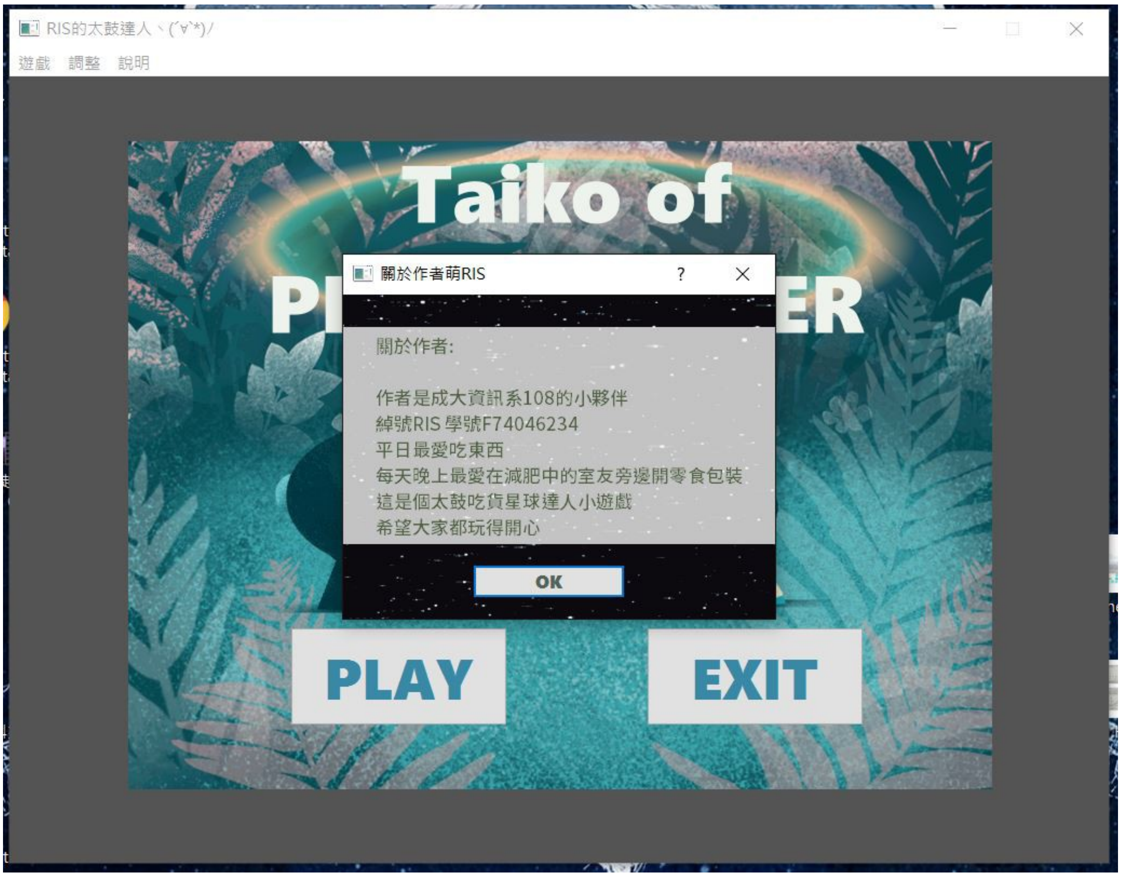
(the dialog of the game rules)



How To Play:
Be my planet eater!
Eat the Tangerine Planet -> press L
Eat the Mint Planet -> press R

The closer the planet is
to the black vertical line,
the higher score you get.
The game ends after 30 seconds,
good luck my planet eater <3

(the dialog of the information about the author)



關於作者:

作者是成大資訊系108的小夥伴
綽號RIS 學號F74046234
平日最愛吃東西
每天晚上最愛在減肥中的室友旁邊開零食包裝
這是個太鼓吃貨星球達人小遊戲
希望大家都玩得開心

# how to play & details of game rules:

In the first window (mainwindow), there are two pushbuttons.
Click 'PLAY' to start the game (playwindow).
Click 'EXIT' to close the game.

In the game window (playwindow), there will be planets (drums) passing through. When the planets are near the black vertical line on the left of the window,
press 'R' on the keyboard to delete green planets,
press 'L' on the keyboard to delete orange planets.
If the middle of planets are closer to the verticle line, you can get higher score (successful deleting one planet will get points range from 0~12). The score you get is shown at the top right of the window.
After 30 seconds (remain time is also shown at the top right of the window), the game will end automatically. Then the dialog will pop out, showing the result of the game (the score you get). And you will be back to the start window (mainwindow).

In both windows above, there are actions in the menu at the top of the windows.

Click '我不玩了' to close the game at any time.

Click '要怎麼玩' to view the dialog of the game rule.

Click '關於作者萌 RIS' to view the dialog of information about the author of the game.

