

Project3

The program is Qt Gui Application

There are two modes you can choose in menu.

(You can change it during game too, but the current score will be discarded.)

Rank won't evaporate after the program terminated.

If you need to clear it, please delete the user.db file in the same directory

Environment

OS: ubuntu 14.04 LTS

gcc version: 4.8.2

QMake version: 3.0

Qt version: 5.4.0

Compilation

`qmake` `make` or

Use Qt creator to compile the project

Rule

- Step Mode:

Get scores in 10 steps

- Time Mode:

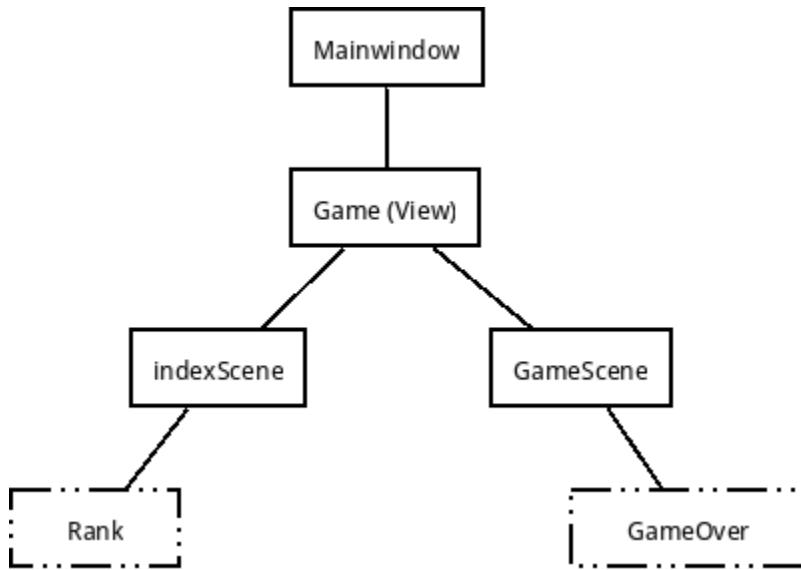
Get scores in 50 seconds

- Star:

0~500	1 Star
500~1000	2 Stars
1000~1500\l	3 Stars

Architecture

- Scene and view



Project requirement

1. Fundamental rule is get scores as high as possible during finite steps or time.
2. Record the number of stars and scores.
3. Show Stones, stars, scores, rule(in menu) and gameover scene.
4. Polymorphism is implemented in Icon based classes including setImage() and setType() functions.
5. Operator overloading is implemented in Square class including == and != and =
6. Emit quit(int, int) signal when user quits game.

