Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине: «ОСиСП»

Тема: «Поддержка сетевого взаимодействия программ»

Выполнил:

Студент 3 курса

Группы ПО-7

Комиссаров А.Е.

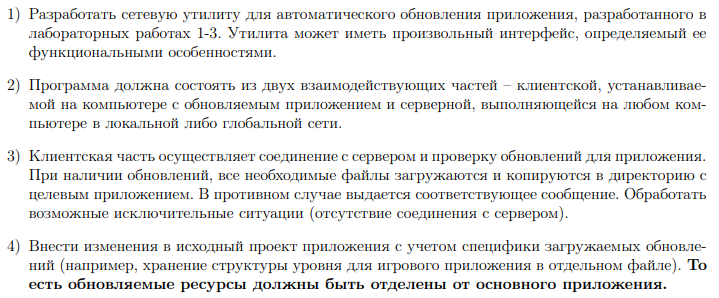
Проверил:

Булей Е.В.

2022

**Цель:** ознакомиться с возможностями, предлагаемыми Qt для поддержки сетевого взаимодействия программ.

**Общее задание:**

****



**Ход работы:**

**tetris\_launcher.py**

from PyQt5 import uic, QtTest

from PyQt5.QtGui import \*

from PyQt5.QtCore import \*

from PyQt5.QtWidgets import \*

from github import Github

import os

from threading import \*

import sys

import tetris

import time as time\_module

class LauncherWindow(QDialog):

def \_\_init\_\_(self):

super(LauncherWindow, self).\_\_init\_\_()

uic.loadUi("UI/Launcher.ui", self)

self.show()

self.buttonLaunch = self.findChild(QPushButton, "LaunchButton")

self.buttonLaunch.clicked.connect(self.LaunchPress)

self.buttonUpdate = self.findChild(QPushButton, "CheckUpdatesButton")

self.buttonUpdate.clicked.connect(self.UpdatePress)

self.buttonExit = self.findChild(QPushButton, "ExitButton")

self.buttonExit.clicked.connect(self.ExitPress)

self.versionText = self.findChild(QLabel, "VersionText")

self.versionText.setText("ver. " + self.getLocalVersion())

self.keepSoundAwake = QTimer()

self.keepSoundAwake.setInterval(2700)

self.keepSoundAwake.start()

def transformToSize(self, width, height):

old\_h = self.height()

old\_w = self.width()

end\_x = round(self.x() + ((self.width() - width)/2))

end\_y = round(self.y() + ((self.height() - height)/2))

tick\_duration = 10 #default = 10

prevH = 9999

prevW = 9999

x = self.x()

y = self.y()

i = 0

while not (self.height() == height and self.width() == width):

diff\_h = (height - self.height())/9

diff\_w = (width - self.width())/9

i+=1

offset\_w = old\_w - self.width()

offset\_h = old\_h - self.height()

new\_x = round(x + (offset\_w/2))

new\_y = round(y + (offset\_h/2))

self.move(new\_x, new\_y)

if(prevW != diff\_w or prevH != diff\_h):

prevW = diff\_w

prevH = diff\_h

self.setFixedHeight(round(self.height() + diff\_h))

self.setFixedWidth( round(self.width() + diff\_w))

else:

self.move(end\_x, end\_y)

self.setFixedHeight(height)

self.setFixedWidth(width)

break

QtTest.QTest.qWait(tick\_duration)

def showLauncher(self):

self.show()

self.game.hide()

self.transformToSize(630, 360)

#Launch button is pressed

def LaunchPress(self):

self.transformToSize(360, 480)

self.hide()

self.game = tetris.launchGame()

self.game.closed.connect(self.showLauncher)

#Update button is pressed

def UpdatePress(self):

self.Update()

#Exit button is pressed

def ExitPress(self):

app.quit()

def getLocalVersion(self):

try:

version\_file = open("version.txt")

version = version\_file.readline()

return version

except OSError:

print("could not open file")

return None

def getRemoteVersion(self):

repo = self.githubInstance.get\_repo(self.repository)

return repo.get\_contents("version.txt").decoded\_content.decode()

def Authenticate(self,token):

g = Github(token)

print("Authenticated as " + g.get\_user().name)

return g

def noUpdateNeeded(self):

print("No update needed.")

self.updateNotFound = QMessageBox(

QMessageBox.Information, "No update needed",

"You are already running the latest version, no need to update.",

(QMessageBox.Ok)

)

self.updateNotFound.exec()

def showUpdateConfirm(self,lcVersion, rmVersion):

'''

Asks the user if he wants to update.

