

Ιόνιο Πανεπιστήμιο

Τμήμα Πληροφορικής



-- Πτυχιακή Εργασία --

*Διεπαφή χρήστη διαχείρισης και εκτέλεσης
εφαρμογών μέσω πολυτροπικών
περιφερειακών συσκευών*

Κωνσταντίνος Τουρτσάκης, Π2019140

Επιβλέπων: – Μιχαήλ Στεφανιδάκης –

May 30, 2024

Επιβλέπων

Μιχαήλ Στεφανιδάκης, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο Τμήμα Πληροφορικής

Τριμελής Επιτροπή

Μιχαήλ Στεφανιδάκης, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο Τμήμα Πληροφορικής
Δημήτριος Ρίγγας, ΕΔΙΠ,
Ιόνιο Πανεπιστήμιο Τμήμα Πληροφορικής
Θεόδωρος Ανδρόνικος, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο Τμήμα Πληροφορικής

Περίληψη

To do...

Contents

<i>A</i>	<i>Εισαγωγή</i>	<i>1</i>
<i>B</i>	<i>Ανάπτυξη εφαρμογών σε Qt6</i>	<i>2</i>
B.1	Τι ΕΙΝΑΙ ΤΟ QT;	2
B.2	Ένα βασικό πρόγραμμα σε Qt6	2
B.3	Προσθήκη στοιχείων γραφικής διεπαφής σε Qt6	4
B.3.1	Δημιουργία QPushButton	4
B.3.2	Δημιουργία γραφικού πλαισίου	6
B.3.3	Δημιουργία λίστας αντικειμένων	8
B.3.4	Διάβασμα αρχείων από directory	9
B.3.5	Δημιουργία input field	10
B.3.6	Αποθήκευση στοιχείων στον δίσκο	11
B.3.7	Ανάκτηση εικονιδίων συστήματος αρχείων	12
<i>C</i>	<i>Η βιβλιοθήκη XInput</i>	<i>14</i>
C.1	Τι ΕΙΝΑΙ ΤΟ XINPUT;	14
C.2	Η κλάση του χειριστήριου	14
<i>D</i>	<i>Διεπαφή χρήστη διαχείρισης και εκτέλεσης εφαρμογών μέσω πολυτροπικών</i>	

<i>Contents</i>	iv
<i>περιφερειακών συσκευών</i>	16
D.1 ΠΕΡΙΓΡΑΦΗ ΛΟΓΙΣΜΙΚΟΥ	16
D.2 ΕΚΤΕΛΕΣΗ ΕΦΑΡΜΟΓΩΝ ΜΕ ΠΟΝΤΙΚΙ	16
<i>E Σχήματα και Πίνακες</i>	18
E.1 ΣΧΗΜΑΤΑ	18
E.2 ΠΙΝΑΚΕΣ	19
<i>F Τελευταία Μέρη</i>	24
<i>Παράρτημα Α'</i>	26
<i>Βιβλιογραφία</i>	27
<i>Συντμήσεις</i>	29
<i>Γλωσσάρι Εθνικών Όρων</i>	30

List of Figures

E.1 Ένα δίκτυο.	19
E.2 Το ίδιο δίκτυο που απεικονίζεται στο Σχήμα E.1 αλλά λίγο μικρότερο. . . .	20
E.3 Το ίδιο δίκτυο σε διπλή απεικόνιση.	21
E.4 Το ίδιο δίκτυο σε διπλή απεικόνιση σε σμίκρυνση και δίχως κείμενο για τα υπο-σχήματα.	21
E.5 Το ίδιο δίκτυο σε τετραπλή απεικόνιση.	22

List of Tables

E.1 Παράδειγμα Πίνακα.	23
--------------------------------	----

Chapter A

Εισαγωγή

Είναι πλέον σίγουρο πως η χρήση ενός προσωπικού υπολογιστή γίνεται μέσω του πληκτρολογίου σε συνδυασμό με την χρήση του ποντικιού για την περιήγηση του χρήστη μέσα στο γραφικό περιβάλλον των σύγχρονων συστημάτων. Όμως αυτό ήταν ανέκαθεν μια μέθοδος περιήγησης που προορίζονταν για εργασιακή χρήση. Με την εξέλιξη των τεχνολογιών και την εισαγωγή ολοένα και περισσότερων πολυμεσικών εφαρμογών στο περιβάλλον του Η/Υ, ήταν αναπόφευκτη η μετάβαση σε μια εποχή όπου ο Η/Υ έχει εφαρμογή σε κάθε σπίτι ανεξαρτήτως του επαγγέλματος του ιδιοκτήτη. Αναγνωρίζοντας την αλλαγή αυτή και το γεγονός πως παραμένουν περιθώρια βελτίωσης από την πλευρά του συστήματος ως προς την επικοινωνία μεταξύ του χρήστη και του υπολογιστή, το πρόγραμμα αυτό έχει ως στόχο την εκτέλεση εφαρμογών ανεξαρτήτως της συσκευής εισόδου του χρήστη. Επομένως, απώτερος σκοπός είναι η αξιοποίηση των νέων περιφερειακών συσκευών στην εκτέλεση και περιήγηση του λειτουργικού συστήματος αλλά και την υλοποίηση νέων διαδικασιών εκτέλεσης εφαρμογών από προϋπάρχουσες συσκευές, όπως το πληκτρολόγιο και το ποντίκι. Για την υλοποίηση αυτής της εφαρμογής είναι απαραίτητο να γίνει χρήση τεχνολογιών για την γραφική διεπαφή χρήστη και την είσοδο δεδομένων από τις περιφερειακές συσκευές στο σύστημα. Για την εξυπηρέτηση αυτών των αναγκών γίνεται κυρίως χρήση της βιβλιοθήκης Qt6, του XInput και της βιβλιοθήκης των Windows.

Chapter B

Ανάπτυξη εφαρμογών σε Qt6

B.1 Τι είναι το Qt;

Το Qt είναι μια δημοφιλής βιβλιοθήκη ανάπτυξης εφαρμογών διεπαφής χρήστη το οποίο είναι διαθέσιμο σε C++ και Python. Με το Qt είναι εφικτή η υλοποίηση cross-platform εφαρμογών με το τελικό αποτέλεσμα να είναι μια αξιοπρεπής διεπαφή χρήστη που λειτουργεί αποτελεσματικά και αξιόπιστα, υποστηρίζοντας μια πληθώρα συστημάτων όπως Windows, Linux, MacOS, Android και ενσωματωμένα συστήματα. Παρακάτω γίνεται περιγραφή της λειτουργίας της βιβλιοθήκης αυτής όπου στην περίπτωση αυτή θα γίνει χρήση παραδειγμάτων σε γλώσσα προγραμματισμού C++ μιας και είναι η γλώσσα στην οποία έχει γραφεί το παρόν πρόγραμμα.

B.2 Ένα βασικό πρόγραμμα σε Qt6

Για την δημιουργία ενός παραθύρου Qt6 πρέπει πρώτα να δημιουργηθεί ένα αντικείμενο τύπου QApplication και στην συνέχεια να αρχικοποιηθεί ένα αντικείμενο QWidget το οποίο θα αποτελεί το παράθυρο της εφαρμογής πάνω στο οποίο θα προστεθούν τα υπόλοιπα στοιχεία της γραφικής διεπαφής για τις λειτουργίες του προγράμματος. Παρακάτω βλέπουμε

ένα παράδειγμα με την αντίστοιχη περιγραφή μεταφρασμένη σε κώδικα.