# program architecture:

main.cpp shows the mainwindow (there is also a global pointer which points to it)

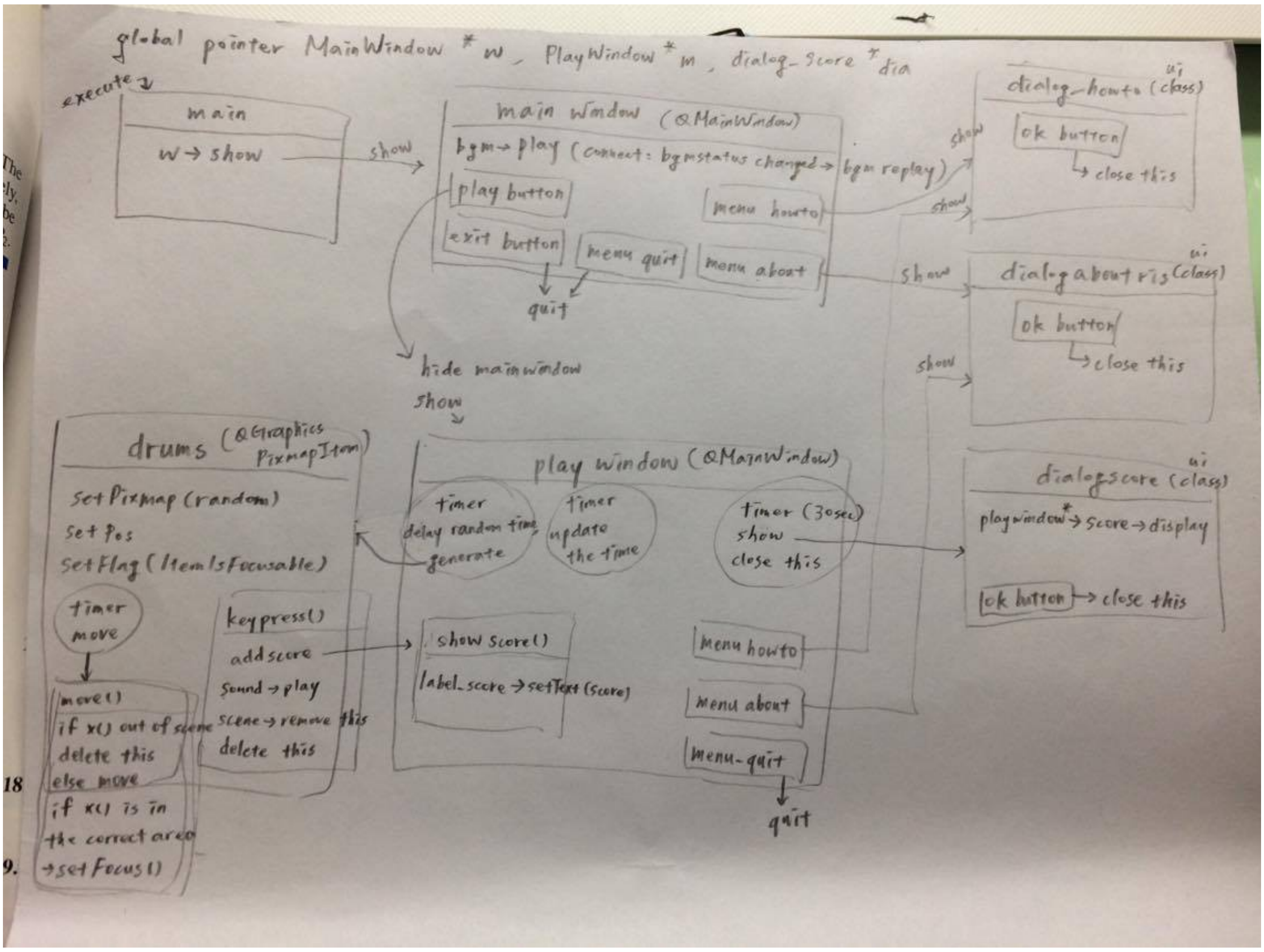
click the 'play' button in the mainwindow, it will generate a playwindow (there is also a global pointer pointing to it) and hide the mainwindow

the actions in the menu on both mainwindow and playwindow will show the different dialog of 'game rules' and 'about the author', or quit the game.

Once the playwindow object was generated, there are different timer controlling the functions of 'generate items', 'display time', and 'end the game'.

The drum is a class. The object of the drum will be generated when the timer in playwindow sends out a timeout signal. In the constructor of the drum, it would randomly set a pixmap (there are four pixmaps). And the drum will always move to the left which is controlled by the timer in itself. There is also a keypressevent, when a specific key is pressed, the drum will be removed from the scene, and delete itself.

When the game ends, the playwindow will close and show the dialog_score, and show the original startwindow.


global pointer MainWindow *w, PlayWindow *m, dialog_Score *dia
execute
main
w → show
show
main window (QMainWindow)
bgm → play (connect: bgmstatus changed → bgm replay)
play button
menu howto
exit button
menu quit
menu about
quit
hide mainwindow
show
show
show
show
show
dialog_howto (class)
ui
ok button
close this
dialogabout ris (class)
ui
ok button
close this
drums (QGraphics PixmapItem)
setPixmap (random)
setPos
setFlag (ItemIsFocusable)
timer move
move()
if x() out of scene
delete this
else move
if x() is in the correct area
→ setFocus()
keypress()
addscore
Sound → play
scene → remove this
delete this
play window (QMainWindow)
timer delay random time generate
timer update the time
timer (30sec) show close this
show Score()
label_score → setText (score)
menu howto
menu about
menu-quit
quit
dialogscore (class)
ui
playwindow* → score → display
ok button → close this