

Island Racer

Generated by Doxygen 1.8.13

Contents

| | | |
|----------|--|----------|
| 1 | Hierarchical Index | 1 |
| 1.1 | Class Hierarchy | 1 |
| 2 | Class Index | 3 |
| 2.1 | Class List | 3 |
| 3 | Class Documentation | 5 |
| 3.1 | Car Class Reference | 5 |
| 3.1.1 | Detailed Description | 6 |
| 3.1.2 | Member Function Documentation | 6 |
| 3.1.2.1 | computeUndergroundImpact() | 6 |
| 3.1.2.2 | setCarParams() | 7 |
| 3.1.2.3 | setPosition() | 7 |
| 3.1.2.4 | updatePosition() | 7 |
| 3.2 | Checkpoint Class Reference | 8 |
| 3.2.1 | Detailed Description | 8 |
| 3.2.2 | Constructor & Destructor Documentation | 8 |
| 3.2.2.1 | Checkpoint() | 8 |
| 3.2.3 | Member Function Documentation | 9 |
| 3.2.3.1 | CheckCheckpoint() | 9 |
| 3.3 | Game Class Reference | 9 |
| 3.3.1 | Detailed Description | 10 |
| 3.3.2 | Member Function Documentation | 10 |
| 3.3.2.1 | loadCircuit() | 10 |

| | | |
|---------|---|----|
| 3.4 | mainMenu Class Reference | 10 |
| 3.4.1 | Detailed Description | 11 |
| 3.4.2 | Member Function Documentation | 11 |
| 3.4.2.1 | setBackgroundMusicVolume | 11 |
| 3.4.2.2 | setButtonSoundVolume | 12 |
| 3.4.2.3 | setCarSoundVolume | 12 |
| 3.4.2.4 | setGameMode | 12 |
| 3.5 | PauseMenu Class Reference | 13 |
| 3.5.1 | Detailed Description | 13 |
| 3.5.2 | Member Function Documentation | 14 |
| 3.5.2.1 | setBackgroundMusicVolume | 14 |
| 3.5.2.2 | setButtonSoundVolume | 14 |
| 3.5.2.3 | setCarSoundVolume | 14 |
| 3.6 | Player Class Reference | 14 |
| 3.6.1 | Detailed Description | 15 |
| 3.6.2 | Member Function Documentation | 15 |
| 3.6.2.1 | endRaceDialog | 15 |
| 3.6.2.2 | savePlayerName | 16 |
| 3.7 | Sound Class Reference | 16 |
| 3.7.1 | Detailed Description | 17 |
| 3.8 | Speedometer Class Reference | 17 |
| 3.8.1 | Detailed Description | 18 |
| 3.8.2 | Member Function Documentation | 18 |
| 3.8.2.1 | setVelocity() | 18 |
| 3.9 | Track Class Reference | 18 |
| 3.9.1 | Detailed Description | 19 |
| 3.9.2 | Member Function Documentation | 19 |
| 3.9.2.1 | getLastCheckpointPosition() | 19 |
| 3.9.2.2 | loadTrack() | 19 |
| 3.9.2.3 | ResetCheckpoint() | 20 |

| | | |
|--------------|--|-----------|
| 3.9.2.4 | updateCheckpoints() | 20 |
| 3.10 | Viewport Class Reference | 21 |
| 3.10.1 | Detailed Description | 22 |
| 3.10.2 | Constructor & Destructor Documentation | 22 |
| 3.10.2.1 | Viewport() | 22 |
| 3.10.3 | Member Function Documentation | 22 |
| 3.10.3.1 | pauseGame() | 22 |
| 3.10.3.2 | saveLapTime | 23 |
| 3.10.3.3 | setLabelStyleSheets() | 23 |
| 3.10.3.4 | updateOverlay() | 23 |
| 3.11 | World Class Reference | 23 |
| 3.11.1 | Detailed Description | 25 |
| 3.11.2 | Member Function Documentation | 25 |
| 3.11.2.1 | colorize | 25 |
| 3.11.2.2 | gameLoop | 25 |
| 3.11.2.3 | loadTrack() | 26 |
| 3.11.2.4 | setCar1Pixmap | 26 |
| 3.11.2.5 | setCar2Pixmap | 27 |
| 3.11.2.6 | startColorizeEffect | 27 |
| 3.11.2.7 | startLoop | 27 |
| 3.12 | WorldPosition Class Reference | 27 |
| 3.12.1 | Detailed Description | 28 |
| Index | | 29 |

