1

00:00:02,560 --> 00:00:06,319

hello everyone

2

00:00:03,520 --> 00:00:07,120

and welcome to another episode of qb64

3

00:00:06,319 --> 00:00:09,760

report

4

00:00:07,120 --> 00:00:12,000

i'm your host felipeto and today we're

5

00:00:09,760 --> 00:00:14,559

gonna have a hybrid episode

6

00:00:12,000 --> 00:00:17,440

let's cover how to use external media

7

00:00:14,559 --> 00:00:20,480

files in your qb64 programs

8

00:00:17,440 --> 00:00:22,720

both for audio and for images the topics

9

00:00:20,480 --> 00:00:25,359

and techniques covered in this podcast

10

00:00:22,720 --> 00:00:26,960

will also be showcased in a companion

11

00:00:25,359 --> 00:00:28,560

youtube video released

12

00:00:26,960 --> 00:00:30,960

today you can find the link in the

13

00:00:28,560 --> 00:00:34,960

description without further ado

14

00:00:30,960 --> 00:00:38,320

let's get to it the basic idea of using

15

00:00:34,960 --> 00:00:42,000

any type of media file with qb64

16

00:00:38,320 --> 00:00:44,800

is load it and use it

17

00:00:42,000 --> 00:00:46,719

we have a set of commands for sounds we

18

00:00:44,800 --> 00:00:47,760

have a similar set of commands for

19

00:00:46,719 --> 00:00:50,239

images

20

00:00:47,760 --> 00:00:51,360

and what they have in common is that

21

00:00:50,239 --> 00:00:54,480

once you use

22

00:00:51,360 --> 00:00:56,239

either sound open or load image we don't

23

00:00:54,480 --> 00:00:59,440

have to worry about

24

00:00:56,239 --> 00:01:02,719

actually opening the file and uh

25

00:00:59,440 --> 00:01:05,760

converting decoding none of that

26

00:01:02,719 --> 00:01:07,760

all of that is done by qb64 in uh

27

00:01:05,760 --> 00:01:09,040

no matter the file type you're going to

28

00:01:07,760 --> 00:01:11,200

be using

29

00:01:09,040 --> 00:01:13,119

all of the compatible file formats are

30

00:01:11,200 --> 00:01:15,840

dealt with the same way

31

00:01:13,119 --> 00:01:16,400

so first let's take a look at how to

32

00:01:15,840 --> 00:01:19,040

load

33

00:01:16,400 --> 00:01:20,880

and manipulate images all the commands

34

00:01:19,040 --> 00:01:23,280

we're going to be covering today

35

00:01:20,880 --> 00:01:24,400

are prefixed commands which means they

36

00:01:23,280 --> 00:01:27,280

are prefixed by

37

00:01:24,400 --> 00:01:28,080

an underscore oqb64 commands which are

38

00:01:27,280 --> 00:01:30,720

not part

39

00:01:28,080 --> 00:01:31,759

of the original qbasic quick basics

40

00:01:30,720 --> 00:01:34,159

subset

41

00:01:31,759 --> 00:01:35,200

are prefixed with an underscore unless

42

00:01:34,159 --> 00:01:37,439

you use

43

00:01:35,200 --> 00:01:39,759

the no prefix meta command that was

44

00:01:37,439 --> 00:01:41,920

introduced in version 1.4

45

00:01:39,759 --> 00:01:42,880

so for example if you're going to use an

46

00:01:41,920 --> 00:01:45,040

image file

47

00:01:42,880 --> 00:01:46,159

you're going to use the command load

48

00:01:45,040 --> 00:01:48,479

image

49

00:01:46,159 --> 00:01:49,680

load image takes two parameters the

50

00:01:48,479 --> 00:01:52,000

first one of them

51

00:01:49,680 --> 00:01:53,600

is a file name which can be stored in a

52

00:01:52,000 --> 00:01:55,680

variable of course

53

00:01:53,600 --> 00:01:57,920

and the second parameter indicates the

54

00:01:55,680 --> 00:02:00,079

screen mode you're going to be using

55

00:01:57,920 --> 00:02:01,680

for most of our coding today of our

56

00:02:00,079 --> 00:02:04,640

examples today we're going to be using

57

00:02:01,680 --> 00:02:05,040

32 as a second parameter or even nothing

58

00:02:04,640 --> 00:02:07,280

because

59

00:02:05,040 --> 00:02:09,759

if your program is already in a 32-bit

60

00:02:07,280 --> 00:02:12,000

mode you don't need to specify a 32-bit