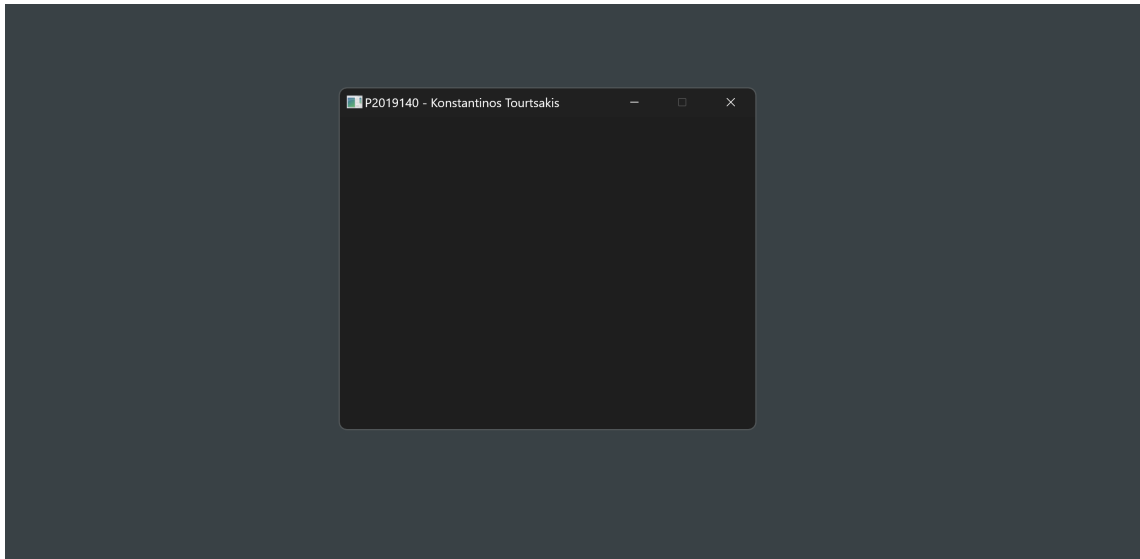
```
#include <QApplication>
#include <QWidget>

class MyWidget : public QWidget
{
public:
    MyWidget(QWidget *parent = nullptr) : QWidget(parent)
    {
        setFixedSize(400, 300);
        setWindowTitle("P2019140 - Konstantinos Tourtsakis");
    }
};

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    MyWidget widget;
    widget.show();

    return app.exec();
}
```



Η κλάση `MyWidget` κληρονομεί την κλάση `QWidget`. Το παράθυρο το οποίο δημιουργείται εμφανίζεται σε πλήρες μέγεθος και στην συνέχεια επιστρέφεται το αντικείμενο της εφαρμογής το οποίο το διαχειρίζεται η βιβλιοθήκη κατά την έξοδο εκτέλεσης του προγράμματος.

B.3 Προσθήκη στοιχείων γραφικής διεπαφής σε Qt6

Κάθε εφαρμογή γραφικής διεπαφής παρέχει στοιχεία μέσω των οποίων γίνεται η διαχείριση των δεδομένων που επεξεργάζεται και η εκτέλεση των λειτουργιών του. Συνήθως τα δεδομένα αυτά δεν είναι τίποτε άλλο από τους βασικούς τύπους δεδομένων που υποστηρίζουν οι γλώσσες προγραμματισμού: `int`, `bool`, `float` και `string`. Επιπλέον υπάρχουν στοιχεία με τα οποία εξυπηρετείται αποτελεσματικότερα ο σκοπός του προγράμματος, είτε λόγω ευκολίας είτε λόγω κατανόησης από τον χρήστη. Παρακάτω βλέπουμε τα στοιχεία που αξιοποιεί το πρόγραμμα της εργασίας για την επίτευξη του σκοπού του.

B.3.1 Δημιουργία `QPushButton`

Ένα `QPushButton` είναι ένα κουμπί το οποίο έχει την ιδιότητα εκτέλεσης λειτουργιών. Για την προσθήκη μιας λειτουργίας πάνω στο κουμπί αυτό χρειάζεται να γίνει η σύνδεση

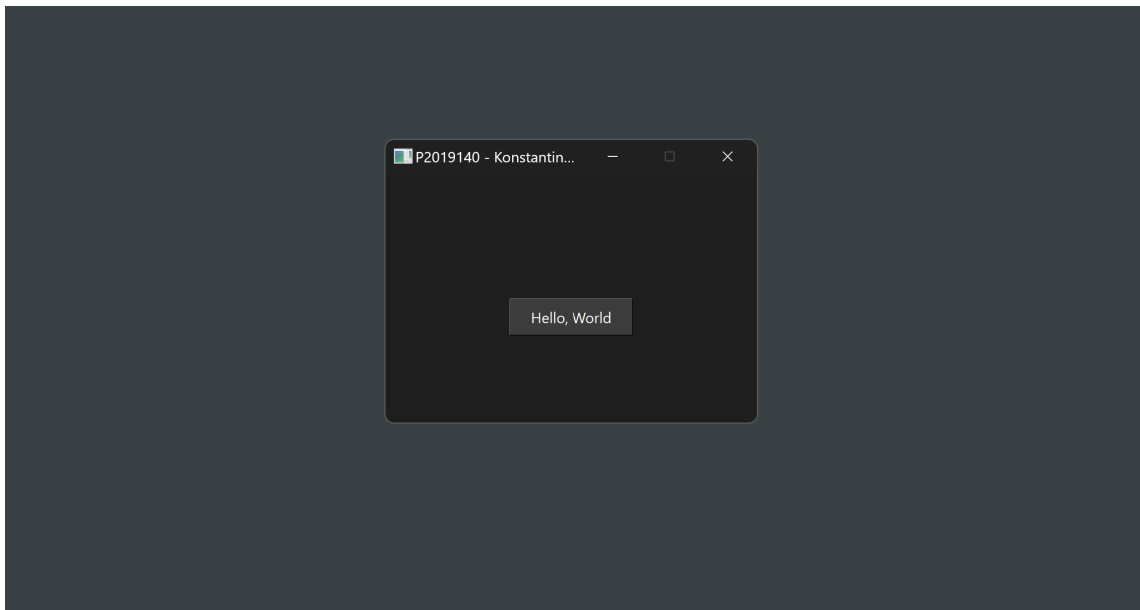
μεταξύ του αντικειμένου αυτού και μιας μεθόδου που ανήκει στην κλάση που κληρονομεί το QWidget. Ορίζεται το signal το οποίο υποστηρίζει το κάθε αντικείμενο μέσω του οποίου θα γίνει η κλήση της μεθόδου που ονομάζεται slot από το Qt. Τα signals που υποστηρίζουν τα στοιχεία του Qt είναι διαφορετικά, ανάλογα με τους στόχους που προσπαθεί να πετύχει το κάθε ένα από αυτά. Επομένως ένα στοιχείο QPushButton μπορεί να έχει περισσότερα ή λιγότερα signals σε σύγκριση με ένα QComboBox. Παρακάτω βλέπουμε ένα παράδειγμα σε κώδικα.

```
class MyWidget : public QWidget
{
public:
    MyWidget(QWidget *parent = nullptr) : QWidget(parent)
    {
        setFixedSize(300, 200);
        setWindowTitle("P2019140 - Konstantinos Tourtsakis");

        QPushButton *button = new QPushButton("Hello, World", this);
        button->setGeometry(100, 100, 100, 30);

        connect(button, &QPushButton::clicked, this, &MyWidget::PrintHello);
    }

public slots:
    void PrintHello()
    {
        std::cout << "Hello, World!" << std::endl;
    }
};
```



Στο παράδειγμα γίνεται ή δημιουργία και ή αρχικοποίηση κουμπιού με το όνομά του και στην συνέχεια η σύνδεση. Στην σύνδεση καλείται η μέθοδος connect στην οποία ορίζεται το signal το οποίο θα πυροδοτήσει την κλήση του slot που έχει ανατεθεί στο αντικείμενο. Στην προκειμένη περίπτωση έχουμε ορίσει το QPushButton::clicked signal το οποίο συνδέει το κουμπί με την μέθοδο PrintHello.