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

| | |
|-------------------------|----|
| QGraphicsPixmapItem | |
| Car | 5 |
| QGraphicsRectItem | |
| Checkpoint | 8 |
| QGraphicsScene | |
| Track | 18 |
| QGraphicsView | |
| Viewport | 21 |
| QLabel | |
| Speedometer | 17 |
| QMainWindow | |
| mainMenu | 10 |
| World | 23 |
| QObject | |
| Car | 5 |
| Game | 9 |
| Player | 14 |
| Sound | 16 |
| QWidget | |
| PauseMenu | 13 |
| WorldPosition | 27 |

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

| | | |
|-------------------------------|---|----|
| Car | Creates a car with interact methods | 5 |
| Checkpoint | Creates the checkpoints and provide methods to interact with them | 8 |
| Game | Loads the game and connect player with world class | 9 |
| mainMenu | Creates the main menu | 10 |
| PauseMenu | Show pause menu if ESC was pressed while playing the game | 13 |
| Player | Handles the user input after a finished race and saves the highscores | 14 |
| Sound | Handles all sound interaction | 16 |
| Speedometer | This class is a graphical widget which displays a speedometer | 17 |
| Track | Provides the scene for the game | 18 |
| Viewport | Provides the view for the track | 21 |
| World | Manage the whole game procedure | 23 |
| WorldPosition | This class provides easier handling with car positions | 27 |

Chapter 3

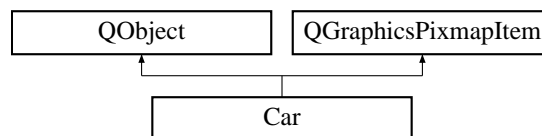
Class Documentation

3.1 Car Class Reference

Creates a car with interact methods.

```
#include <car.h>
```

Inheritance diagram for Car:



Public Slots

- void **setToResetPos** ()
- void **setCarPixmap** (int carIndex)

Signals

- void **playCarSound** ()
Signal to start car sound.
- void **playWaterSound** ()
Signal to start water sound.
- void **startUnderwaterEffect** (int index)
Signal to start the underwater effect if user drove into water.
- void **stopCarSound** ()
Signal to stop car sound.

Public Member Functions

- [Car](#) (b2World *world, [Track](#) *track)
Fill the world with car object.
- void [render](#) ()
Show the car on screen.
- void [killOrthogonalVelocity](#) (b2Body *targetBody)
Let wheels only roll "forward".
- void [computeDriving](#) ()
Compute the driving force.
- void [computeSteering](#) ()
Compute the joint motor torque.
- void [computeSwirl](#) ()
Compute angular velocity for swirl animation.
- void [computeUserInput](#) (enum InputState input)
Compute events on which key pressed.
- void [computeUndergroundImpact](#) (int index)
Compute the different impact to the car.
- void [updatePosition](#) (int index, bool underwaterAnimationActive)
New car position for each time step is calculated.
- void [setPosition](#) (int x, int y, double angle)
New car is created and set to position.
- void [setPosition](#) ([WorldPosition](#) position)
Set car to position.
- void [setCarParams](#) (int speedValue, int accelerationValue, int handlingValue)
Calculate parameter with "garage" values that impact car values.

3.1.1 Detailed Description

Creates a car with interact methods.

3.1.2 Member Function Documentation

3.1.2.1 [computeUndergroundImpact\(\)](#)

```
void Car::computeUndergroundImpact (
    int index )
```

Compute the different impact to the car.

Parameters

| | |
|--------------|---------------------------|
| <i>index</i> | Number of player (1 or 2) |
|--------------|---------------------------|

3.1.2.2 setCarParams()

```
void Car::setCarParams (
    int speedValue,
    int accelerationValue,
    int handlingValue )
```

Calculate parameter with "garage" values that impact car values.

Parameters

| | |
|--------------------------|-------------------------------|
| <i>speedValue</i> | Garage value for topspeed |
| <i>accelerationValue</i> | Garage value for acceleration |
| <i>handlingValue</i> | Garage value for handling |

3.1.2.3 setPosition()

```
void Car::setPosition (
    int x,
    int y,
    double angle )
```

New car is created and set to position.

Parameters

| | |
|--------------|----------------|
| <i>x</i> | x-coordinate |
| <i>y</i> | y-coordinate |
| <i>angle</i> | rotation angle |

3.1.2.4 updatePosition()

```
void Car::updatePosition (
    int index,
    bool underwaterAnimationActive )
```

New car position for each time step is calculated.