61

00:02:09,759 --> 00:02:14,319

mode for loading the image

62

00:02:12,000 --> 00:02:15,360

the load image command returns you a

63

00:02:14,319 --> 00:02:17,520

handle

64

00:02:15,360 --> 00:02:19,840

which is just a number that's going to

65

00:02:17,520 --> 00:02:22,640

be stored in a long variable

66

00:02:19,840 --> 00:02:23,520

and this number indicates for qb64 where

67

00:02:22,640 --> 00:02:26,400

in memory

68

00:02:23,520 --> 00:02:27,360

your image is stored with that in hand

69

00:02:26,400 --> 00:02:30,800

you can use the

70

00:02:27,360 --> 00:02:34,000

put image command the put image command

71

00:02:30,800 --> 00:02:36,000

takes a coordinate on screen and

72

00:02:34,000 --> 00:02:37,120

an image handle that was already loaded

73

00:02:36,000 --> 00:02:39,440

in memory by you

74

00:02:37,120 --> 00:02:40,879

and then it simply does what it says it

75

00:02:39,440 --> 00:02:43,040

will do it puts

76

00:02:40,879 --> 00:02:44,400

the image on the screen the put image

77

00:02:43,040 --> 00:02:46,879

command however

78

00:02:44,400 --> 00:02:48,640

holds many more possibilities than that

79

00:02:46,879 --> 00:02:52,000

it allows you for example

80

00:02:48,640 --> 00:02:54,480

to stretch or shrink an image on screen

81

00:02:52,000 --> 00:02:55,680

because you can specify a coordinate for

82

00:02:54,480 --> 00:02:58,959

placing the image

83

00:02:55,680 --> 00:03:01,200

which corresponds to the top left corner

84

00:02:58,959 --> 00:03:02,239

or you can also specify the coordinates

85

00:03:01,200 --> 00:03:05,040

for the bottom

86

00:03:02,239 --> 00:03:05,680

right corner of the image which means

87

00:03:05,040 --> 00:03:08,879

that

88

00:03:05,680 --> 00:03:10,959

you can specify a larger area

89

00:03:08,879 --> 00:03:12,400

or a smaller area where the image is

90

00:03:10,959 --> 00:03:15,040

going to be placed

91

00:03:12,400 --> 00:03:16,959

qb64 handles the stretching or the

92

00:03:15,040 --> 00:03:17,680

shrinking all internally you don't have

93

00:03:16,959 --> 00:03:20,560

to worry

94

00:03:17,680 --> 00:03:22,000

about any of that also with the put

95

00:03:20,560 --> 00:03:25,599

image command

96

00:03:22,000 --> 00:03:28,799

you get what was possible in qbasic

97

00:03:25,599 --> 00:03:30,799

only by using two different commands for

98

00:03:28,799 --> 00:03:34,080

graphics back in the day you could use

99

00:03:30,799 --> 00:03:38,000

put and get get would save

100

00:03:34,080 --> 00:03:41,280

a portion of the screen to a an array

101

00:03:38,000 --> 00:03:44,319

in memory and you could later use put

102

00:03:41,280 --> 00:03:46,480

to place that portion of image to

103

00:03:44,319 --> 00:03:48,959

another part of the screen this was a

104

00:03:46,480 --> 00:03:51,680

technique very much used for games

105

00:03:48,959 --> 00:03:54,239

for sprites and they can still be used

106

00:03:51,680 --> 00:03:56,000

nowadays but put image embodies all of

107

00:03:54,239 --> 00:03:58,879

that functionality which means

108

00:03:56,000 --> 00:03:59,120

that using just put image you can place

109

00:03:58,879 --> 00:04:01,040

an

110

00:03:59,120 --> 00:04:02,400

image everywhere on the screen and you

111

00:04:01,040 --> 00:04:04,879

can also pick

112

00:04:02,400 --> 00:04:05,920

a specific area of the source image to

113

00:04:04,879 --> 00:04:08,080

be placed

114

00:04:05,920 --> 00:04:10,080

that allows you for example to load a

115

00:04:08,080 --> 00:04:13,120

sprite sheeting memory

116

00:04:10,080 --> 00:04:15,680

and place only a portion of it

117

00:04:13,120 --> 00:04:18,079

at a time which allows for example if

118

00:04:15,680 --> 00:04:20,720

you have a character spreadsheet

119

00:04:18,079 --> 00:04:22,240