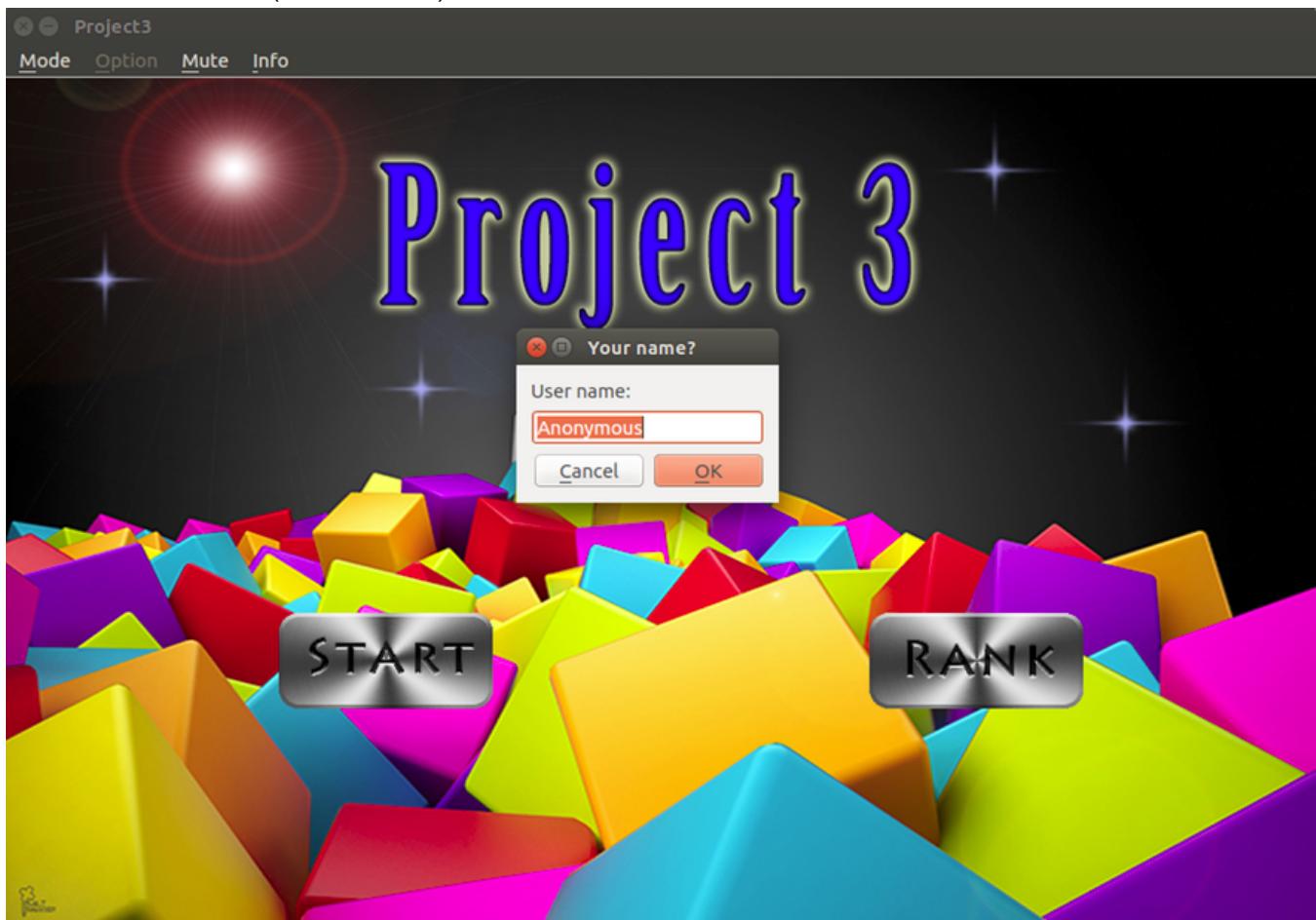
Features

1. Ask user name to create database (default is Anonymous).
2. Rank will keep even close the program (sqlite3 database).
3. Choose two modes in index or game scene (not allow during animation).
4. Sound and BGM
5. Mute sound and music
6. Info menu can get rules and last record

