project2—太鼓達人~

1. UML class diagram

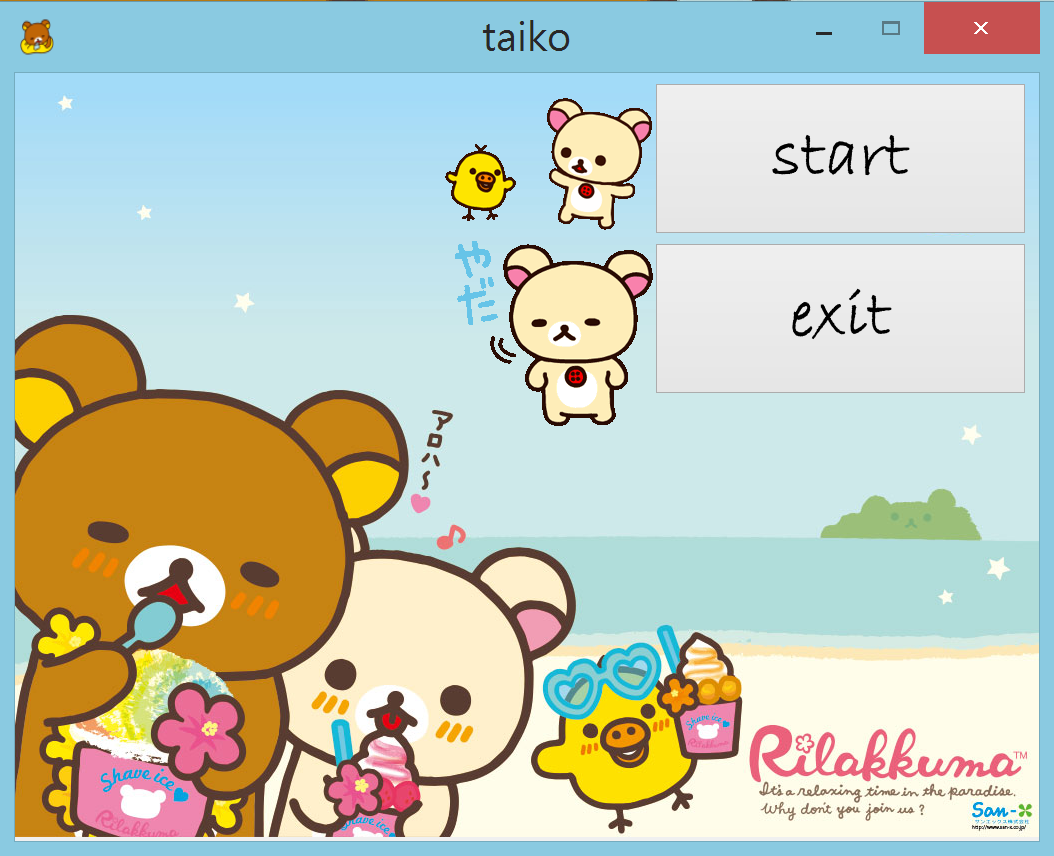
|  |
| --- |
| mainwindow |
| +music : QMediaPlayer \*  -g : GameWindow \*  -ui : Ui |
| <<constructor>>  -on\_pushButton\_clicked()  -on\_pushButton\_2\_clicked() |

|  |
| --- |
| gamewindow |
| +firstScore : Interer  +count : Interer  +R: Interer  +Scene : scene \*  +timer : QTimer \*  +music \_gaming : QMediaPlayer \*  +movie\_3 : QMovie \*  +movie\_4 : QMovie \* |
| <<constructor>>  -showTime()  -gameoverbg()  -on\_pushButton\_clicked()  -on\_pushButton\_2\_clicked()  -on\_pushButton\_3\_clicked()  -on\_pushButton\_4\_clicked() |

|  |
| --- |
| scene |
| +pillow : itemmm \*  +heart : itemmm \*  +boom : itemmm \*  +movingHead[9] : QGraphicsPixmapItem  +moveTimer : QTimer \*  +bornHeadTimer : QTimer \*  +hearttimer : QTimer \*  +boomtimer : QTimer \*  +R : Integer  +originPlace[9] : Integer  +picType[9] : Integer |
| << constructor >>  -movehead()  -removeheart()  -removeboom() |

|  |
| --- |
| itemmm |
|  |
|  |

1. how to play:
2. The binginning view is like this:



->press the “exit” to quit the game.

->press the “start” button to enter the game.

b. enter the game:

After enter the game, the items(drums) will start to appear.



The four different items will running by the deep yellow rail.

Here are the introduce for the pour items:

キイロイトリ(Kiiroitori) : 以3/11的機率出現，每消掉她一次就加一分。到枕頭的時候按A消掉她。

 コリラックマ(Korilakkuma) : 以3/11的機率出現，每消掉她一次就加兩分。到枕頭的時候按S消掉她。

 リラックマ(Rilakkuma) : 以2/11的機率出現， 每消掉她一次就加三分。到枕頭的時候按D消掉她。

 張哲head : 以低機率1/11出現，只要在她到枕頭的時候按到A, S, D鍵其中任意一個，就會爆炸而gameover。所以不可以按，要讓她乖乖離開。

The game will give you 30 seconds to play. You have to get the most score in the time limit. When the items move to the target(pillow), press the correct button to delete the item and get the point. While you press the correct button in the right x’s range, there will appear a little heart, and you will get your score.