with all of its frames that indicate uh

120

00:04:20,720 --> 00:04:24,160

it's walking for example

121

00:04:22,240 --> 00:04:26,880

you don't need to load several different

122

00:04:24,160 --> 00:04:28,400

files and place them according to what

123

00:04:26,880 --> 00:04:29,520

frame you want to show you can load a

124

00:04:28,400 --> 00:04:31,680

single file

125

00:04:29,520 --> 00:04:34,560

and use input image you can indicate to

126

00:04:31,680 --> 00:04:37,360

qb64 what portion of the original image

127

00:04:34,560 --> 00:04:39,120

you want to display at a given time now

128

00:04:37,360 --> 00:04:40,479

regarding the handle you get when you

129

00:04:39,120 --> 00:04:42,639

load an image

130

00:04:40,479 --> 00:04:43,680

it's going to be a long variable as i

131

00:04:42,639 --> 00:04:46,560

said before

132

00:04:43,680 --> 00:04:47,440

and it's going to hold a negative number

133

00:04:46,560 --> 00:04:50,479

if the number

134

00:04:47,440 --> 00:04:52,639

you get after you load an image is -1

135

00:04:50,479 --> 00:04:54,320

that indicates that loading failed

136

00:04:52,639 --> 00:04:55,040

either because the file could not be

137

00:04:54,320 --> 00:04:56,560

found

138

00:04:55,040 --> 00:04:59,360

you're not going to get a file not found

139

00:04:56,560 --> 00:05:02,479

error in this case or because

140

00:04:59,360 --> 00:05:04,880

the image format was corrupted or maybe

141

00:05:02,479 --> 00:05:07,039

it was a format qb64 cannot handle

142

00:05:04,880 --> 00:05:08,880

kv64 can handle most formats most

143

00:05:07,039 --> 00:05:12,639

popular formats of today

144

00:05:08,880 --> 00:05:15,680

like bmp png jpeg

145

00:05:12,639 --> 00:05:16,560

gif if you want to call it gif i'm not

146

00:05:15,680 --> 00:05:19,360

gonna mind

147

00:05:16,560 --> 00:05:20,800

even psd files can be loaded if

148

00:05:19,360 --> 00:05:24,000

depending on how they were

149

00:05:20,800 --> 00:05:26,080

saved by adobe photoshop qb64 can load

150

00:05:24,000 --> 00:05:27,759

them of course with some limitations

151

00:05:26,080 --> 00:05:30,880

you're not going to be able to

152

00:05:27,759 --> 00:05:32,639

handle the the layers but if it was

153

00:05:30,880 --> 00:05:36,000

saved a certain way

154

00:05:32,639 --> 00:05:38,479

you can load and display images in qb64

155

00:05:36,000 --> 00:05:39,199

even if they were saved in photoshop

156

00:05:38,479 --> 00:05:41,120

format

157

00:05:39,199 --> 00:05:42,320

another observation regarding formats is

158

00:05:41,120 --> 00:05:45,120

that gif images

159

00:05:42,320 --> 00:05:47,120

if they are animated originally qb64 is

160

00:05:45,120 --> 00:05:50,080

going to load the first frame for you

161

00:05:47,120 --> 00:05:50,400

if you want to actually have an animated

162

00:05:50,080 --> 00:05:51,919

gif

163

00:05:50,400 --> 00:05:54,320

in your programs you're going to have to

164

00:05:51,919 --> 00:05:56,240

roll your own we even have

165

00:05:54,320 --> 00:05:58,560

an implementation of a gift player in

166

00:05:56,240 --> 00:06:00,880

our wiki which is pretty handful and

167

00:05:58,560 --> 00:06:03,280

useful i've used it sometimes

168

00:06:00,880 --> 00:06:05,759

and i've even integrated it with inform

169

00:06:03,280 --> 00:06:06,479

for qb64 so that it can play animated

170

00:06:05,759 --> 00:06:08,960

gifs

171

00:06:06,479 --> 00:06:10,880

but qb64 will load a gif image and

172

00:06:08,960 --> 00:06:13,759

display the first frame for you

173

00:06:10,880 --> 00:06:14,160

back to handles so back to handles if

174

00:06:13,759 --> 00:06:17,120

you get

175

00:06:14,160 --> 00:06:19,039

a handle value of -1 that indicates the

176

00:06:17,120 --> 00:06:20,080

