

MBRDNA iTunes Top 100 App Specification

By Richard Ying

Table of Contents

- Overview
- Build & Run
- Usage
- Classes
- Design
- Use Cases
- Test Cases

Overview

This was my first exposure to QT and QML, so it was a good learning experience. The framework has great cross-platform compatibilities and its separation of concerns encourages powerful and well organized object oriented software. I tried to follow the model/view structure in this application, utilizing slots and signals when interacting between the QML and C++. I also followed object oriented practices like inheritance and encapsulation in designing my classes.

Build & Run

Currently optimized for desktop use for showcasing purposes. To optimize for mobile use, set the property variable `tap_to_play` to true in `view.qml`. This disables double click and enables single tap.

Mac OSX (El Capitan 10.11.3 with QT Creator 5.7.0)

Run

Run the app from the included 0-100RealQuick.dmg file

Build

Install QT Creator for Mac and open the 0-100RealQuick.pro file in QT Creator and build and run, making sure the default kit is detected in QT Creator>Preferences>Build & Run>Kits

Deploy

Run `macdeployqt` following these instructions

<http://dragly.org/2012/01/13/deploy-qt-applications-for-mac-os-x/>

<http://stackoverflow.com/questions/17475788/qt-5-1-and-mac-bug-making-macdeployqt-not-working-properly/17591828#17591828>

Usage

Currently optimized for desktop use for showcasing purposes. See use cases for general usage.

Double Click: plays the double clicked song, pauses it if it is already active and playing on the bottom bar

Single Click: highlights the clicked song, and activates other on screen UI controls.

Enter/Return: plays the highlighted song, or the first song if none is highlighted, or replays the highlighted song if it is also the active song in the bottom bar

Space: plays/pauses the active song in the bottom bar, or the highlighted song if none is active, or the first song if none is highlighted.

Left: plays previous song, replay the first song if the first song is active in the bottom bar, or the first song if no songs were played before.

Right: plays next song, or the first song if none is active in the bottom bar, or the first song if the last song is active

Up: highlights the previous song, no wrap

Down: highlights the next song, no wrap

Scroll: scrolls through the list, no wrap

Classes (C++)

Song Class

Song QObject holding information for ListView model and QMediaPlaylist

Derived From: QObject

QProperties: index, title, artist, album, imageUrl

Private Variables: m_index, m_title, m_artist, m_album, m_image_url, m_itunes_url, m_song_file_url

Public Functions:

index()	index accessor for QObject returns QString
title()	title accessor for QObject returns QString
artist()	artist accessor for QObject returns QString
album()	album accessor for QObject returns QString
imageUrl()	album image accessor for QObject returns QString
itunesUrl()	iTunes URL accessor for QObject returns QString
songFileUrl()	song file accessor for QMediaPlaylist returns QString

ItunesList Class

Top itunes list QObject with network access and song list object

Derived From: QObject

Private Variables: m_network_access_manager, m_song_list, m_player_song_list

Public Functions:

songList()
Song list accessor for ListView

Params: none

Return: m_song_list, the song list for ListView model

playerSongList()

Player song list accessor for Player

Params: none

Return: m_player_song_list, the song list for QMediaPlaylist

buildSongList()

Slot to build song list from network reply. Reads and parses XML reply into m_song_list and m_player_song_list

Params: reply, the network reply

Return: void

Player Class

Music player controller QObject with QQuickView, QMediaPlayer, and QMediaPlaylist

Derived From: QObject

Private Variables: m_view, m_controller, m_player, m_playlist, itunes_list

Private Slots:

play()

Play song at index

Params: index, the song index

Return: void

pause()

Pause song

Params: none

Return: void

restart(qint32 index)

Restart song at index

Params: index, the song index

Return: void

setSongView()

Set QML view to current song

Params: none

Return: void

finished(QMediaPlayer::State state)

Set stop state and send signal to view if no errors caused the stop

Params: state, the QMediaPlayer state

Return: void

position(qint64 position)

Set QML view progress bar position and fade song if <2.5 seconds left

Params: position, the QMediaPlayer position

Return: void

setPosition(qreal percent)

Set QMediaPlayer position from view

Params: percent, the percent position of total progress bar clicked

Return: void

setAutoPlay(bool autoplay)

Set QMediaPlaylist autoplay

Params: autoplay, true or false

Return: void

error()