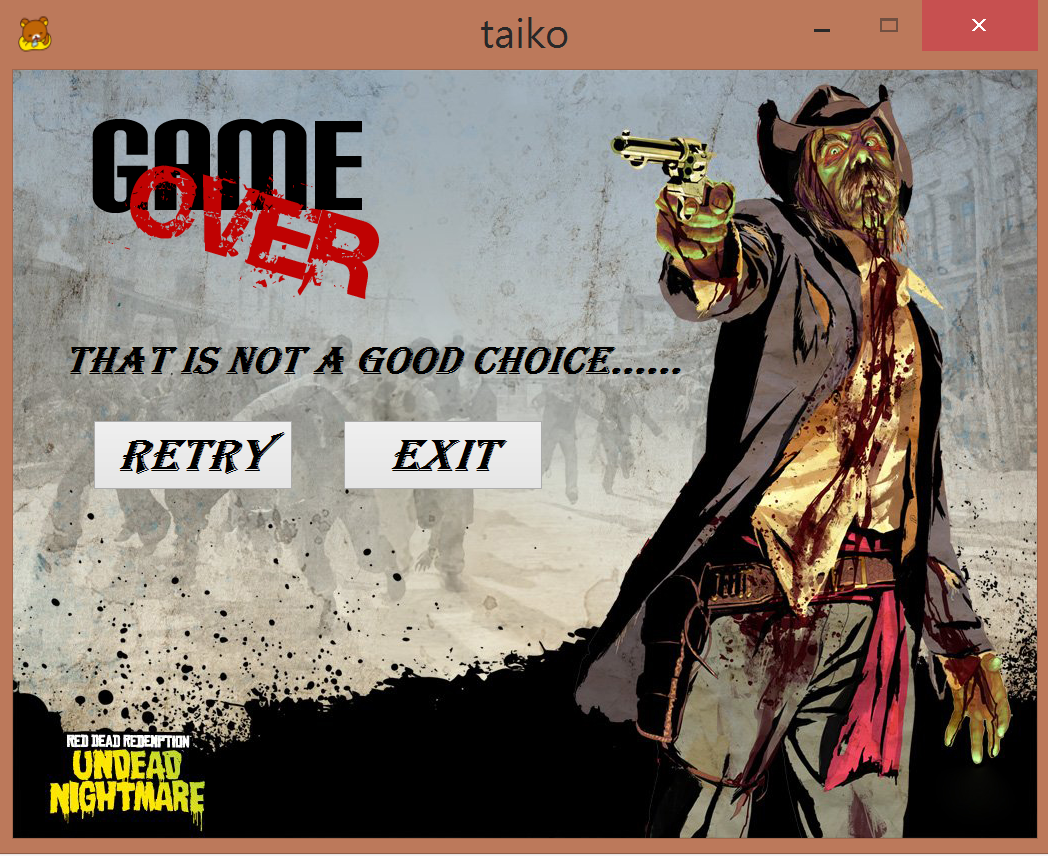
But, sometimes you may get an evil 張哲head.



If you are so nervous and press any key of A, S, D, the 張哲head will explode and every thing, such as the timer, and other items, will stop for 2 second.



And then, you dead.



However, most people can easily avoid the evil 張哲head and complete their challenge. If you completely avoid all of the 張哲head, after the timer turns into 0, the game will finish and you will get your score.

If your score is lower than 10 point, you will get a sad Rilakkuma.



If your score is between 10 and 20, then you will get an OK Rilakkuma.



And if you have great skill and your score is higher than 20, you will get a happy Kiiroitori.



And if you press the restart button, you can play again and again.



->

program architecture:

1. mainwindow : Build the beginning interface. Use ui to set the background and the push buttons. And by click the start button, the function will help to go to the gamewindow.
2. gamewindow : Build the game interface. Here does several things. First, hide some items that will be used in the end window. Then call the headArrFunc in scene.cpp to add the moving heads outside the interface. To add item, I write a scene class and do all the things that need to add item in it. Here use ui to put the timer and score. Timer use a showTime function. The function minus the count every second. When the count become 0, show the end interface that are hided at the bigining of gamewindow. The end interface has a restart button can hide the end button and reset the arguments to play again. If you click exit then the window will be closed. If you press when 張哲head is in the range, then the game over background will be showed. Press retry to reset the arguments and play again. Gamewindow also do the keypressevent. Judge the key you press and decide what to do. If you press the correct key when the items are in the correct range, the items will be setpos to the outside of the window. And the pushButton\_click() functions are also here.
3. scene : This is write to add items. Such as the moving items. Set an originPlace[9]to put the x of the movingHead[9]. Use the originPlace[9] and a timer to set position of the movingHead[9] per 5\*10^-3 second. When all the originPlace[] are lower than 0, reset them to the original place. And when you pressin the correct timing, scene will add the heart or boom. The heart will be removed after 0.5 second. The boom will be remove after 2 seconds.