file could not be loaded or it was not

177

00:06:19,039 --> 00:06:22,560

even found

178

00:06:20,080 --> 00:06:23,840

any other valid values for a handle will

179

00:06:22,560 --> 00:06:26,319

be negative

180

00:06:23,840 --> 00:06:27,199

to check that an image was properly

181

00:06:26,319 --> 00:06:30,720

loaded

182

00:06:27,199 --> 00:06:33,280

check that it's lesser than -1

183

00:06:30,720 --> 00:06:34,160

if the value of a handle is lesser than

184

00:06:33,280 --> 00:06:35,840

-1

185

00:06:34,160 --> 00:06:37,600

that indicates the image was properly

186

00:06:35,840 --> 00:06:39,680

loaded and you can work with it

187

00:06:37,600 --> 00:06:40,639

as i said before a handle indicates to

188

00:06:39,680 --> 00:06:42,400

qb64

189

00:06:40,639 --> 00:06:44,000

where an image can be found in memory

190

00:06:42,400 --> 00:06:44,960

but that's internal

191

00:06:44,000 --> 00:06:47,520

you're not going to be able to

192

00:06:44,960 --> 00:06:49,360

manipulate the images data directly

193

00:06:47,520 --> 00:06:51,680

using the handle the handle will be

194

00:06:49,360 --> 00:06:54,000

passed to functions like put image

195

00:06:51,680 --> 00:06:55,440

so that they can do their thing it's

196

00:06:54,000 --> 00:06:58,240

also very important to

197

00:06:55,440 --> 00:06:59,840

free unused images when you're done

198

00:06:58,240 --> 00:07:01,360

working with them so if you don't need

199

00:06:59,840 --> 00:07:03,199

to display an image anymore

200

00:07:01,360 --> 00:07:05,599

you're going to use the free image

201

00:07:03,199 --> 00:07:07,520

command to release the memory that was

202

00:07:05,599 --> 00:07:09,199

being used for that image remember that

203

00:07:07,520 --> 00:07:12,479

if you have a very tiny

204

00:07:09,199 --> 00:07:14,400

uh png file for example when you load it

205

00:07:12,479 --> 00:07:16,319

it's not going to take for example five

206

00:07:14,400 --> 00:07:18,319

kilobytes in memory it's going to take

207

00:07:16,319 --> 00:07:19,120

much more because qb64 is going to

208

00:07:18,319 --> 00:07:21,440

decode it

209

00:07:19,120 --> 00:07:22,319

and what's actually stored in memory is

210

00:07:21,440 --> 00:07:24,960

the full

211

00:07:22,319 --> 00:07:26,560

image data images in memory take a

212

00:07:24,960 --> 00:07:28,560

little bit more memory than they take in

213

00:07:26,560 --> 00:07:29,120

this because of the decoding process of

214

00:07:28,560 --> 00:07:31,840

course

215

00:07:29,120 --> 00:07:33,520

so the basic outline of the process is

216

00:07:31,840 --> 00:07:36,400

you load an image

217

00:07:33,520 --> 00:07:37,440

you use it with put image to display it

218

00:07:36,400 --> 00:07:39,360

on screen

219

00:07:37,440 --> 00:07:40,880

and after you're done with it you use

220

00:07:39,360 --> 00:07:43,280

free image

221

00:07:40,880 --> 00:07:44,240

now regarding sounds the process is very

222

00:07:43,280 --> 00:07:45,759

similar

223

00:07:44,240 --> 00:07:47,360

regarding the file formats you can load

224

00:07:45,759 --> 00:07:50,879

with qb64

225

00:07:47,360 --> 00:07:53,840

there are ogg files wav files or

226

00:07:50,879 --> 00:07:54,720

mp3 files we have the sound open

227

00:07:53,840 --> 00:07:58,000

function

228

00:07:54,720 --> 00:08:01,199

snd open actually which is also going

229

00:07:58,000 --> 00:08:02,400

to take a file parameter which can be

230

00:08:01,199 --> 00:08:05,199

passed as a literal

231

00:08:02,400 --> 00:08:05,919

or as a variable and in return you get a

232

00:08:05,199 --> 00:08:07,840

handle

233

00:08:05,919 --> 00:08:10,000

the difference here is that sound

234

00:08:07,840 --> 00:08:12,080

handles are actually positive so

235

00:08:10,000 --> 00:08:14,160

image handles are negative in memory