Parameters

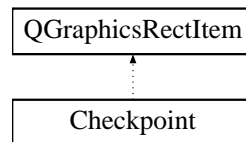
| | |
|----------------------------------|---------------------------------|
| <i>index</i> | Number of player (1 or 2) ! |
| <i>underwaterAnimationActive</i> | underwater effect running state |

3.2 Checkpoint Class Reference

Creates the checkpoints and provide methods to interact with them.

```
#include <checkpoint.h>
```

Inheritance diagram for Checkpoint:



Public Member Functions

- [Checkpoint](#) (int checkpointCount, [WorldPosition](#) *checkpointPositions, [WorldPosition](#) *carResetPositions)
Creates the checkpoints.
- [QGraphicsRectItem](#) * [GetCheckpoint](#) (int index)
Returns checkpoint[index].
- void [CheckCheckpoint](#) ([QGraphicsItem](#) *)
Check if QGraphicsItem colides with checkpoint.
- int [GetNumberOfCheckpoints](#) ()
Returns the total number of checkpoints.
- void [ResetCheckpointcounter](#) ()
Set mCheckpointcounter to starting checkpoint.
- [WorldPosition](#) [getLastCheckpointPosition](#) ()
Returns the reset positon of the last checkpoint you pase.
- int [GetLaps](#) ()
Returns your current lap.

3.2.1 Detailed Description

Creates the checkpoints and provide methods to interact with them.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 Checkpoint()

```
Checkpoint::Checkpoint (
    int checkpointCount,
    WorldPosition * checkpointPositions,
    WorldPosition * carResetPositions )
```

Creates the checkpoints.

Reset positions are also generated.

Parameters

| | |
|----------------------------|---|
| <i>checkpointCount</i> | Number of checkpoints |
| <i>checkpointPositions</i> | Array with x,y and rotation value for each checkpoint |
| <i>carResetPositions</i> | Array with x,y and rotation value for each reset position |

3.2.3 Member Function Documentation

3.2.3.1 CheckCheckpoint()

```
void Checkpoint::CheckCheckpoint (
    QGraphicsItem * car )
```

Check if QGraphicsItem colides with checkpoint.

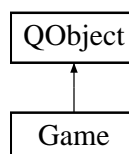
mCheckpointcounter and mLapcounter is increased if necessary.

3.3 Game Class Reference

Loads the game and connect player with world class.

```
#include <game.h>
```

Inheritance diagram for Game:



Public Slots

- void [setGameMode](#) (bool mp)
Slot to select multi- or singleplayer.

Public Member Functions

- [Game](#) ()
Generates world with standard screen size.
- [Game](#) (int screenWidth, int screenHeight, bool fullscreen)
Generates world with selected screen size.
- void [loadCircuit](#) (enum Circuit circuit, int speedValue, int accelerationValue, int handlingValue, int carValue)
Opens selected .circuit file and reads all necessary values.

3.3.1 Detailed Description

Loads the game and connect player with world class.

3.3.2 Member Function Documentation

3.3.2.1 loadCircuit()

```
void Game::loadCircuit (
    enum Circuit circuit,
    int speedValue,
    int accelerationValue,
    int handlingValue,
    int carValue )
```

Opens selected .circuit file and reads all necessary values.

The read values are passed to world class

Parameters

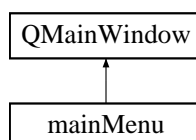
| | |
|--------------------------|---|
| <i>circuit</i> | Circuit enum values which are defined in player.h |
| <i>speedValue</i> | Garage value for topspeed |
| <i>accelerationValue</i> | Garage value for acceleration |
| <i>handlingValue</i> | Garage value for handling |

3.4 mainMenu Class Reference

Creates the main menu.

```
#include <mainmenu.h>
```

Inheritance diagram for mainMenu:



Signals

- void [playBackgroundMusic](#) ()
Signal to start background music.

- void `stopBackgroundMusic` ()
Signal to stop background music.
- void `setBackgroundMusicVolume` (int volume)
Signal to set volume of background music.
- void `playCarSound` ()
Signal to start car sound.
- void `stopCarSound` ()
Signal to stop car sound.
- void `setCarSoundVolume` (int volume)
Signal to set volume of car sound.
- void `playButtonSound` ()
Signal to play button sound.
- void `setButtonSoundVolume` (int volume)
Signal to set volume of button sound.
- void `setGameMode` (bool mp)
Signal to select multi- or singleplayer.

Public Member Functions

- **mainMenu** (QWidget *parent=0)
- void `insertHighscoreToTable` ()
The highscore gets written in the table.
- void `computeMaximumValue` ()
The maximum value for the garage gets computed from the highscores.
- void `setStars` ()
The number of stars above the levels get set according to the highscores.
- void `saveGarage` ()
The values set in the garage get saved.
- void `setCar` ()
The car picture gets set in the garage.