7. Option menu can give up and restart (not allow during animation).
8. Exchange, disappear, move animations.

Screen Shot

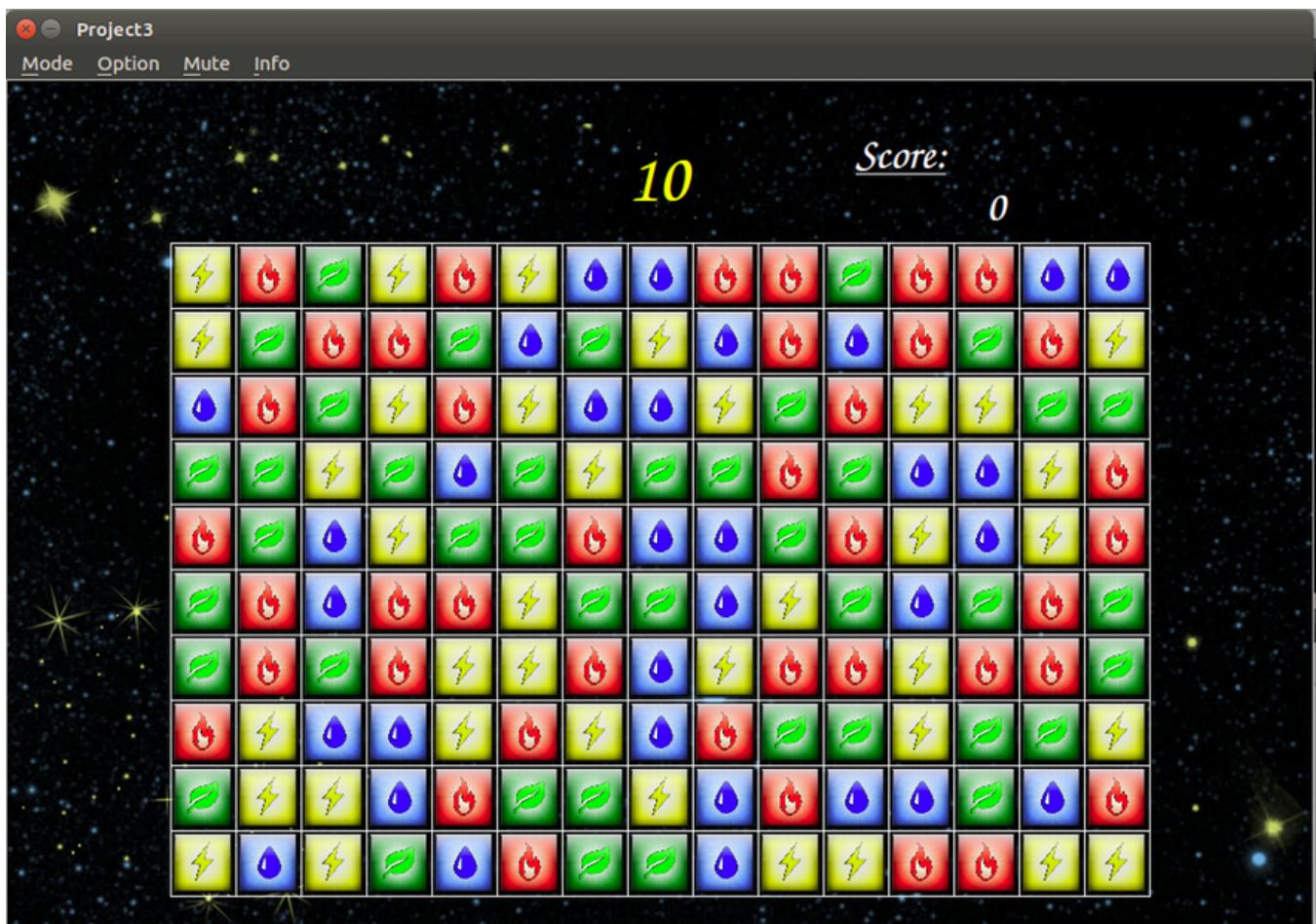
- Ask user name (Start scene)