B.3.2 Δημιουργία γραφικού πλαισίου

Ένα layout είναι ένα πλαίσιο στο οποίο μπορούν να τοποθετηθούν άλλα στοιχεία του Qt, όπως το QPushButton που προαναφέρθηκε. Το Qt6 παρέχει 3 βασικά είδη layouts. Το QVBoxLayout, το QHBoxLayout και το QGridLayout. Τα πρώτα δύο παρέχουν ένα παρόμοιο πλαίσιο με την μόνη τους διαφορά να βρίσκεται στην κατεύθυνση των στοιχείων μέσα στο πλαίσιο. Επομένως, ένα QVBoxLayout χρησιμοποιείται για στοιχεία που θα τοποθετηθούν κάθετα (vertical) και ένα QHBoxLayout θα χρησιμοποιηθεί για στοιχεία που πρόκειται να τοποθετηθούν οριζόντια (horizontal). Τέλος, ένα QGridLayout χρησιμοποιείται για την τοποθέτηση στοιχείων σε μορφή πίνακα. Το πλαίσιο αυτό διαθέτει θέσεις που αναπαριστούν ένα σημείο σε έναν δισδιάστατο χώρο. Κάθε σημείο έχει μια θέση η οποία

είναι μοναδική και είναι προσβάσιμη μέσω της τιμής της σειράς και της στήλης στην οποία βρίσκεται. Παρακάτω βλέπουμε κώδικα με την χρήση ενός QVBoxLayout και την προσθήκη ενός QPushButton σε αυτό.

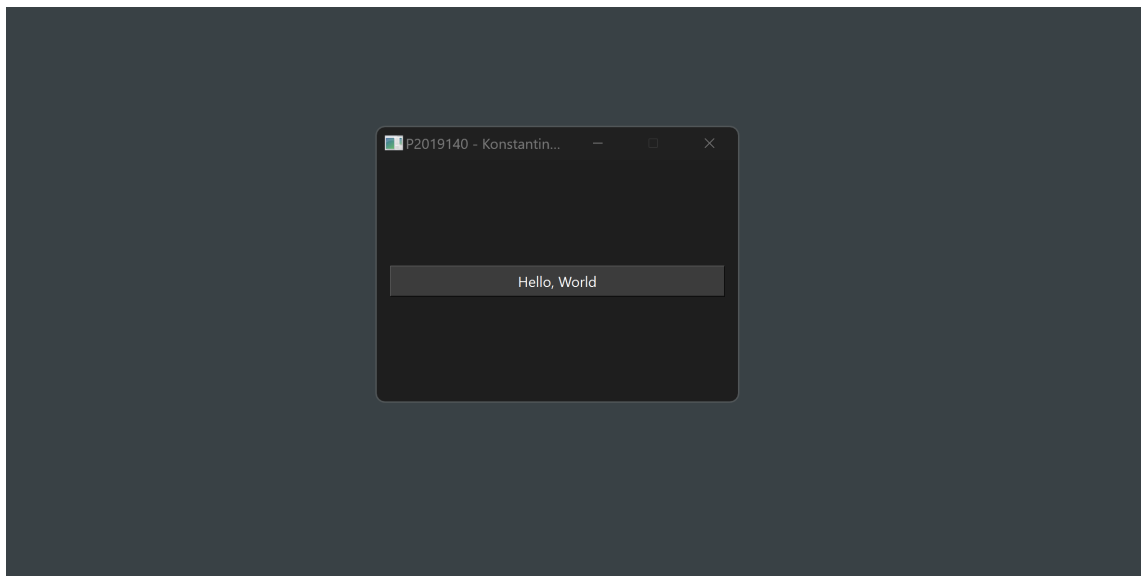
```
MyWidget(QWidget *parent = nullptr) : QWidget(parent)
{
    setFixedSize(300, 200);
    setWindowTitle("P2019140 - Konstantinos Tourtsakis");

    QVBoxLayout *layout = new QVBoxLayout(this);

    QPushButton *button = new QPushButton("Hello, World", this);
    button->setGeometry(100, 100, 100, 30);

    layout->addWidget(button);

    connect(button, &QPushButton::clicked, this, &MyWidget::PrintHello);
}
```



Κάθε στοιχείο τύπου QWidget προστίθεται πάνω στο layout με την κλήση της μεθόδου addWidget και αντίστοιχα η αφαίρεση του γίνεται με την μέθοδο removeWidget.