Handle media error, disables autoplay and sets view

Params: none

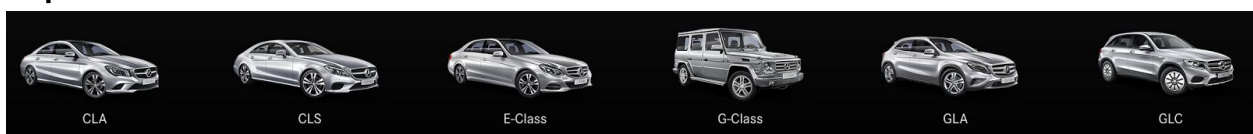
Return: void

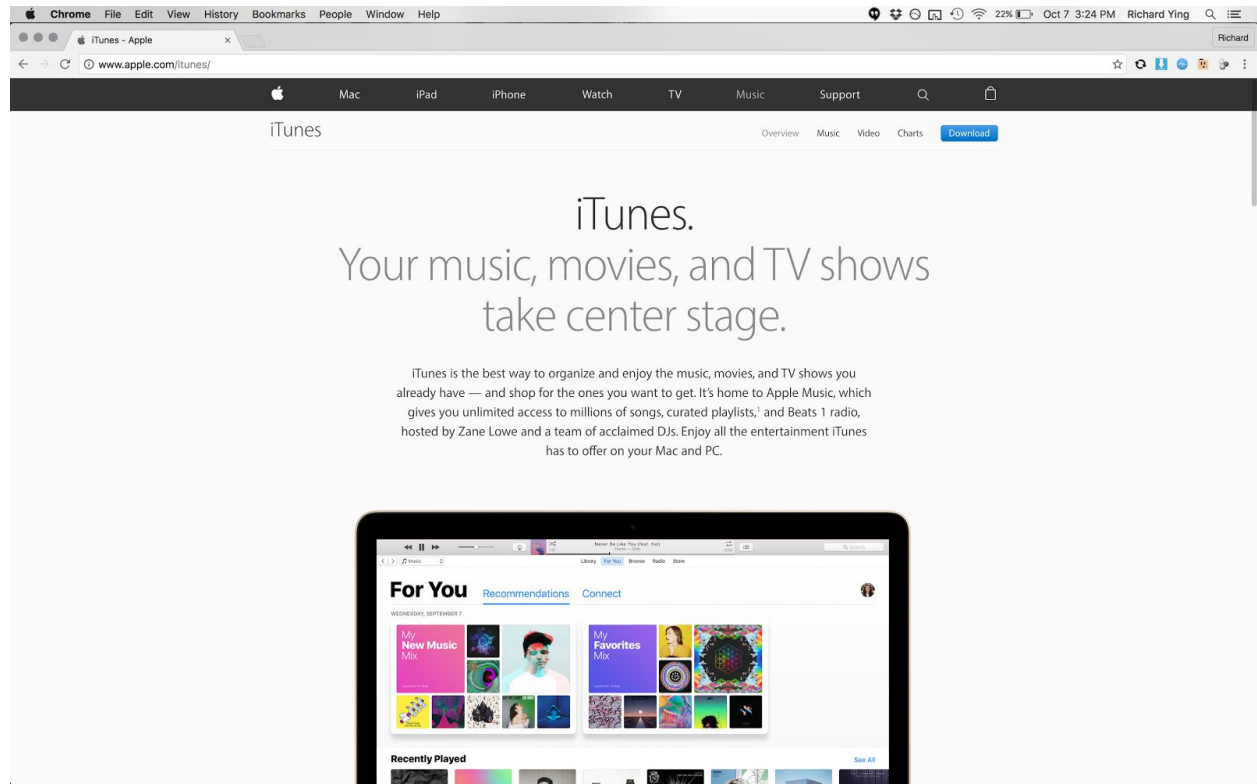
See code comments for QML elements, functions, signals, and slots

Design

Initially mocked up in Sketch based on user stories from two music aficionados and one regular user. Made design decisions based on a sample oriented playlist, such as omitting timestamps and disabling autoplay by default, hiding the bottom controller bar when a song is finished. The overall design and color scheme consists of neutral greys, reflecting the image of the Mercedes brand as well as Apple and iTunes. The name 0-100RealQuick refers to both the function of the app to get the top 100 songs real quick, as well as the popular phrase among the target market of millennial pop song listeners, originating from Drake's song of the same name, which is hopefully fitting considering the phrase's pop music origin. I implemented smooth transitions such as the progress bar's hover and progression, the gradient fade of long titles in the list view, as well as fading out the end of samples since they're usually cut off at an odd spot.