Public Attributes

- QFont **GillSansMT**

3.4.1 Detailed Description

Creates the main menu.

3.4.2 Member Function Documentation

3.4.2.1 setBackgroundMusicVolume

```
void mainMenu::setBackgroundMusicVolume (
    int volume ) [signal]
```

Signal to set volume of background music.

Parameters

| | |
|---------------|--------------------------------|
| <i>volume</i> | Volume of the background music |
|---------------|--------------------------------|

3.4.2.2 setButtonSoundVolume

```
void mainMenu::setButtonSoundVolume (  
    int volume ) [signal]
```

Signal to set volume of button sound.

Parameters

| | |
|---------------|----------------------------|
| <i>volume</i> | Volume of the button sound |
|---------------|----------------------------|

3.4.2.3 setCarSoundVolume

```
void mainMenu::setCarSoundVolume (  
    int volume ) [signal]
```

Signal to set volume of car sound.

Parameters

| | |
|---------------|-------------------------|
| <i>volume</i> | Volume of the car sound |
|---------------|-------------------------|

3.4.2.4 setGameMode

```
void mainMenu::setGameMode (  
    bool mp ) [signal]
```

Signal to select multi- or singleplayer.

Parameters

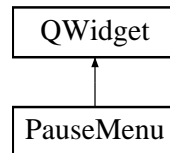
| | |
|-----------|--|
| <i>mp</i> | Value to select game mode (false -> singleplayer, true -> multiplayer) |
|-----------|--|

3.5 PauseMenu Class Reference

Show pause menu if ESC was pressed while playing the game.

```
#include <pausemenu.h>
```

Inheritance diagram for PauseMenu:



Signals

- void `resumeGame` ()
Signal to resume game.
- void `restartGame` ()
Signal to restart game.
- void `quitGame` ()
Signal to quit game.
- void `playBackgroundMusic` ()
Signal to start background music.
- void `stopBackgroundMusic` ()
Signal to stop background music.
- void `setBackgroundMusicVolume` (int volume)
Signal to set volume of background music.
- void `playCarSound` ()
Signal to start car sound.
- void `stopCarSound` ()
Signal to stop car sound.
- void `setCarSoundVolume` (int volume)
Signal to set volume of car sound.
- void `playButtonSound` ()
Signal to play button sound.
- void `setButtonSoundVolume` (int volume)
Signal to set volume of button sound.

Public Member Functions

- **PauseMenu** (int screenHeight, QWidget *parent=0)
- void `playButtonSoundIfOn` ()
Plays button [Sound](#) if active.

3.5.1 Detailed Description

Show pause menu if ESC was pressed while playing the game.

3.5.2 Member Function Documentation

3.5.2.1 setBackgroundMusicVolume

```
void PauseMenu::setBackgroundMusicVolume (
    int volume ) [signal]
```

Signal to set volume of background music.

Parameters

| | |
|---------------|--------------------------------|
| <i>volume</i> | Volume of the background music |
|---------------|--------------------------------|

3.5.2.2 setButtonSoundVolume

```
void PauseMenu::setButtonSoundVolume (
    int volume ) [signal]
```

Signal to set volume of button sound.

Parameters

| | |
|---------------|----------------------------|
| <i>volume</i> | Volume of the button sound |
|---------------|----------------------------|

3.5.2.3 setCarSoundVolume

```
void PauseMenu::setCarSoundVolume (
    int volume ) [signal]
```

Signal to set volume of car sound.

Parameters

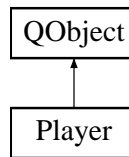
| | |
|---------------|-------------------------|
| <i>volume</i> | Volume of the car sound |
|---------------|-------------------------|

3.6 Player Class Reference

Handles the user input after a finished race and saves the highscores.

```
#include <player.h>
```

Inheritance diagram for Player:



Public Slots

- void [endRaceDialog](#) (QString *mLapTimeEnd, QString mTotalTimeEnd)
- void [savePlayerName](#) (QString name)
Saves the name to highscore file or restarts the dialog if name is empty Connected to the event textValueSelected of mInputDialog.

Signals

- void [playerInputFinished](#) ()
Emitted when user name is saved or user aborted.

Public Member Functions

- void [updateFile](#) ()
Searches for the right insertion point of the highscore, truncates and updates the highscore file.
- void [SetCircuit](#) (Circuit circuit)
Get the right circuit from game.

3.6.1 Detailed Description

Handles the user input after a finished race and saves the highscores.

