

# Programmeerimine keeles C++

## ***Praktikum 5: lisateegi kasutamine: Qt raamistik.***

Märkus: Palun kasutage Qt raamistikku 4.4 või uuemat.

### **Üldised nõuded**

Kirjutage joonistusprogramm, mille abil saab joonistada tippe ning neid ühendavaid sirglõike. Meenutage neljanda praktikumi DynamicLine klassi ning seal juures kirja pandud rakenduseideed. NB! Ülesanne eeldab, et te taaskasutate sealset klassi DynamicLine.

Praktikumi alusprogrammis on toodud lihtne Qt projekt, mille projektifailid on .pro laiendiga. Qt Creator nimelises töökeskkonnas saab programmi triviaalsel viisil kompileerida ja käivitada. Käesolev praktikum on erandlik, sest lahendi esitamisel pole Makefile'i olemasolu kohustuslik. Lahenduste kontrollimisel kasutatakse Qt Creatorit. **NB! Erand ei laiene teistele praktikumidele.**

Teie ülesandeks on vastavalt ülesande tekstile täiendada alusprogrammi. Dokumenteerimise nõue kehtib klassidele ja funktsioonidele, mida te oma lahendustes kirjutate. Nagu tavaliselt, on viited ülesande lahendamiseks vajalikele materjalidele praktikumi veebilehel.

### **Osa 1 – Fikseerige alusprogrammi akna suurus**

Alusprogrammi Joonistamine akna suurst saab hetkel muuta, mis aga võib muuta keeruliseks hilisemad ülesanded. Fikseerige joonistusala suuruseks 800x600 pikslit ning keelake peaakna suuruse muutmise. (1 punkt).

### **Osa 2 – Programmi üldine väljanägemine**

Programmi akna 800x600 joonistusala taust peab olema valge. Programmi peamenüüs on tööriista valimiseks eraldi menüü „Tools“, kus on järgnevad elemendid:

```
Add vertex
Move vertex
Delete vertex
Add line
Delete line
```

Lisage programmi peaakna alumisse äärde olekuriba, millel on kolm osa. Esimene osa näitab parasjagu aktiivse tööriista nime. Teine ja kolmas osa näitavad vastavalt mitu tippu ja joont on joonisel („Vertices: <arv>“, „Lines: <arv>“). Vihje: olekuribale tekstialade lisamiseks saab kasutada QLabel klassi. See funktsionaalsus annab 1 punkti.

Vajalikud käsud leiate Qt dokumentatsioonist, kas Qt Creatori dokumentatsioonist või internetist:

- Signaalid ja pesad: <http://qt.nokia.com/doc/4.4/signalsandslots.html>
- QMainWindow: <http://qt.nokia.com/doc/4.4/qmainwindow.html>
- QWidget: <http://qt.nokia.com/doc/4.4/qwidget.html>
- QMenuBar: <http://qt.nokia.com/doc/4.4/qmenubar.html>
- QStatusBar: <http://qt.nokia.com/doc/4.4/qstatusbar.html>
- QMenu: <http://qt.nokia.com/doc/4.4/qmenu.html>
- QAction: <http://qt.nokia.com/doc/4.4/qaction.html>
- QLabel: <http://qt.nokia.com/doc/4.4/qlabel.html>

### **Osa 3: Tippude lisamine**

Kui kasutaja on menüüst valinud tipu lisamise, on programm tipu lisamise režiimis. Selle tähistamiseks ilmub olekuribale esimesele osale tekst „Adding vertices“. Kui programm on tipu lisamise režiimis, siis hiire vasakule nupule vajutamine mingis joonistamisala punktis tekitab sellesse punkti tipu. Tippu tähistatakse musta ringiga, mille läbimõõduks ligikaudu 20 pikslit. Tippu ei lisata, kui 10 piksli raadiuses juba on tipp. Lähedaste tippude otsimisel kasutage varem loodud vektori klassi meetodeid. Kui üks tipp on lisatud, siis programm jääb tipu lisamise režiimi.

Selleks tuleb joonistusosalal realiseerida hiireklikki töötlev meetod:

```
void QWidget::mousePressEvent(QMouseEvent *event);
```

Ning täiendada meetodit:

```
void QWidget::paintEvent(QPaintEvent *event);
```

Vihje: peate looma andmestruktuurid oleku (tipud, lõigud) salvestamiseks, hiirekliki peale seda uuendama ning paintEvent klassis andmete põhjal ekraanile joonistama.

Hiirenuppe loetleb Qt::MouseButton (QMouseEvent'i juurest leiate). Veel on abiks QPainter klass: <http://qt.nokia.com/doc/4.4/qpainter.html> Renderdamise kohta on natuke kirjas ka klassi QRect dokumentatsioonis: <http://doc.trolltech.com/4.4/qrect.html>

See funktsionaalsus annab lahendusele juurde 2 punkti.