'''

self.updateConfirm = QMessageBox(

QMessageBox.Information, "Update detected",

"A new update has been detected. Your local version is " + lcVersion + ". Are you willing to update your game version to " + rmVersion + "?",

(QMessageBox.Yes | QMessageBox.No)

)

return self.updateConfirm.exec()

def Update(self):

self.access\_token = "ghp\_gAXxjYPP0FFmcKstWjafTvh0JA4ALn0Vjzs1"

self.repository = "combo-wombo/OSiSP\_Lab4"

self.githubInstance = self.Authenticate(self.access\_token)

self.repo = self.githubInstance.get\_repo(self.repository)

self.contents = self.repo.get\_contents("")

local = self.getLocalVersion()

remote = self.getRemoteVersion()

if(local < remote):

print("Update detected (local : " + local + ", remote : " + remote + ").")

if(self.showUpdateConfirm(local, remote) == QMessageBox.Yes):

self.updProg = QWidget()

self.updProg.setWindowTitle('QProgressBar')

self.updProg.pbar = QProgressBar(self.updProg)

self.updProg.pbar.setValue(0)

self.updProg.resize(300, 100)

self.updProg.vbox = QVBoxLayout()

self.updProg.vbox.addWidget(self.updProg.pbar)

self.updProg.setLayout(self.updProg.vbox)

self.updProg.show()

self.updProg.pbar.show()

self.thread = Thread(self.contents,self.repo)

self.thread.\_signal.connect(self.signal\_accept)

self.thread.start()

else:

self.noUpdateNeeded()

def signal\_accept(self, msg):

msg = int(msg)

print("received",msg)

if msg == 999:

self.versionText.setText("ver. " + self.getLocalVersion())

self.updProg.close()

self.updProg = QMessageBox(

QMessageBox.Information, "App updated",

"The app has been updated to the latest version.",

(QMessageBox.Ok)

)

self.updProg.exec()

else:

if self.updProg.pbar.value() > 99:

self.updProg.pbar.setValue(100)

else:

self.updProg.pbar.setValue(msg)

#=================================##=================================#

app = QApplication(sys.argv)

launcher = LauncherWindow()

app.exec\_()

#=================================##=================================#

**Результат работы программы:**

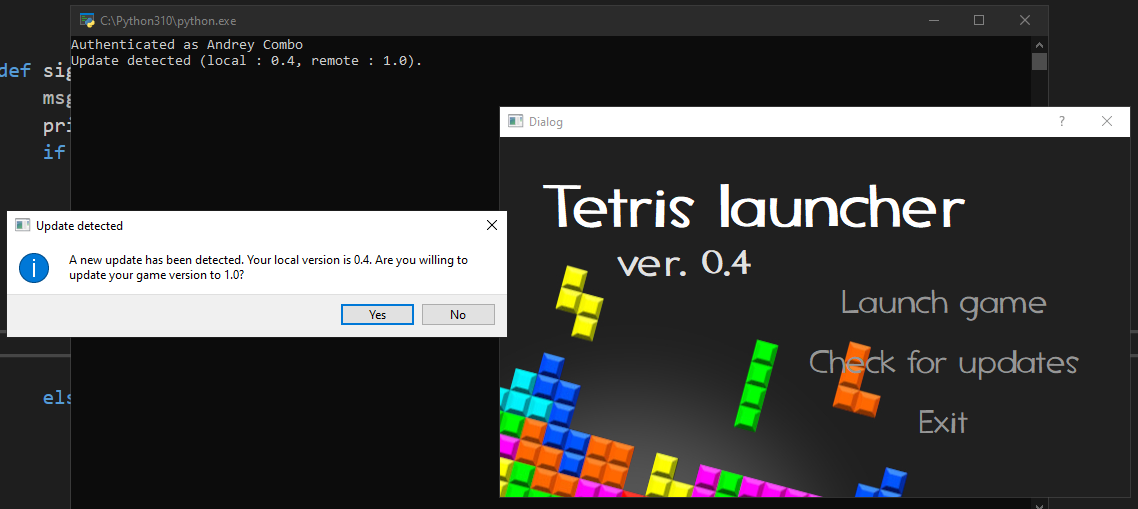
****

Рис. 1,2 – Результат работы программы

**Вывод:** я ознакомился с возможностями, предлагаемыми Qt для поддержки сетевого взаимодействия программ.