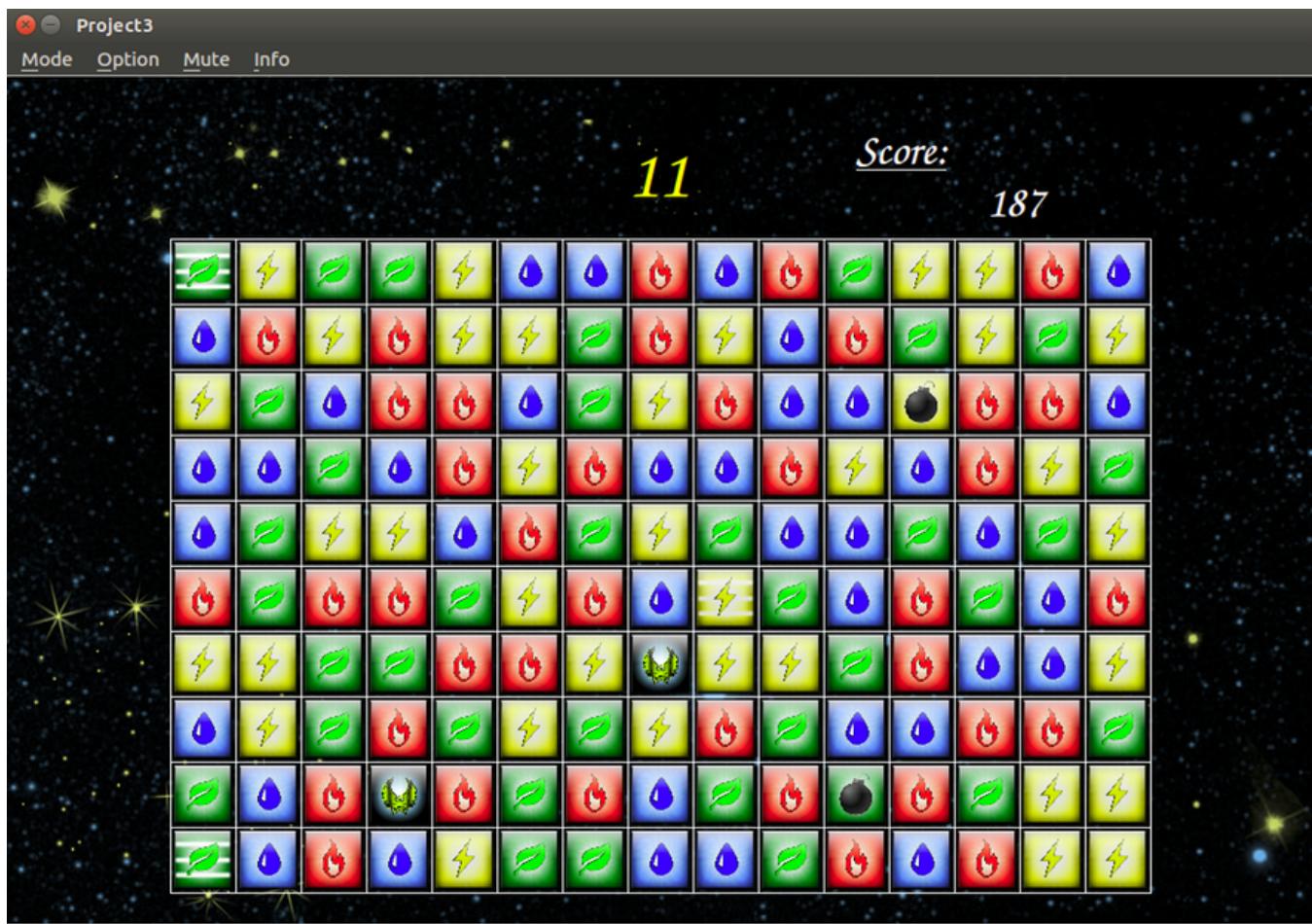
- Rank



- Step mode



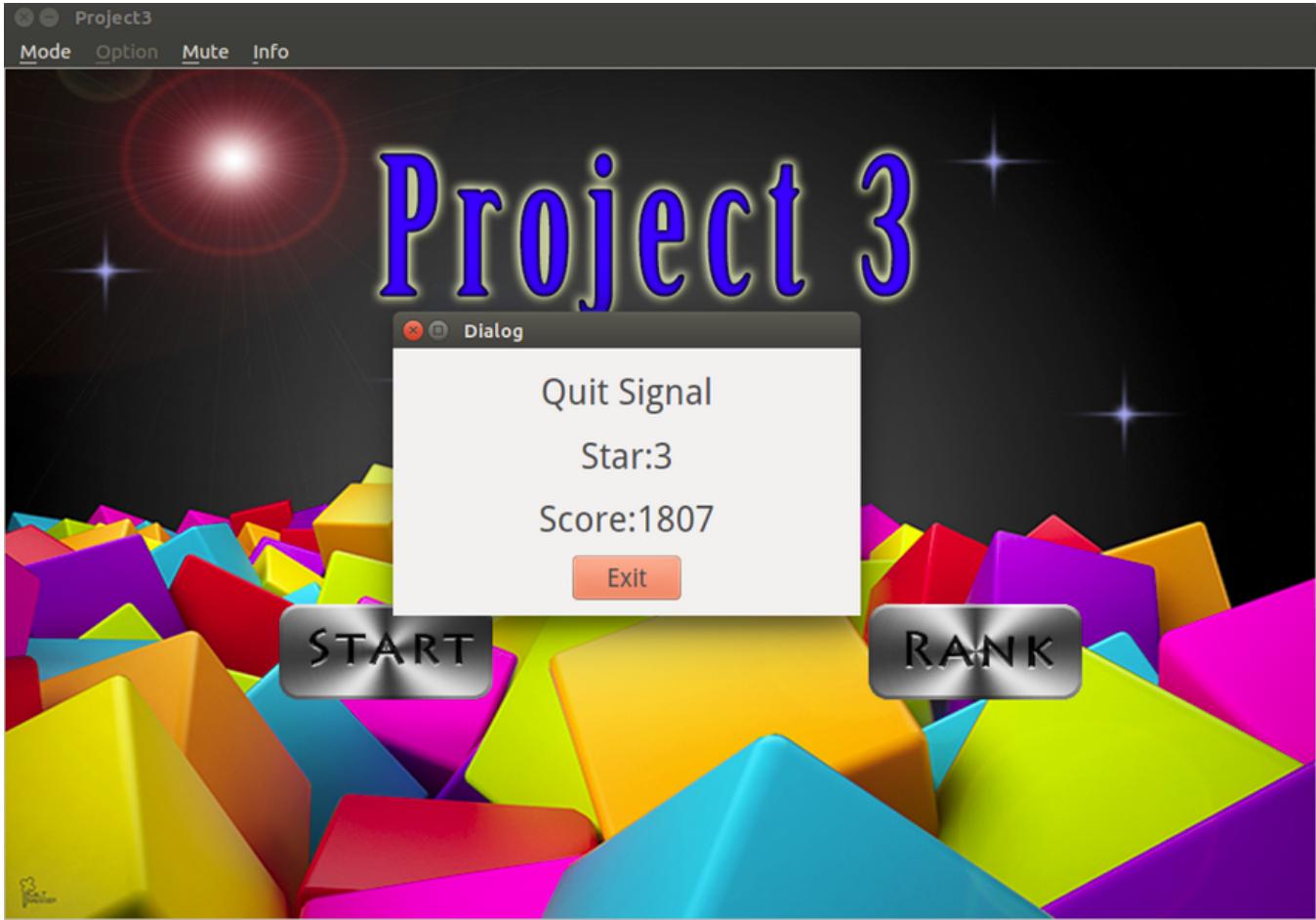
- Time mode



- Game Over



- Signal check



** User name input function will be skipped in this version

UML

- Mainwindow

```

<<QMainwindow>>
Mainwindow

+soundMute: static bool
+musicMute: static bool
-optionMenu: QMenu *
-modeMenu: QMenu *
-muteMenu: QMenu *
-infoMenu: QMenu *
-restartAct: QAction *
-giveupAct: QAction
-stepAct: QAction *
-timeAct: QAction *
-soundMuteAct: QAction *
-musicMuteAct: QAction *
-lastrecordAct: QAction *
-rule_enAct: QAction *
-rule_zhAct: QAction *
-ruleDia: QMessageBox *
-mapper: QSignalMapper *
-game: Game *

+MainWindow(QWidget*, Qt::WindowFlags)
+~MainWindow()
+startGame(): void
+askUserName(): void
+closeEvent(QCloseEvent*): void
+<<signal>> quit(star:int,score:int): void
+<<slot>> menuEnable(): void
+<<slot>> menuDisable(): void
+<<slot>> muteSound(): void
+<<slot>> showRule(x:int): void