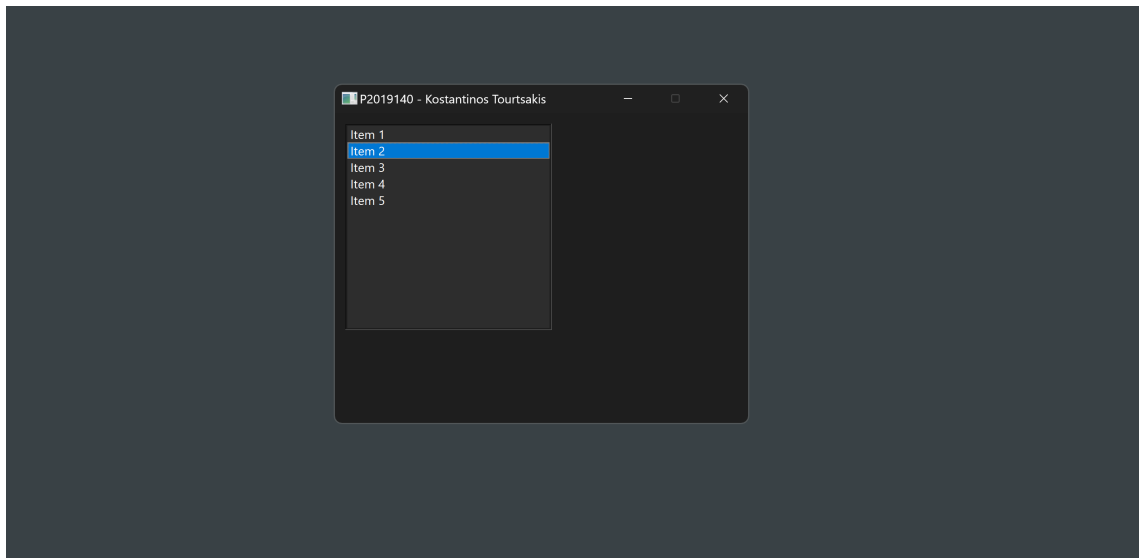
B.3.3 Δημιουργία λίστας αντικειμένων

Μια λίστα QWidget μπορεί να αποθηκεύσει στην μνήμη στοιχεία τύπου QListWidgetItem. Τα στοιχεία αυτά είναι αντικείμενα του Qt τα οποία δεν μπορούν να αποθηκευθούν σε κάποια άλλη

```
#include <QListWidget>
#include <QListWidgetItem>

class MyWidget : public QWidget
{
public:
    MyWidget(QWidget *parent = nullptr) : QWidget(parent)
    {
        setFixedSize(400, 300);
        setWindowTitle("P2019140 - Konstantinos Tourtsakis");

        QListWidget *list_widget = new QListWidget(this);
        list_widget->addItem(new QListWidgetItem("Item 1"));
        list_widget->addItem(new QListWidgetItem("Item 2"));
        list_widget->addItem(new QListWidgetItem("Item 3"));
        list_widget->addItem(new QListWidgetItem("Item 4"));
        list_widget->addItem(new QListWidgetItem("Item 5"));
        list_widget->setGeometry(10, 10, 200, 200);
    }
};
```

B.3.4 Διάβασμα αρχείων από directory

Το διάβασμα αρχείων από ένα directory γίνεται μέσω της `QDir` κλάσης στην οποία αρχικοποιείται ένα αντικείμενο με το path του directory του οποίου θέλουμε να διαβάσουμε. Στην συνέχεια αποθηκεύουμε τα αρχεία του directory σε μια λίστα από `QString`s και τα προσπελαύνουμε για την προσθήκη τους σε ένα `QListWidget` με στόχο την προβολή τους στον χρήστη.

```
#include <QDir>
#include <QStringList>

class MyWidget : public QWidget
{
public:
    MyWidget(QWidget* parent = nullptr) : QWidget(parent)
    {
        setFixedSize(300, 200);
        setWindowTitle("P2019140 - Konstantinos Tourtsakis");

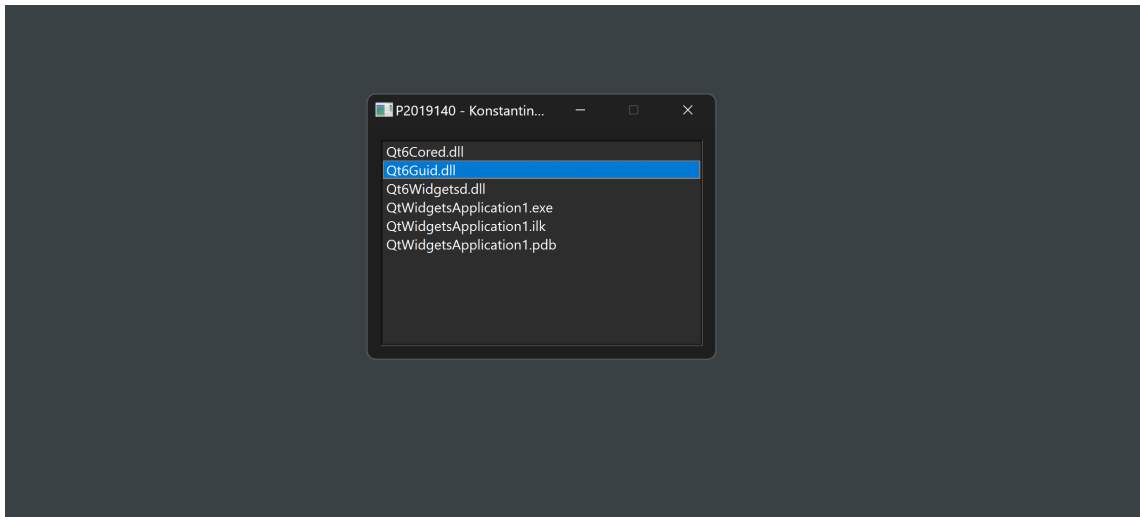
        QVBoxLayout* layout = new QVBoxLayout(this);
        QListWidget* list_widget = new QListWidget(this);
```

```
layout->addWidget(listWidget);

QDir directory("C:\\Users\\kosta\\Documents\\Git\\Thesis\\source\\x64\\Debug");

QStringList files = directory.entryList(QDir::Files);
for (const QString& file : files)
{
    list_widget->addItem(file);
}

};
```



B.3.5 Δημιουργία input field

Βασικό στοιχείο κάθε εφαρμογής γραφικής διεπαφής αποτελεί ένα input field μιας και επιτρέπει στον χρήστη να πληκτρολογήσει δεδομένα εισόδου για την εκτέλεση μιας λειτουργίας του προγράμματος. Για τον σκοπό αυτό το Qt παρέχει τα αντικείμενα τύπου `QLineEdit`. Όπως και τα `QPushButton`, τα αντικείμενα αυτά αρχικοποιούνται και στην συνέχεια προστίθενται πάνω σε ένα πλαίσιο μέσω της μεθόδου `addWidget`.

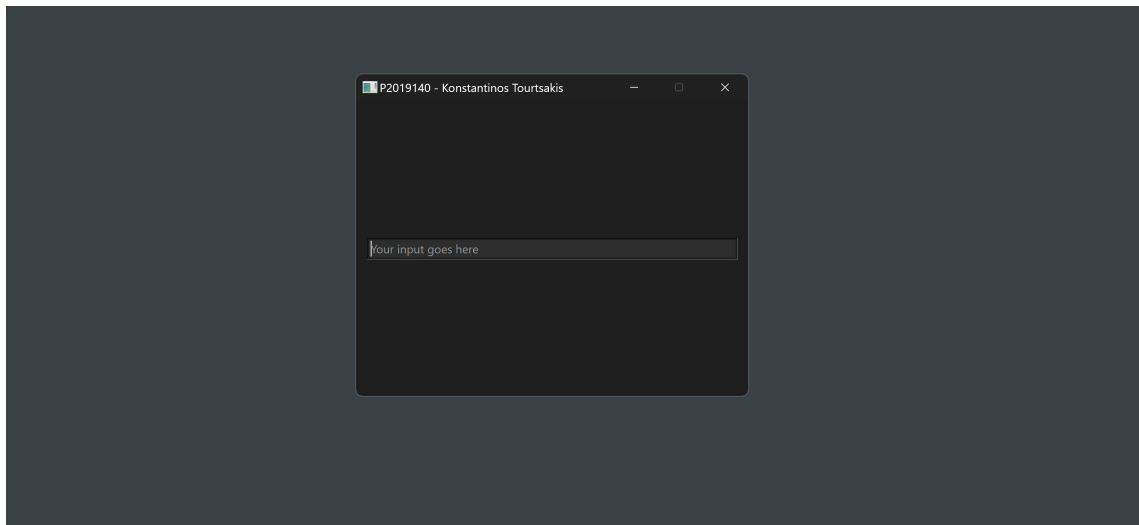
```
#include <QLineEdit>
```

```
class MyWidget : public QWidget
{
public:
    MyWidget(QWidget* parent = nullptr) : QWidget(parent)
    {
        setFixedSize(400, 300);
        setWindowTitle("P2019140 - Konstantinos Tourtsakis");

        QLineEdit* lineEdit = new QLineEdit(this);
        lineEdit->setPlaceholderText("Your input goes here");

        QVBoxLayout* layout = new QVBoxLayout(this);
        layout->addWidget(lineEdit);

        setLayout(layout);
    }
};
```