Inspiration:





I made mockups and all design assets on Sketch, and created an interactive wireframe on Invision to receive user feedback regarding the layout of the controls and information display.















<https://invis.io/Q68TCSN28#/193738783> Main

I implemented the application's QML down to the pixel of the mockup, and made further adjustments given user feedback.

Final Sketch Mockup

	SONG	ARTIST	ALBUM	0-100REALQUICK
	1 This Town	Niall Horan	This Town - Single	
	2 Closer (feat. Halsey)	The Chainsmokers	Closer (feat. Halsey) - Single	
	3 All We Know (feat. Phoebe Ryan)	The Chainsmokers	All We Know (feat. Phoebe Ryan) - Single	
	4 Heathens	twenty one pilots	Suicide Squad: The Album	
	5 Gold	Kiara	low kii savage - EP	
	6 Let Me Love You (feat. Justin Bieber)	DJ Snake	Encore	
	7 Starving (feat. Zedd)	Hailee Steinfeld & Grey	Starving (feat. Zedd) - Single	
	8 Forever Country	Artists Of Then, Now & Forever	Forever Country - Single	
	9 All My Friends	Jacob Sartorius	All My Friends - Single	
	10 Cold Water (feat. Justin Bieber & MØ)	Major Lazer	Cold Water (feat. Justin Bieber & MØ) - Single	
	11 Blue Ain't Your Color	Keith Urban	Ripcord	
	12 Juju On That Beat (TZ Anthem)	Zay Hilfigerrr & Zayion McCall	Juju On That Beat (TZ Anthem) - Single	
	13 CAN'T STOP THE FEELING! (Original...	Justin Timberlake	CAN'T STOP THE FEELING! (Original Song From DreamWorks Animation's Trolls)	
	14 Unsteady	X Ambassadors	VHS	
<div> <div> <div>◀</div> <div>⏸</div> <div>▶</div> </div> <div> <div>Closer (feat. Halsey)</div> <div>The Chainsmokers</div> </div> <div> <div>↺</div> </div> </div>				

Final QML Application

	SONG	ARTIST	ALBUM	O-100REALQUICK
	1 24K Magic	Bruno Mars	24K Magic	
	2 Closer (feat. Halsey)	The Chainsmokers	Closer (feat. Halsey) - Single	
	3 Today	Brad Paisley	Today - Single	
	4 Heathens	twenty one pilots	Suicide Squad: The Album	
	5 Million Reasons	Lady Gaga	Joanne (Deluxe)	
	6 Love Me Now	John Legend	Love Me Now - Single	
	7 Juju On That Beat (TZ Anthem)	Zay Hilfigerrr & Zayion McCall	Juju On That Beat (TZ Anthem) - Single	
	8 Gold	Kiiara	low kii savage - EP	
	9 Let Me Love You (feat. Justin Bieber)	DJ Snake	Encore	
	10 Blue Ain't Your Color	Keith Urban	Ripcord	
	11 Starving (feat. Zedd)	Hailee Steinfeld & Grey	Starving (feat. Zedd) - Single	
	12 This Town	Niall Horan	This Town - Single	
	13 Setting the World on Fire (with P!nk)	Kenny Chesney	Cosmic Hallelujah	
	14 All We Know (feat. Phoebe Ryan)	The Chainsmokers	All We Know (feat. Phoebe Ryan) - Single	

⏮
▶
⏭

Setting the World on Fire (with P!nk)
Kenny Chesney

🔄

Known limitations: the app is currently not horizontally responsive, and extreme edge cases such as long song titles will overlap and get cut off in the controller bar.

Use Cases

1. View iTunes Top 100

Description: Describes the ability for the user to view iTunes Top 100

Outcome: The user shall view a list of iTunes Top 100 content

Goals: The user wants to view iTunes Top 100

Pre-conditions: none

Post-conditions: the application opens to a display of the iTunes Top 100

Workflow (desktop): the user opens the 0-100RealQuick application and scrolls through the list or uses the up and down arrow keys to navigate the selection

Workflow (mobile): the user opens the 0-100RealQuick application and swipes through the list

2. Play a Song

Description: Describes the ability for the user to play a song

Outcome: The user shall listen to the desired song

Goals: The user wants to listen to a song

Pre-conditions: the application is open

Post-conditions: the application displays the selected song and plays its audio, displays its position in the progress bar, then returns to the stopped state (when autoplay is off)

Workflow (desktop): the user double clicks a song, or uses arrow keys and presses enter on the highlighted song.