```

- GameScene

GameScene	
+hnum: #define = 10 +wnum: #define = 15 +enum Mode{STEP, TIME} +mode: GameScene::Mode +gameBG: QGraphicsPixmapItem * +title: QGraphicsPixmapItem * +scoreLabel: QGraphicsSimpleTextItem * +score: QGraphicsSimpleTextItem * +gameoverBG: QGraphicsRectItem * +gameoverLabel: QGraphicsSimpleTextItem * +gameoverScoreLabel: QGraphicsSimpleTextItem * +gameoverScore: QGraphicsSimpleTextItem * +bestScoreLabel: QGraphicsSimpleTextItem * +bestScore: QGraphicsSimpleTextItem * +squares[hnum][wnum]: Square * +isLink[hnum][wnum]: bool +squaresRect[hnum][wnum]: QRect +theEnd: bool +isAnimation: bool +backIcon: Icon * +againIcon: Icon * +againIconRect: QRect * +backIconRect: QRect * +linkgroup: QParallelAnimationGroup * +fallgroup: QParallelAnimationGroup * +fallgroup2: QParallelAnimationGroup * +fallSequenceGroup: QSequentialAnimationGroup * +exchangeGroup: QParallelAnimationGroup * +reexchangeGroup: QParallelAnimationGroup * +isReExchange: bool +timer: QTimer * +limitLabel: QGraphicsSimpleTextItem * +userName: QString +recordNum: int +hasClick: bool +lasti: int +lastj: int +curl: int +Int +specialLink: bool +sposi: int +sposj: int +newspecial[hnum][wnum]: int +addvalue: int +star[3]: QGraphicsPixmapItem * +finalstar[3]: QGraphicsPixmapItem * +starNum: int	
+GameScene(parent:QObject *=0) +~GameScene() +initSquares(); void +setMode(mode:GameScene::Mode); void +mousePressEvent(event:QGraphicsSceneMouseEvent *); void +keyPressEvent(event:QKeyEvent *); void +init(); void +getIsAnimation(); bool +<<signal>> pressBack(); void +<<signal>> quit(star:int,score:int); void +<<slot>> endDisappearAnimation(); void +<<slot>> endFallAnimation(); void +<<slot>> reshow(); void +<<slot>> endExchangeAnimation(); void +<<slot>> countDown(); void -checkEnd(): bool -gameover(): void -addScore(): void -startLinkAnimation(): void -startFallAnimation(): void -insertRank(); void -set3Link(); bool -checkL(); bool -doEffect(bool:recur_star=true); void -Fall(); void -checkX(x:int); bool -dfs(i:int,j:int,x:int,cnt:int,dir:int,lastType:Square::Type); bool -setLink(); bool -setSpecial(); void -updateStar(); void -polymorphismSetIconPixmap(); void	