3.6.2 Member Function Documentation

3.6.2.1 endRaceDialog

```
void Player::endRaceDialog (  
    QString * mLapTimeEnd,  
    QString mTotalTimeEnd ) [slot]
```

Saves the times of the race to member variables for further processing and starts input dialog

Parameters

| | |
|----------------------|--|
| <i>mLapTimeEnd</i> | QString Array with the three lap times |
| <i>mTotalTimeEnd</i> | QString with the total race time |

3.6.2.2 savePlayerName

```
void Player::savePlayerName (
    QString name ) [slot]
```

Saves the name to highscore file or restarts the dialog if name is empty Connected to the event textValueChanged of mInputDialog.

Parameters

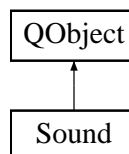
| | |
|-------------|---|
| <i>name</i> | Content of the mInputDialog Box. Sometimes the player's name, mostly crap |
|-------------|---|

3.7 Sound Class Reference

Handles all sound interaction.

```
#include <sound.h>
```

Inheritance diagram for Sound:



Public Slots

- void [playBackgroundMusic](#) ()
Start background music.
- void [stopBackgroundMusic](#) ()
Stop background music.
- void [setBackgroundMusicVolume](#) (int volume)
Adjust background music volume.
- void [playCarSound](#) ()
Start car sound.
- void [stopCarSound](#) ()
Stop car sound.
- void [setCarSoundVolume](#) (int volume)
Adjust car sound volume.

- void `playButtonSound` ()
Start button sound.
- void `playWaterSound` ()
Start water sound.
- void `setButtonSoundVolume` (int volume)
Stop button sound.
- void `playRaceStart1Sound` ()
Start race sound 1.
- void `playRaceStart2Sound` ()
Start race sound 2.
- void `playFinishSound` ()
Start finish sound.

Signals

- void `finished` ()
Signal when thread is finished.

Static Public Member Functions

- static `Sound * getSoundInstance` (QObject *parent)
Static method for singleton access.

3.7.1 Detailed Description

Handles all sound interaction.

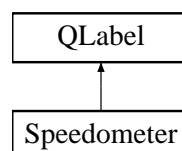
This class is implemented with the singleton design pattern to ensure that only one instance is running. The sound is played in a separate background thread.

3.8 Speedometer Class Reference

This class is a graphical widget which displays a speedometer.

```
#include <speedometer.h>
```

Inheritance diagram for Speedometer:



Public Member Functions

- **Speedometer** (int width, int height, QWidget *parent=nullptr)
- void `setVelocity` (double velocity)

3.8.1 Detailed Description

This class is a graphical widget which displays a speedometer.

3.8.2 Member Function Documentation

3.8.2.1 setVelocity()

```
void Speedometer::setVelocity (
    double velocity )
```

Sets the angle of the needle in speedometer relative to velocity Updates the text label with velocity

Parameters

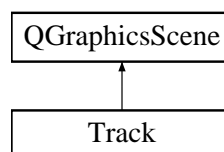
| | |
|-----------------|------------------|
| <i>velocity</i> | current velocity |
|-----------------|------------------|

3.9 Track Class Reference

Provides the scene for the game.

```
#include <track.h>
```

Inheritance diagram for Track:



Signals

- void [LapChanged1](#) ()
Signal if lap of player 1 has changed.
- void [LapChanged2](#) ()
Signal if lap of player 2 has changed.

Public Member Functions

- [Track](#) ()
Init values.
- Underground [getUnderground](#) (int x, int y)
Get the underground for position (x,y)
- [WorldPosition](#) [getLastCheckpointPosition](#) (int index)
Returns the position of the last checkpoint you passed.
- void [updateCheckpoints](#) (QGraphicsItem *item, int index)
Calls the CheckCheckpoint function in checkpoint class and check if lap has changed.
- void [ResetCheckpoint](#) (int index)
Calls the ResetCheckpointcounter function of checkpoint class.
- void [loadTrack](#) (int width, int height, QImage background, QImage grayImage, int checkpointCount, [WorldPosition](#) *checkpointPositions, [WorldPosition](#) *carResetPositions, bool isMultiplayer)
Set up the scene and generating the necessary checkpoints.
- QGraphicsColorizeEffect * [getEffect](#) ()
Get colorize effect.

3.9.1 Detailed Description

Provides the scene for the game.

3.9.2 Member Function Documentation

3.9.2.1 [getLastCheckpointPosition\(\)](#)

```
WorldPosition Track::getLastCheckpointPosition (
    int index )
```

Returns the position of the last checkpoint you passed.