B.3.6 Αποθήκευση στοιχείων στον δίσκο

Κάθε εφαρμογή στις ημέρες μας χρειάζεται να αποθηκεύει την κατάσταση στην οποία βρίσκεται την τελευταία φορά που εκτελέστηκε με στόχο να γίνει πιο κατανοητή η χρήση

του. Για τον σκοπό αυτό, το Qt φροντίζει να παρέχει τα απαραίτητα εργαλεία για την αποθήκευση των στοιχείων της εφαρμογής βελτιώνοντας με αυτόν τον τρόπο τον πηγαίο κώδικα που γράφεται για την προσθήκη της λειτουργίας αυτής. Παρακάτω βλέπουμε μια μέθοδο που αποθηκεύει τα στοιχεία ενός Qt6 προγράμματος.

```
void SaveSettings()
{
    QSettings settings("Company_name", "Application_title");
    settings.beginGroup("group_name");

    settings.setValue("WindowSize", window()->size());
    settings.setValue("Username", "JohnDoe");

    settings.endGroup("group_name");
}
```

Αρχικά δημιουργείται ένα αντικείμενο τύπου QSettings. Ακολουθεί ο ορισμός της ομάδας στην οποία θα γίνουν οι αλλαγές και στην συνέχεια ορίζονται οι τιμές πάνω στις οποίες θα γραφούν οι μεταβλητές του προγράμματος. Στο τέλος επαναρχικοποιείται η ομάδα καλώντας την μέθοδο endGroup.

B.3.7 Ανάκτηση εικονιδίων συστήματος αρχείων

Όπως είναι γνωστό τα γραφικά περιβάλλοντα στηρίζονται πολύ στην χρήση εικονιδίων για την διευκόλυνση του χρήστη κατά την περιήγησή του μιας και είναι πολύ πιο αποτελεσματική η συσχέτιση μιας λειτουργίας με μια εικόνα παρά με κάποιο κείμενο. Επομένως, είναι σημαντικό να υπάρχει μια συνάρτηση που επιστρέφει το εικονίδιο ενός αρχείου δυναμικά, ώστε ο χρήστης να καταλαβαίνει από την αρχή την λειτουργία που κρύβεται πίσω από την ενεργοποίηση του εικονιδίου αυτού. Παρακάτω βλέπουμε την συνάρτηση αυτή.

```
QIcon GetFileIcon(const QString& file_path)
{
    SHFILEINFO shfi;
    memset(&shfi, 0, sizeof(SHFILEINFO));
```

```
if (SHGetFileInfo(reinterpret_cast<const wchar_t*>(file_path.utf16()), 0, &shfi, sizeof(
{
    QPixmap pixmap = QPixmap::fromImage(QImage::fromHICON(shfi.hIcon)).scaled(QSize(Perce
    QIcon icon(pixmap);

    // Cleanup the icon resource
    DestroyIcon(shfi.hIcon);

    return icon;
}

return QIcon();
}
```

Επειδή η εφαρμογή στην οποία αναφέρεται το παρόν κείμενο προορίζεται για χρήση σε συστήματα Windows, η συγκεκριμένη συνάρτηση στηρίζεται πάνω σε αυτά. Αρχικά δημιουργείται ένα struct τύπου SHFILEINFO πάνω στο οποίο θα αποθηκευτούν τα δεδομένα του αρχείου, συμπεριλαμβανομένου και του εικονιδίου. Στην συνέχεια γίνεται έλεγχος για την ανάκτηση του εικονιδίου από το σύστημα. Αφού η ανάκτηση γίνει με επιτυχία, δημιουργείται ένα QPixmap με το εικονίδιο του αρχείου. Στην συνέχεια το pixmap χρησιμοποιείται για την δημιουργία του QIcon εικονιδίου το οποίο και θα επιστραφεί από την συνάρτηση. Αν το εικονίδιο δεν βρεθεί η συνάρτηση θα επιστρέψει ένα προεπιλεγμένο QIcon.

Chapter C

Η βιβλιοθήκη *XInput*

C.1 Τι είναι το *XInput*;

Το *XInput* είναι μια βιβλιοθήκη διαχείρισης εντολών εισόδου για συσκευές χειρισμού βιντεοπαιχνιδιών κατασκευασμένες από την Microsoft. Οι συγκεκριμένες συσκευές είναι χειριστήρια Xbox οποιασδήποτε γεννιάς.

C.2 Η κλάση του χειριστηρίου

Για την διαχείριση των εντολών εισόδου του χειριστηρίου έχει δημιουργηθεί η αντίστοιχη κλάση. Η κλάση περιλαμβάνει δεδομένα που αφορούν καταστάσεις του χειριστηρίου, μεθόδους σχετικά με τον έλεγχο καταστάσεων των κουμπιών του αλλά και μεθόδους που ελέγχουν την συνδεσιμότητα της συσκευής. Παρακάτω βλέπουμε αναλυτικότερα τον ορισμό της κλάσης και τα στοιχεία που την απαρτίζουν.

```
class XBOXController
{
private:
    XINPUT_STATE controller_state;
    int controller_number;
```

```
public:
    float left_trigger = 0.0f;
    float right_trigger = 0.0f;

    XBOXController(int playerNumber);
    XINPUT_STATE GetState();
    bool IsConnected();
    void Vibrate(const int left_val = 0, const int right_val = 0);
    SHORT GetLeftAnalogX();
    SHORT GetLeftAnalogY();
    SHORT GetRightAnalogX();
    SHORT GetRightAnalogY();
    bool IsButtonDown(const int button);
    bool IsButtonJustUp(const int button);
    bool IsButtonJustDown(const int button);
};
```

Chapter D

Διεπαφή χρήστη διαχείρισης και εκτέλεσης εφαρμογών μέσω πολυτροπικών περιφερειακών συσκευών

D.1 Περιγραφή λογισμικού

Το λογισμικό αυτό αποτελείται από 3 μέρη τα οποία ουσιαστικά αποτελούν και τις περιφερειακές συσκευές τις οποίες υποστηρίζει για την επίλυση του προβλήματος. Αυτά είναι τα εξής:

D.2 Εκτέλεση εφαρμογών με το ποντίκι

Το ποντίκι όπως είναι γνωστό είναι ένα εξαιρετικά απλό μέσο με το οποίο επικοινωνεί ο χρήστης με τον υπολογιστή και επιτρέπει την χρήση ενός μόνο χεριού για την περιήγηση μέσα σε αυτόν. Επομένως, η υποστήριξη της συσκευής αυτής από την εφαρμογή γίνεται μέσω της γραφικής διεπαφής από όπου και είναι εύκολα προσβάσιμα τα στοιχεία που

CHAPTER D. ΔΙΕΠΑΦΉ ΧΡΗΣΤΗ ΔΙΑΧΕΪΡΙΣΗΣ ΚΑΙ ΕΚΤΕΛΕΣΗΣ ΕΦΑΡΜΟΓΩΝ ΜΕΣΩ ΠΟΛΥΤΡΟΠΙ

διαχειρίζονται την εκτέλεση, προσαρμογή και διαχείριση των προγραμμάτων που είναι εγκατεστημένα στο περιβάλλον του χρήστη. Για τον σκοπό αυτό έχουν προστεθεί τα κατάλληλα στοιχεία γραφικής διεπαφής που υλοποιούν την προβλεπόμενη από τον τελικό χρήστη λειτουργικότητα της εφαρμογής. Πιο συγκεκριμένα, η γραφική διεπαφή έχει χωριστεί σε τρία κομμάτια. Αυτά υλοποιούνται μέσω του Qt σε καρτέλες οι οποίες στην συνέχεια προστίθενται στο κεντρικό Widget που διαχειρίζεται την βασικό γραφικό στρώμα του προγράμματος.