- IndexScene

```

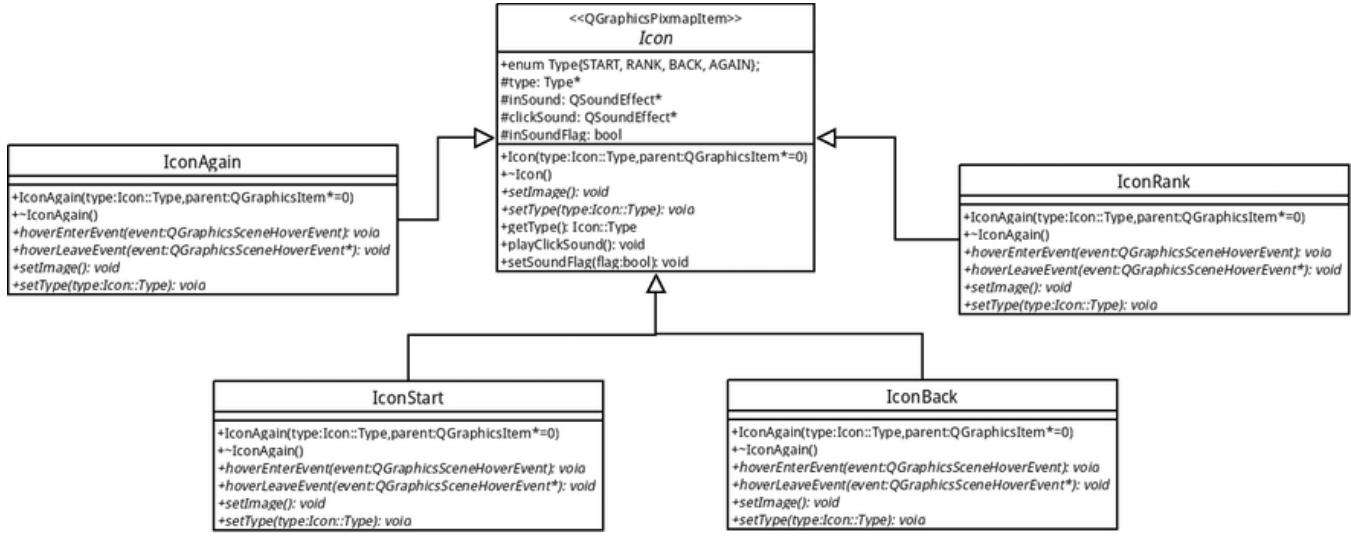
<<QGraphicsScene>>
IndexScene

+indexBG: QGraphicsPixmapItem *
+rankBG: QGraphicsPixmapItem *
+startIcon: Icon *
+rankIcon: Icon *
+backIcon: Icon *
+startIconRect: QRect *
+rankIconRect: QRect *
+backIconRect: QRect *
+rankon: bool
+rankName[10]: QGraphicsSimpleTextItem *
+rankScore[10]: QGraphicsSimpleTextItem *
+rankID[10]: QGraphicsSimpleTextItem *
+bestScore: QGraphicsSimpleTextItem *
+userName: QString

+IndexScene(parent:QObject *=0)
+~IndexScene()
+mousePressEvent(event: QGraphicsSceneMouseEvent *): void
+resetIcon(): void
+<<signal>> pressStart(): void
-showRank(): void
-hideRank(): void
-updateRank(): void
-polymorphismSetIconPixmap(): void

```

- Icon



- Square

```

<<QGraphicsPixmapItem>>
Square

+enum Type{FIRE, WOOD, THUNDER, WATER, STAR}
+enum Effect{VERTICAL, HORIZONTAL, BOMB, NO_EFFECT}
+typenum: static const int = 4
+type: Type
+effect: Effect
+recoverPoint: QPointF
+move: QPropertyAnimation *
+disappear: QPropertyAnimation *

+Square(parent: QGraphicsItem *=0)
+~Square()
+setType(value:Type): void
+getType(): Type
+setEffect(value:Effect): void
+getEffect(): Effect
+getMoveAnimation(): QPropertyAnimation *
+getDisappearAnimation(): QPropertyAnimation *
+setMoveStart(s:QPointF): void
+setDisapper(): void
+recoverAnimation(): void
+setRecoverPoint(): void
+operator==(x:Square const &): bool
+operator!=(x:Square const &): bool
+operator=(x:Square const &): const Square &
+randCreate(): void
+updatePixmap(): void
-init(): void

```

Reference

Library : [Qt Documentation](#)

Music : [魔王魂](#)

Index background : [HDW](#)

Framework : [Qt Game Tutorial](#)

Other images : Youtube, Google and PhotoShop