Parameters

| | |
|--------------|-------------------------------|
| <i>index</i> | Indicates the player (1 or 2) |
|--------------|-------------------------------|

3.9.2.2 [loadTrack\(\)](#)

```
void Track::loadTrack (
    int width,
    int height,
    QImage background,
    QImage grayImage,
```

```

    int checkpointCount,
    WorldPosition * checkpointPositions,
    WorldPosition * carResetPositions,
    bool isMultiplayer )

```

Set up the scene and generating the necessary checkpoints.

Parameters

| | |
|----------------------------|---|
| <i>width</i> | Your screen width |
| <i>height</i> | Your screen height |
| <i>background</i> | The track image |
| <i>grayImage</i> | The grey scale image |
| <i>checkpointCount</i> | Number of checkpoints on selected track |
| <i>checkpointPositions</i> | Array with position of all checkpoints |
| <i>carResetPositions</i> | Array with position of all reset positions |
| <i>isMultiplayer</i> | Indicates if you need elements for SP or MP |

3.9.2.3 ResetCheckpoint()

```

void Track::ResetCheckpoint (
    int index )

```

Calls the ResetCheckpointcounter function of checkpoint class.

Parameters

| | |
|--------------|-------------------------------|
| <i>index</i> | Indicates the player (1 or 2) |
|--------------|-------------------------------|

3.9.2.4 updateCheckpoints()

```

void Track::updateCheckpoints (
    QGraphicsItem * item,
    int index )

```

Calls the CheckCheckpoint function in checkpoint class and check if lap has changed.

Parameters

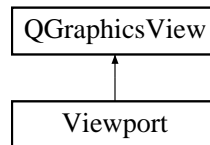
| | |
|--------------|-------------------------------|
| <i>item</i> | The car object |
| <i>index</i> | Indicates the player (1 or 2) |

3.10 Viewport Class Reference

Provides the view for the track.

```
#include <viewport.h>
```

Inheritance diagram for Viewport:



Public Slots

- void [saveLapTime](#) ()
Slot which is called every time a lap is completed.
- void [showLooserLabel](#) ()
Shows loser label to the loser in MP.
- void [showWinnerLabel](#) ()
Shows winner label to the winner in MP.
- void [updateLabelOpacity](#) ()
Fade out winner/loser label in MP.

Signals

- void [stopGame](#) ()
Signal to the world that all laps are completed and stop game loop.
- void [quitGame](#) ()
Signal to quit game after loser/winner label faded out.
- void [raceFinished](#) (QString *mLapTimeEnd, QString mTotalTimeEnd)
Signal that race is finished and start end dialog in player class.

Public Member Functions

- [Viewport](#) (int width, int height, [Track](#) *track, bool isMultiplayer)
Generates a QGraphicsView with some display elements.
- void [startGame](#) ()
Init label params for game start.
- void [resumeGame](#) ()
Restarts the lap and total time counter after pausing game.
- void [pauseGame](#) (bool running)
Save previous time when pausing the game.
- void [restartGame](#) ()
Resets all values (e.g. mCurLap or mLaps) to beginning default.
- void [updateOverlay](#) (QPointF carpos, int fps)
Each time step during gameLoop this function is called to update all labels.
- void [setLabelStyleSheets](#) (int r, int g, int b, int alpha)
Sets Style of labels.

3.10.1 Detailed Description

Provides the view for the track.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 Viewport()

```
Viewport::Viewport (
    int width,
    int height,
    Track * track,
    bool isMultiplayer )
```

Generates a QGraphicsView with some display elements.

Parameters

| | |
|----------------------|--|
| <i>width</i> | The width of the view |
| <i>height</i> | The height of the view |
| <i>track</i> | The QGraphics scene which is displayed |
| <i>isMultiplayer</i> | Needed to adjust size for MP or SP |

3.10.3 Member Function Documentation

3.10.3.1 pauseGame()

```
void Viewport::pauseGame (
    bool running )
```

Save previous time when pausing the game.

If you pause the game during startLoop, running is False and time won't be saved

Parameters

| | |
|----------------|--|
| <i>running</i> | Indicats if pausing the game during gameLoop or starLoop |
|----------------|--|

3.10.3.2 saveLapTime

```
void Viewport::saveLapTime ( ) [slot]
```

Slot which is called every time a lap is completed.

Updates all label params for a new lap

3.10.3.3 setLabelStyleSheets()

```
void Viewport::setLabelStyleSheets (
    int r,
    int g,
    int b,
    int alpha )
```

Sets Style of labels.