Chapter E

Σχήματα και Πίνακες

ΤΑ σχήματα είναι απαραίτητα πολλές φορές για να κατανοήσει ο αναγνώστης καλύτερα το περιεχόμενο του κειμένου μας. Μερικές φορές είναι απαραίτητα και για το κείμενό μας. Επ' ουδενί όμως, δεν επιτρέπεται η αντιγραφή (scan) σχημάτων άλλων συγγραφέων που υπάρχουν σε βιβλία, επιστημονικές εργασίες κτλ.

Όπως τα σχήματα, χρήσιμοι είναι και οι πίνακες γιατί μπορεί κάποιος με εύκολο τρόπο να βρει χρήσιμη πληροφορία, όπως π.χ. παράθεση πειραματικών αποτελεσμάτων που από τη φύση της είναι δύσκολη και ίσως δυσνόητη όταν βρίσκεται εντός του κειμένου.

E.1 Σχήματα

Στη συνέχεια θα δωθούν μερικά παραδείγματα μορφής σχημάτων.

Το “κλασικό” σχήμα είναι αυτό που απεικονίζεται στο Σχήμα E.1. Προφανώς, το σχήμα μπορεί να θέλει να είναι λίγο μικρότερο (π.χ., όπως στο Σχήμα E.2) αλλά φροντίζουμε να μην είναι το πλάτος του μικρότερο από το $1/2$ του πλάτους του κειμένου. Σε κάθε περίπτωση, τα σχήματα αριθμούνται ξεχωριστά για κάθε κεφάλαιο αρχίζοντας από το 1 με πρόθεμα το νούμερο του κεφαλαίου.

Υπάρχουν περιπτώσεις που μπορεί κάποιος να θέλει δύο ίδια σχήματα δίπλα-δίπλα,

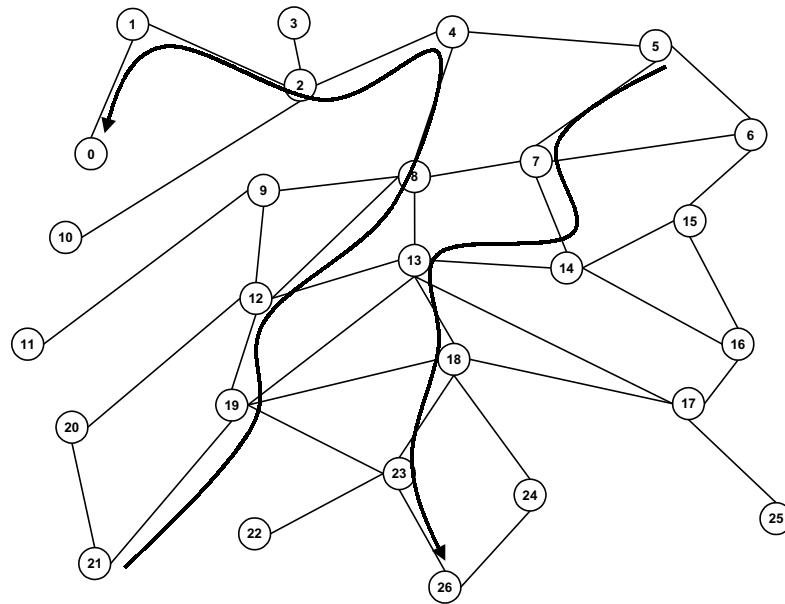


Figure E.1: Ένα δίκτυο.

όπως είναι η περίπτωση του Σχήματος E.3. Προφανώς μπορεί να γίνει ξεχωριστή αναφορά στα δύο υπο-σχήματα E.1.α και E.1.β με τον αυτό τρόπο. Το κείμενο για τα δύο υποσχήματα είναι προαιρετικό αλλά η αρίθμηση υποχρεωτική.

Όπως προηγουμένως, μπορεί να είναι επιθυμητή η αλλαγή του μεγέθους των σχημάτων, όπως φαίνεται στο Σχήμα E.4.

Τέλος, υπάρχει το ενδεχόμενο να είναι αναγκαία η παρουσίαση κάποιων σχημάτων σε τετράδα, όπως είναι το Σχήμα E.5.

Με όμοιο τρόπο όπως προηγουμένως μπορεί να επιτευχθεί αλλαγή του μεγέθους.

E.2 Πίνακες

Οι πίνακες ομοιάζουν με το κλασικό σχήμα στη μορφή και την αναφορά σε αυτούς, με μόνη διαφορά πως προηγούνται αντί να έπονται, όπως φαίνεται και στο παράδειγμα του

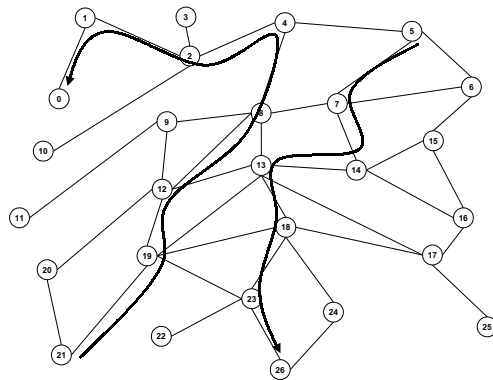


Figure E.2: Το ίδιο δίκτυο που απεικονίζεται στο Σχήμα E.1 αλλά λίγο μικρότερο.

Πίνακα E.1.

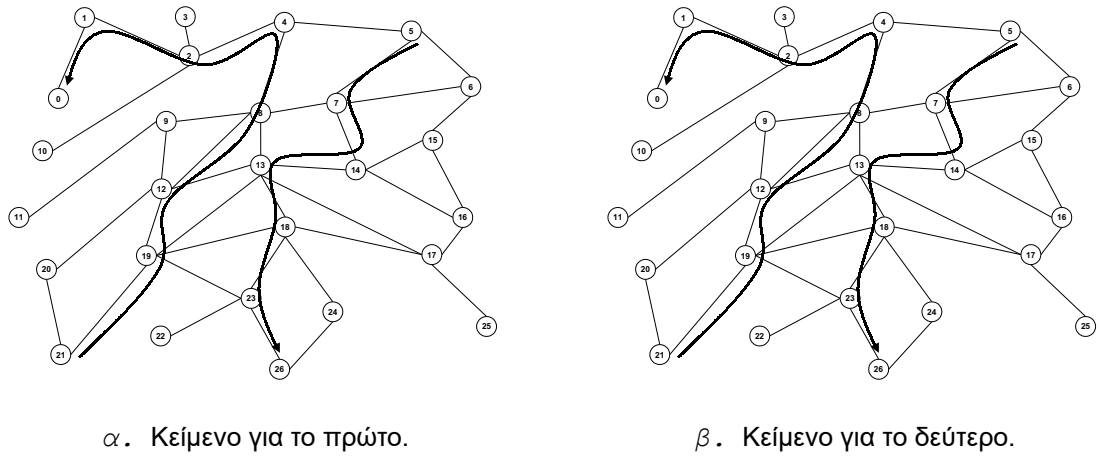


Figure E.3: Το ίδιο δίκτυο σε διπλή απεικόνιση.