236

00:08:12,080 --> 00:08:16,319

sound handles are positive

237

00:08:14,160 --> 00:08:18,879

so if you try to open a sound file with

238

00:08:16,319 --> 00:08:20,319

a sound open command and you get a zero

239

00:08:18,879 --> 00:08:22,879

that means it couldn't be loaded

240

00:08:20,319 --> 00:08:24,960

positive handles indicate that the sound

241

00:08:22,879 --> 00:08:26,800

was properly loaded and can be played

242

00:08:24,960 --> 00:08:28,720

so with the sound loaded in memory you

243

00:08:26,800 --> 00:08:31,199

can use the sound play command

244

00:08:28,720 --> 00:08:33,039

actually snd play and then you can

245

00:08:31,199 --> 00:08:34,880

manipulate the sound in different ways

246

00:08:33,039 --> 00:08:35,599

you can pause it with the smd pause

247

00:08:34,880 --> 00:08:38,240

command

248

00:08:35,599 --> 00:08:39,120

you can stop playback with the smd stop

249

00:08:38,240 --> 00:08:41,440

command

250

00:08:39,120 --> 00:08:44,320

you can confirm that a sound is playing

251

00:08:41,440 --> 00:08:46,800

or not with the smd playing command

252

00:08:44,320 --> 00:08:47,680

you can loop a sound using the smd loop

253

00:08:46,800 --> 00:08:49,279

command

254

00:08:47,680 --> 00:08:51,279

and if you need to play a sound

255

00:08:49,279 --> 00:08:54,240

repeatedly i'm imagining here

256

00:08:51,279 --> 00:08:57,519

a game with a shooting sound for example

257

00:08:54,240 --> 00:08:59,360

you can use the snd play copy statement

258

00:08:57,519 --> 00:09:00,720

that's going to also take a handle

259

00:08:59,360 --> 00:09:02,320

parameter

260

00:09:00,720 --> 00:09:04,399

and before it plays a sound it's going

261

00:09:02,320 --> 00:09:06,080

to generate a copy memory and play it

262

00:09:04,399 --> 00:09:08,080

which allows for a same sound to be

263

00:09:06,080 --> 00:09:10,959

played simultaneously with

264

00:09:08,080 --> 00:09:11,839

another instance of it so outlining the

265

00:09:10,959 --> 00:09:14,160

process

266

00:09:11,839 --> 00:09:14,880

you use snd open to load a sounding

267

00:09:14,160 --> 00:09:16,399

memory

268

00:09:14,880 --> 00:09:17,920

depending on the size of the file it's

269

00:09:16,399 --> 00:09:18,720

going to take a little while to decode

270

00:09:17,920 --> 00:09:20,560

and load

271

00:09:18,720 --> 00:09:22,399

that will give you a handle that's a

272

00:09:20,560 --> 00:09:23,600

long variable and it's going to be a

273

00:09:22,399 --> 00:09:26,080

positive value

274

00:09:23,600 --> 00:09:28,160

if loading was successful then you use

275

00:09:26,080 --> 00:09:30,480

the smd play command to play it

276

00:09:28,160 --> 00:09:31,680

s and d pause to pause it s and d stop

277

00:09:30,480 --> 00:09:33,760

to stop it

278

00:09:31,680 --> 00:09:35,360

just that you can also get information

279

00:09:33,760 --> 00:09:38,959

about how long a sound file

280

00:09:35,360 --> 00:09:41,040

is in seconds using the smd lan function

281

00:09:38,959 --> 00:09:44,160

which returns the length of the file

282

00:09:41,040 --> 00:09:44,800

in seconds so as i stated before this is

283

00:09:44,160 --> 00:09:46,959

a mini

284

00:09:44,800 --> 00:09:49,040

episode i'm just giving you a quick

285

00:09:46,959 --> 00:09:50,000

glimpse at the commands we can use to

286

00:09:49,040 --> 00:09:53,519

play different types of

287

00:09:50,000 --> 00:09:55,519

media in qb64 sounds and images

288

00:09:53,519 --> 00:09:57,519

you can find a link to the youtube video

289

00:09:55,519 --> 00:09:59,839

that complements this podcast

290

00:09:57,519 --> 00:10:01,120

in the description below thank you very

291

00:09:59,839 --> 00:10:05,839

much for listening

292

00:10:01,120 --> 00:10:05,839

catch you guys next time