Parameters

| | |
|--------------|--------------|
| <i>r</i> | Red value |
| <i>g</i> | Green value |
| <i>b</i> | Blue value |
| <i>alpha</i> | Transparency |

3.10.3.4 updateOverlay()

```
void Viewport::updateOverlay (
    QPointF carpos,
    int fps )
```

Each time step during gameLoop this function is called to update all labels.

carpos and fps is need to calculate current speed

Parameters

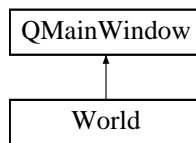
| | |
|---------------|-------------------------------|
| <i>carpos</i> | Current position of car |
| <i>fps</i> | Frames per second of the game |

3.11 World Class Reference

Manage the whole game procedure.

```
#include <world.h>
```

Inheritance diagram for World:



Public Slots

- void [gameLoop](#) ()
Slot for the game loop.
- void [startLoop](#) ()
Slot for start loop.
- void [resumeGame](#) ()
Resume the game after pause.
- void [restartGame](#) ()
Restarts the game and sets all values like you just loaded the track.
- void [stopGame](#) ()
Stops game loop after completing all laps.
- void [exitGame](#) ()
End Game.
- void [startColorizeEffect](#) (int index)
- void [setColorizeStrengthPlayer1](#) ()
Updates colorize strength player1.
- void [setColorizeStrengthPlayer2](#) ()
Updates colorize strength player1.

Signals

- void [colorize](#) (qreal strength)
- void [setCar1Back](#) ()
Resets the car1 to last checkpoint.
- void [setCar2Back](#) ()
Resets the car2 to last checkpoint.
- void [setCar1Pixmap](#) (int i)
- void [setCar2Pixmap](#) (int i)
- void [playRaceSound1](#) ()
Sets the sound effects when a race start (short beep)
- void [playRaceSound2](#) ()
Sets the sound effects when a race start (long beep)

Public Member Functions

- [World](#) (int width, int height)
Set up basic game elements.
- void [loadTrack](#) (int width, int height, QString background_path, QString gray_path, int checkpointCount, [WorldPosition](#) *checkpointPositions, [WorldPosition](#) *carResetPositions, int carCount, [WorldPosition](#) *carPositions, bool isMultiplayer, int speedValue, int accelerationValue, int handlingValue, int carValue)
Load the selected [Track](#) and generates all missing elements for the game.
- void [keyPressEvent](#) (QKeyEvent *keyEvent)
Manage all key press events.
- void [keyReleaseEvent](#) (QKeyEvent *keyEvent)
Manage all key release events.
- void [pauseGame](#) ()
Stops game loop to pause the game and display pausemenu.
- [Viewport](#) * [getViewPlayer](#) (int number)
Get the QGraphicsView of the choosen player.

Static Public Member Functions

- static void [GameExit](#) ()
Removes/Sets world elements to load a new track.

3.11.1 Detailed Description

Manage the whole game procedure.

3.11.2 Member Function Documentation

3.11.2.1 [colorize](#)

```
void World::colorize (
    qreal strength ) [signal]
```

Calls the setStrength Slot of the Colorize Effect to change the transparency

Parameters

| | |
|-----------------|----------------|
| <i>strength</i> | Strength value |
|-----------------|----------------|

3.11.2.2 [gameLoop](#)

```
void World::gameLoop ( ) [slot]
```

Slot for the game loop.

[Car](#) movement is calculated and checkpoints are checked

3.11.2.3 loadTrack()

```
void World::loadTrack (
    int width,
    int height,
    QString background_path,
    QString gray_path,
    int checkpointCount,
    WorldPosition * checkpointPositions,
    WorldPosition * carResetPositions,
    int carCount,
    WorldPosition * carPositions,
    bool isMultiplayer,
    int speedValue,
    int accelerationValue,
    int handlingValue,
    int carValue )
```

Load the selected [Track](#) and generates all missing elements for the game.

Parameters

| | |
|----------------------------|--|
| <i>width</i> | Your screen width |
| <i>height</i> | Your screen height |
| <i>background_path</i> | Dir location of the track image |
| <i>gray_path</i> | Dir location of the grey scale track image |
| <i>checkpointCount</i> | Number of checkpoints on selected track |
| <i>checkpointPositions</i> | Array with position of all checkpoints |
| <i>carResetPositions</i> | Array with position of all reset positions |
| <i>carCount</i> | Number of Players |
| <i>carPositions</i> | Car starting position |
| <i>isMultiplayer</i> | Bool if you selected MP or SP |
| <i>speedValue</i> | Garage value for topspeed |
| <i>accelerationValue</i> | Garage value for acceleration |
| <i>handlingValue</i> | Garage value for handling |
| <i>carValue</i> | Garage QPixmap for the car |

3.11.2.4 setCar1Pixmap

```
void World::setCar1Pixmap (
    int i ) [signal]
```

Sets the pixmap of the car1

Parameters

| | |
|----------|---------------------------|
| <i>i</i> | Car index |
|----------|---------------------------|

3.11.2.5 setCar2Pixmap

```
void World::setCar2Pixmap (
    int i ) [signal]
```

Sets the pixmap of the car2

Parameters

| | |
|----------|---------------------------|
| <i>i</i> | Car index |
|----------|---------------------------|

3.11.2.6 startColorizeEffect

```
void World::startColorizeEffect (
    int index ) [slot]
```

Start swirl effect

Parameters

| | |
|------------|---|
| <i>car</i> | Car which has water contact |
|------------|---|

3.11.2.7 startLoop

```
void World::startLoop ( ) [slot]
```

Slot for start loop.