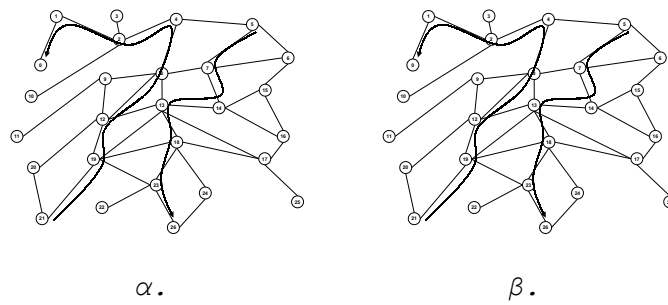
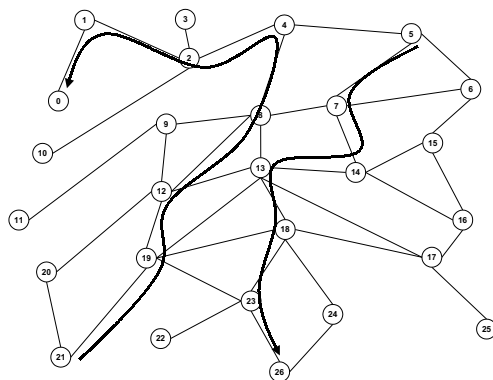
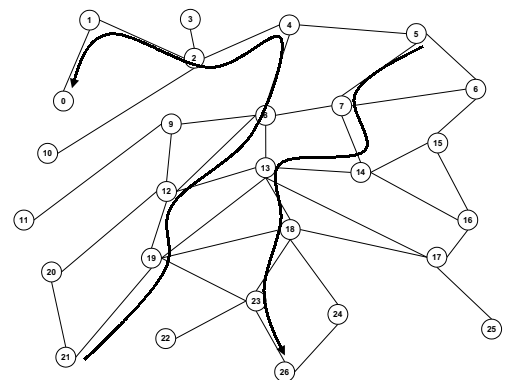


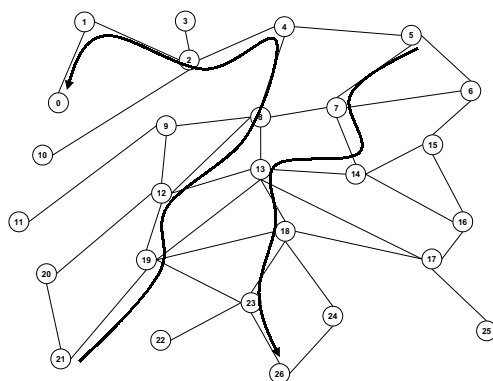
Figure E.4: Το ίδιο δίκτυο σε διπλή απεικόνιση σε σμίκρυνση και δίχως κείμενο για τα υπο-σχήματα.



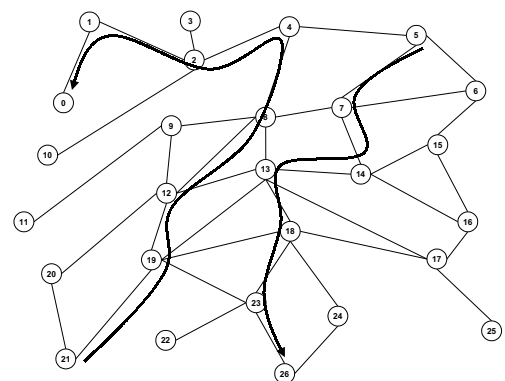
α. Κείμενο για το πρώτο.



β. Κείμενο για το δεύτερο.



γ. Κείμενο για το τρίτο.



δ. Κείμενο για το τέταρτο.

Figure E.5: Το ίδιο δίκτυο σε τετραπλή απεικόνιση.

Table E.1: Παράδειγμα Πίνακα.

	s_0	s_1	s_2	s_3	s_4	s_5	s_6
$f_0(s_\chi)$	1	2	3	4	5	6	0
$f_1(s_\chi)$	1	3	5	0	2	4	6
$f_2(s_\chi)$	4	0	3	6	2	5	1
$f_3(s_\chi)$	3	0	4	1	5	2	6
$f_4(s_\chi)$	2	5	1	4	0	3	6
$f_5(s_\chi)$	2	3	4	5	6	0	1
$f_6(s_\chi)$	6	4	2	0	5	3	1
$f_7(s_\chi)$	1	0	6	5	4	3	2
$f_8(s_\chi)$	5	6	0	1	2	3	4
$f_9(s_\chi)$	3	2	1	0	6	5	4

Chapter F

Τελευταία Μέρη

Η κύρια δομή της Πτυχιακής Εργασίας δεν μπορεί να προκαθοριστεί και ούτε θα ήταν σωστό ώστε να υπόκειται πάντα στη δημιουργικότητα του φοιτητή υπό τη συμβουλή του επιβλέποντα. Είναι όμως σημαντικό να υπάρχει ένα κεφάλαιο συμπερασμάτων στο οποίο να γίνεται τελικά σύνοψη της εργασίας και των συμπερασμάτων αυτής. Βέβαια, μια τέτοια σύνοψη έλαβε χώρα και στην Εισαγωγή. Όμως τότε ο αναγνώστης δεν είχε διαβάσει την εργασία και ο κύριος σκοπός ήταν να του προκαλέσει το ενδιαφέρον αλλά και να τον εισάγει ομαλά. Εδώ, όμως, ο σκοπός είναι να τον βοηθήσει όλα όσα διάβασε να τα καταχωρήσει στο μυαλό του ακόμα καλύτερα.

Μετά τα συμπεράσματα, και το τέλος ουσιαστικά της Πτυχιακής Εργασίας, υπάρχει ένας αριθμός από κεφάλαια τα οποία είναι χρήσιμα για τον αναγνώστη και ουσιαστικά αποτελούν το "κερασάκι στην τούρτα."

- Καταρχάς μπορεί να είναι κάποια επιπλέον κεφάλαια Παραρτήματος τα οποία όμως απαριθμούνται αυτόνομα (δεν έχουν συσχέτιση με την αρίθμηση των κεφαλαίων του κυρίου μέρους).
- Είναι οπωσδήποτε το κεφάλαιο της βιβλιογραφίας το οποίο αριθμεί της βιβλιογραφικές αναφορές κατά αύξοντα αριθμό πρώτης εμφάνισής τους στο κείμενο. Αν π.χ. στο σημείο αυτό γινόταν η πρώτη αναφορά τότε εν μέσω του κειμένου θα παρεμβάλλονταν

το [16]. Η βιβλιογραφία μπορεί να είναι είτε στα αγγλικά είτε στα ελληνικά ανάλογα με το προς αναφορά κείμενο.

