Brick Break Documentation

Database connectivity - in HighScoreWindow constructor. Reads score table from class database.

- in GameManager saveScores function. Writes score to class database. If it’s in the top 10 it’ll show up on the scoreboard after entering your name.

QProcess - in MainWindow’s createGameWidget function. Reads in the text file generated from CreateText.exe, which is seen by the Controls text (under Score) on the Brick Break game screen

Read/Write to File - in NoteWindow’s saveFile and openText functions. Brings up a dialog box to write notes down. If you hit Save as you can save what’s written as a text file. If you hit Open File, it can read in a text file and display it on the QLineEdit.

Server connection- can be seen in the GlobalScoresWindow for the communicating with server and reading in information from it. The server itself is in the seperate folder BrickBreakerServer. Once that is run alongside the BrickBreak program, it acts as server for BrickBreaker (accessed from Misc->Global High Scores in the QStatusBar) that generates a random High Scores list that displays on a QTableWidget.

Various Qt classes used:

* QMenu & QAction are used by the MainWindow class for the status bar and the options available in that status bar. Located in MainWindow’s createMenus and createActions functions.
* QRect for the Paddle, Brick, and Ball objects. Ball detects collision between the rect of paddle and bricks for movement. Located in their respective .h and .cpp files.
* QStackedWidget for the main menu screen of the game. MenuWidget screen is stacked on top of the gameWidget to switch between the menu page and game page.
* QKeyEvent is used by the GameManager class to detect keyboard presses for Paddle movement, pausing, starting, or restarting the game.
* QPaintEvent is used by the GameManager class to paint the paddle, ball, and bricks and repaint the bricks whenever one is destroyed.
* QTimerEvent is used by the GameManager class for the balls movement, detecting collisions and showing its movement about the screen.
* QDialog represented by the 4 dialog windows in the program (HighScoreWindow, GlobalHighScoreWindow, AboutWindow, and NoteWindow)
* QMainWindow represented by the MainWindow class. It’s the mainWindow of the program

Other than that all that’s left is the basic classes used for the GUI (QPushButton, QVBoxLayout or QHBoxLayout, QMessageBox, QDialogButtonBox, QString, etc..)