Displays the beginning sequence

3.12 WorldPosition Class Reference

This class provides easier handling with car positions.

```
#include <worldposition.h>
```

Public Member Functions

- [WorldPosition](#) ()
Init x, y and angle value to default 0.
- [WorldPosition](#) (int [x](#), int [y](#), double [angle](#))
Init x, y and angle value selected values.
- int [x](#) () const
Get x value.
- void [setX](#) (int [x](#))
Set y value.
- int [y](#) () const
Get x value.
- void [setY](#) (int [y](#))
Set y value.
- double [angle](#) () const
Get angle value.
- void [setAngle](#) (double [angle](#))
Set angle value.

3.12.1 Detailed Description

This class provides easier handling with car positions.

Index

- Car, [5](#)
 - computeUndergroundImpact, [6](#)
 - setCarParams, [6](#)
 - setPosition, [7](#)
 - updatePosition, [7](#)
- CheckCheckpoint
 - Checkpoint, [9](#)
- Checkpoint, [8](#)
 - CheckCheckpoint, [9](#)
 - Checkpoint, [8](#)
- colorize
 - World, [25](#)
- computeUndergroundImpact
 - Car, [6](#)
- endRaceDialog
 - Player, [15](#)
- Game, [9](#)
 - loadCircuit, [10](#)
- gameLoop
 - World, [25](#)
- getLastCheckpointPosition
 - Track, [19](#)
- loadCircuit
 - Game, [10](#)
- loadTrack
 - Track, [19](#)
 - World, [26](#)
- mainMenu, [10](#)
 - setBackgroundMusicVolume, [11](#)
 - setButtonSoundVolume, [12](#)
 - setCarSoundVolume, [12](#)
 - setGameMode, [12](#)
- pauseGame
 - Viewport, [22](#)
- PauseMenu, [13](#)
 - setBackgroundMusicVolume, [14](#)
 - setButtonSoundVolume, [14](#)
 - setCarSoundVolume, [14](#)
- Player, [14](#)
 - endRaceDialog, [15](#)
 - savePlayerName, [16](#)
- ResetCheckpoint
 - Track, [20](#)
- saveLapTime
 - Viewport, [22](#)
- savePlayerName
 - Player, [16](#)
- setBackgroundMusicVolume
 - mainMenu, [11](#)
 - PauseMenu, [14](#)
- setButtonSoundVolume
 - mainMenu, [12](#)
 - PauseMenu, [14](#)
- setCar1Pixmap
 - World, [26](#)
- setCar2Pixmap
 - World, [27](#)
- setCarParams
 - Car, [6](#)
- setCarSoundVolume
 - mainMenu, [12](#)
 - PauseMenu, [14](#)
- setGameMode
 - mainMenu, [12](#)
- setLabelStyleSheets
 - Viewport, [23](#)
- setPosition
 - Car, [7](#)
- setVelocity
 - Speedometer, [18](#)
- Sound, [16](#)
- Speedometer, [17](#)
 - setVelocity, [18](#)
- startColorizeEffect
 - World, [27](#)
- startLoop
 - World, [27](#)
- Track, [18](#)
 - getLastCheckpointPosition, [19](#)
 - loadTrack, [19](#)
 - ResetCheckpoint, [20](#)
 - updateCheckpoints, [20](#)
- updateCheckpoints
 - Track, [20](#)
- updateOverlay
 - Viewport, [23](#)
- updatePosition
 - Car, [7](#)
- Viewport, [21](#)
 - pauseGame, [22](#)
 - saveLapTime, [22](#)

setLabelStyleSheets, [23](#)
updateOverlay, [23](#)
Viewport, [22](#)

World, [23](#)
 colorize, [25](#)
 gameLoop, [25](#)
 loadTrack, [26](#)
 setCar1Pixmap, [26](#)
 setCar2Pixmap, [27](#)
 startColorizeEffect, [27](#)
 startLoop, [27](#)
WorldPosition, [27](#)