- Το κεφάλαιο των συντμήσεων είναι ιδιαίτερα σημαντικό καθώς η Επιστήμη της Πληροφορικής έχει πολλούς ξενικούς όρους που χρησιμοποιούμε καθημερινά με συντετμημένη μορφή. Για παράδειγμα, η υπηρεσία ονομάτων περιοχής (Domain Name Services – DNS). Σε τέτοιες περιπτώσεις δίνουμε την ελληνική μετάφραση με πλάγια γράμματα και στην παρένθεση έχουμε την αγγλική έκφραση και τη σύντμηση. Στη συνέχεια είμαστε ελεύθεροι να χρησιμοποιήσουμε την ελληνική έκφραση ή τον συντμημένο αγγλικό τύπο. Προσοχή, όμως, γιατί χρησιμοποιούμε μόνο ένα από τα δύο! Το ίδιο κάνουμε και με τα ελληνικά, βάζοντας σε παρένθεση όμως μόνο τη σύντμηση. Π.χ., Οργανισμός Τηλεπικοινωνιών Ελλάδος (ΟΤΕ). Στο κεφάλαιο των συντμήσεων έχουμε διαφορετικά μέρη για τις ελληνικές και τις ξενικές συντμήσεις αλλά με αλφαβητική διάταξη.
- Το γλωσσάρι είναι ουσιαστικά η μετάφραση των ξενικών όρων που συναντάμε στο κείμενο (είτε τους χρησιμοποιούμε γιατί είθισται είτε όχι). Το γλωσσάρι δεν είναι μόνο λεξικό αλλά είναι θεμιτό να περιέχει και μια μικρή ερμηνεία του όρου.
- Τέλος, είναι σημαντικό το ευρετήριο. Στο κεφάλαιο αυτό δίνονται με αλφαβητική διάταξη οι σελίδες που απαντώνται οι σημαντικότεροι όροι μέσα στο κείμενο. Όχι κατ' ανάγκη όλοι οι σημαντικοί όροι αλλά κυρίως εκείνοι που έχουν να κάνουν με την ουσία της Πτυχιακής Εργασίας.

Παράρτημα Α'

Ενδεικτικό παράδειγμα παραρτήματος. Έπεται το κεφάλαιο της βιβλιογραφίας.

Bibliography

- [1] Authors, title, information about the book, paper journal.
- [2] The Qt Company Ltd, QListWidget class, Qt6 Documentation, <https://doc.qt.io/qt-6/qlistwidget.html>
- [3] The Qt Company Ltd, QPushButton class, Qt6 Documentation, <https://doc.qt.io/qt-6/qlistwidget.html>
- [4] The Qt Company Ltd, QLineEdit class, Qt6 Documentation, <https://doc.qt.io/qt-6/qlistwidget.html>
- [5] The Qt Company Ltd, QIcon class, Qt6 Documentation, <https://doc.qt.io/qt-6/qlistwidget.html>
- [6] The Qt Company Ltd, QApplication class, Qt6 Documentation, <https://doc.qt.io/qt-6/qapplication.html>
- [7] The Qt Company Ltd, QWidget class, Qt6 Documentation, <https://doc.qt.io/qt-6/qwidget.html>
- [8] The Qt Company Ltd, QVBoxLayout class, Qt6 Documentation, <https://doc.qt.io/qt-6/qvboxlayout.html>
- [9] The Qt Company Ltd, QHBoxLayout class, Qt6 Documentation, <https://doc.qt.io/qt-6/qhboxlayout.html>

- [10] The Qt Company Ltd, QGridLayout class, Qt6 Documentation, <https://doc.qt.io/qt-6/qgridlayout.html>
- [11] The Qt Company Ltd, QDir class, Qt6 Documentation, <https://doc.qt.io/qt-6/qdir.html>
- [12] The Qt Company Ltd, QSettings class, Qt6 Documentation, <https://doc.qt.io/qt-6/qsettings.html>
- [13] The Qt Company Ltd, QIcon class, Qt6 Documentation, <https://doc.qt.io/qt-6/qicon.html>
- [14] Ivan Akulov, Extracting icon using WinAPI in Qt app, Stack Overflow ερώτηση, <https://stackoverflow.com/questions/12459145/extracting-icon-using-winapi-in-qt-app>
- [15] Microsoft, Getting Started With XInput in Windows applications, Windows Win32 Documentation, <https://learn.microsoft.com/en-us/windows/win32/xinput/getting-started-with-xinput>
- [16] Minalien, Xbox 360 Controller Input in C++ with XInput, Code Project article, <https://www.codeproject.com/articles/26949/xbox-360-controller-input-in-c-with-xinput>

Συντμήσεις

<i>DNS</i>	Domain Name Server
<i>DSU/CSU</i>	Data Service Unit/Channel Service Unit
<i>HDL</i>	High Level Data Link
<i>FTP</i>	File Transfer Protocol
<i>NVRAM</i>	Nonvolatile RAM
<i>PPP</i>	Point-to-Point
<i>TCP</i>	Transport Control Protocol
<i>UDP</i>	User Datagram Protocol
<i>UTP</i>	Unshield Twisted Pair
<i>WAN</i>	Wide Area Network

Γλωσσάρι Ενικών Όρων

<i>Access Permissions</i>	Δικαιώματα Πρόσβασης
<i>Adapter</i>	Προσαρμογέας
<i>Administrator</i>	Επιβλέπων
<i>Agent</i>	Πράκτορας
<i>Background Process</i>	Διεργασία Παρασκηνίου
<i>Capture</i>	Καταγράφω
<i>Capture Filter</i>	Φίλτρο Καταγραφής
<i>Command Mode</i>	Κατάσταση Εντολών
<i>Configuration</i>	Καθορισμός Παραμέτρων
<i>Configuration Files</i>	Αρχεία Παραμέτρων
<i>Connector</i>	Σύνδεσμος
<i>Console Port</i>	Θύρα Κονσόλας
<i>Crossover Cable</i>	Διασταυρωμένο Καλώδιο
<i>Demodulation</i>	Αποδιαμόρφωση
<i>Desktop</i>	Επιφάνεια Εργασίας
<i>Device Files</i>	Αρχεία Συσκευών
<i>Display Filter</i>	Φίλτρο Προβολής

<i>Driver</i>	Πρόγραμμα Οδήγησης
<i>Download</i>	Κατέβασμα Αρχείων
<i>Foreground Process</i>	Διεργασία Προσκήνιου
<i>Forwarding</i>	Προώθηση
<i>Header</i>	Κεφαλίδα
<i>Hub</i>	Συζεύκτης
<i>Internet Lab</i>	Εργαστήριο Διαδικτύου
<i>Internet Lab Manual</i>	Εγχειρίδιο του Εργαστηρίου Διαδικτύου
<i>Kernel</i>	Πυρήνας
<i>Modem</i>	Διαμορφωτής-Αποδιαμορφωτής
<i>Modulation</i>	Διαμόρφωση
<i>Mount</i>	Αγκίστρωση Συστήματος Αρχείων
<i>Network Interface</i>	Διεπαφή Δικτύου
<i>Network Protocol</i>	Αναλυτές Δικτυακών Πρωτοκόλλων
<i>Analyzers</i>	
<i>Overwrite</i>	Επανεγγραφή
<i>Packet Sniffer</i>	Συλλέκτης Πακέτων
<i>Pathname</i>	Μονοπάτι
<i>Physical Layer</i>	Φυσικό Επίπεδο

Pin

Απόληξη

Prompt

Χαρακτήρα Εισόδου Εντολών

Rebooting

Επανεκκίνηση

Rollover Cable

Ανεστραμμένο Καλώδιο

Router

Δρομολογητής

Shell

Φλοιός

Straight-through

Απευθείας Καλώδιο

Cable

Upload

Ανέβασμα Αρχείων

Window Manager

Διαχειριστής Παραθύρων