### **Osa 4: Tippude liigutamine**

Kui menüüst on valitud tipu liigutamine, satub programm tipu lisamise režiimi. Vastavaks olekuriba tekstiks olgu „Moving vertices“. Kui kasutaja vajutab hiire vasaku nupu alla joonisel oleval tipul või sellest kuni 10 piksli kaugusele, siis muutub tipp liikuvaks ning kasutaja saab hiirt liigutades tippu liigutada. Kui tipuga on seotud mõni joon, siis peab see kaasa liikuma. Kui kasutaja laseb hiirenupu lahti, jääb tipp uude punkti paigale. Tippu ei saa liigutada joonistusosalalt välja. Pärast ühe tipu liigutamist saab kasutaja valida järgmise tipu.

Lahendus teenib 2 punkti siis, kui tippe saab liigutada ning jooned liiguvad tipuga kaasa.

Selleks tuleb joonistusosalal realiseerida lisaks meetodid:

```
void QWidget::mouseMoveEvent(QMouseEvent *event);  
void QWidget::mouseReleaseEvent(QMouseEvent *event);
```

### ***Osa 5: Tippude kustutamine***

Menüüst tippude kustutamise režiimi valik lubab joonistatud tippe kustutada. Kasutajat teavitab sellest olekuribal kiri „Deleting vertexes“. Vasaku hiirenupuga tipul (või 10 piksli raadiuses) vajutades kustutatakse tipp ja kõik jooned, mis sellega seotud on. Kui üks tipp on kustutatud, saab kasutaja kohe järgmise tipu kustutada. Funktsionaalsus annab 2 punkti.

### ***Osa 6: Sirglõikude lisamine***

Menüüst sirglõigu lisamise valik viib programmi sirglõikude lisamise režiimi. Seda vaid juhul, kui on lisatud kaks või rohkem tippu. Sisenemisel seatakse olekuriba tekstiks „Adding lines“. Kasutaja saab nüüd mõnel tipul vasaku hiirenupu alla vajutades valida esimese tipu (10 piksli raadiuse nõue kehtib ikka). Vajutades nüüd hiire vasakut klahvi mõne teise tipu lähedal (taas 10 piksli nõue) valib kasutaja valida teise tipu. Kui kasutaja on edukalt kaks erinevat tippu valinud, luuakse nende vahele sirglõik. Sirglõiku tähistatakse 1-piksli paksuse musta joonega. Programm on uuesti joone lisamise režiimis ning kasutaja saab valida esimest tippu. Programm, mis seda oskab, saab 2 punkti juurde.

### ***Lisaülesanne 1: Sirglõigu kustutamine***

Kui kasutaja valib Menüüst lõikude kustutamise, siis satub ta režiiminimega „Deleting lines“. Vasakut hiirenuppu alla vajutades kustutatakse

Kasutaja peab joone kustutamiseks vajutama vasaku nupuga joone peale või sellest piksli kaugusele. Pikslit läbivate sirglõikude leidmiseks kasutage näiteks joone võrrandeid. Selle kohta leiate materjali siit: <http://www.cut-the-knot.org/Curriculum/Calculus/DistanceToLine.shtml>

Kui ühte pikslit läbib mitu joont, siis kustutage esimene, mille leiate. Kasutaja saab kustutada mitu sirglõiku järjest.

See funktsionaalsus lisab punktisummale 1 lisapunkti.

### ***Lisaülesanne 2: Valitava tipu või sirge värvimine***

Tipud muutuvad ajutiselt punaseks kuni hiirekursor on nende kohal (10 piksli raadiuses) ning sirglõigud muutuvad siniseks, kui ollakse nende kohal (või ühe piksli läheduses). Kui hiirekursor nende kohalt lahkub, taastavad nad oma värvi. Lahendus annab veel 1 lisapunkti.

### **Vihjeid lahenduse jaoks**

- 1) Kui te ei kasuta Qt Creatorit, peab uute failide lisamisel nende nimed lisama ka vastavasse Qt projektfaili (faililaiend .pro).
- 2) Kasutage mingit andmestruktuuri joonisele lisatud tippude ja sirgete hoidmiseks.
- 3) Kirjutage endale funktsioon, mis ekraani koordinaatide järgi tagastab andmestruktuurist lähima tipu. Võite teha ka nii, et kui 10 piksli raadiuses tippu ei ole, siis ei tagastata midagi. Kasutage sellist funktsiooni igal pool, kus on vaja tippu valida. Sarnase funktsiooni võib kirjutada ka sirglõikude jaoks.
- 4) Kui mõni meetod tahab parameetriks QStringi, siis tr()-funktsiooni abil saab sõnesid vastavalt teisendada. Näiteks: `m_label->setText (tr ("Siia tuleb tekst"));`

### **Tähelepanu – küsimused ülesande kohta ja muud tingimused**

Võttes arvesse, et ülesanne nõuab tugevalt iseseisvat tööd, siis julgustan teid muredega puhul listi kirjutama.

Materjalid ja lahenduste üleslaadimise süsteem on aine veebilehel. Tähtaeg on praktikumi toimumisnädala pühapäeva õhtu kell 23:59 (aega on 7 päeva).