Workflow (mobile): the user taps a song

3. Continuously Play Songs

Description: Describes the ability for the user to autoplay songs

Outcome: The user shall listen to the songs in order

Goals: The user wants to listen to the songs in order

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application displays the autoplay indicator is on and songs will play one after another

Workflow (desktop): the user clicks the autoplay icon (two arrows in a circle)

Workflow (mobile): the user taps the autoplay icon (two arrows in a circle)

4. Pause a Song

Description: Describes the ability for the user to pause a song

Outcome: The user shall pause the song

Goals: The user wants to pause the song

Pre-conditions: the song is playing

Post-conditions: the application displays the selected song in the paused position

Workflow (desktop): the user presses space or the pause button or the currently active song

Workflow (mobile): the user taps the pause button

5. Skip a Song

Description: Describes the ability for the user to skip a song

Outcome: The user shall skip the song

Goals: The user wants to skip the song

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application displays the next or previous song and plays its audio

Workflow (desktop): the user presses the next or prev arrows or the left or right arrow keys on the keyboard

Workflow (mobile): the user presses the next or prev arrows

6. Skip within a Song

Description: Describes the ability for the user to navigate within a song

Outcome: The user shall set the position of the current song to play

Goals: The user wants to set the position of the current song to play

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application displays the updated position of the song in the progress bar

Workflow (desktop): the user hovers over the progress bar and clicks or drags to a new position

Workflow (mobile): the user taps or drags the progress bar to a new position

6. View iTunes Link

Description: Describes the ability for the user to view a song's iTunes link

Outcome: The user shall view the song's iTunes link

Goals: The user wants to view the song's iTunes link

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application opens a browser window with the iTunes link of the song

Workflow (desktop): the user clicks the song title/artist in the bottom bar

Workflow (mobile): the user taps the song title/artist in the bottom bar

Test Cases

1. iTunes Top 100 View

Pre-conditions: none

Post-conditions: the application opens to a display of the iTunes Top 100

Workflow (desktop): the user opens the 0-100RealQuick application and scrolls through the list or uses the up and down arrow keys to navigate the selection

Workflow (mobile): the user opens the 0-100RealQuick application and swipes through the list

Errors: the application display error message if no songs are available

2. Play a Song

Pre-conditions: the application is open

Post-conditions: the application displays the selected song and plays its audio, displays its position in the progress bar, then returns to the stopped state (when autoplay is off)

Workflow (desktop): the user double clicks a song, or uses arrow keys and presses enter on the highlighted song. Prev/left arrow plays the first song if none is selected or the first song is already selected. Next/right arrow plays the next song, or the first song if none is selected or the last song is selected. Enter/Return key plays the highlighted song, or the first song if none is highlighted, or restarts the current song if Enter/Return is hit again. The highlight should follow a newly played song. The volume indicator should display.

Workflow (mobile): the user taps a song

Errors: the application display error message if the song is not available

3. Continuously Play Songs

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application displays the autoplay indicator is on and songs will play one after another

Workflow (desktop): the user clicks the autoplay icon (two arrows in a circle)

Workflow (mobile): the user taps the autoplay icon (two arrows in a circle)

Errors: autoplay shall turn off if the songs are not available

4. Pause a Song

Pre-conditions: the song is playing

Post-conditions: the application displays the selected song in the paused position. The volume indicator should hide. The progress bar should stop.

Workflow (desktop): the user presses space or the pause button or the currently active song

Workflow (mobile): the user taps the pause button

Errors: None

5. Skip a Song

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application displays the next or previous song and plays its audio. The song should start playing regardless of the previous pause/play state.

Workflow (desktop): the user presses the next or prev arrows or the left or right arrow keys on the keyboard. Prev/left arrow plays the first song if none is selected or the first song is already selected. Next/right arrow plays the next song, or the first song if none is selected or the last song is selected.

Workflow (mobile): the user presses the next or prev arrows

Errors: the application displays an error message and disables autoplay if the song is not available.

6. Skip within a Song

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application displays the updated position of the song in the progress bar

Workflow (desktop): the user hovers over the progress bar and clicks or drags to a new position. When hovering, enlarge the progress bar. When dragging, enlarge the mouse area vertically to fill the application window.

Workflow (mobile): the user taps or drags the progress bar to a new position

Errors: none

6. View iTunes Link

Pre-conditions: a song is currently active (playing or paused)

Post-conditions: the application opens a browser window with the iTunes link of the song

Workflow (desktop): the user clicks the song title/artist in the bottom bar

Workflow (mobile): the user taps the song title/artist in the bottom bar

Errors: none