

Yleiskuvaus

Ohjelma on hiirellä toimiva piano. Toteutin projektin vaikeana. Kosketinta klikattaessa kuuluu ääni ja vastaava nuotti piirtyy nuottiviivastolle. Nuotin nimi näkyy koskettimiston vieressä painalluksen ajan. Piirtyviä nuotteja on neljä erilaista: neljäsosanuotti, puolinuotti, $\frac{3}{4}$ -nuotti ja kokonuotti. Pituus määrittyy painalluksen kestosta. Jos nuotti ei mahdu tahtiin, se piirtyy tahtiviivan molemmille puolille kaarella yhdistettynä. Viivastolle voi myös lisätä taukoja nappia painamalla. Alkuperäisen suunnitelman mukaista sävellajinmuutosta ei implementoitu lopulliseen projektiin.

Käyttöohje

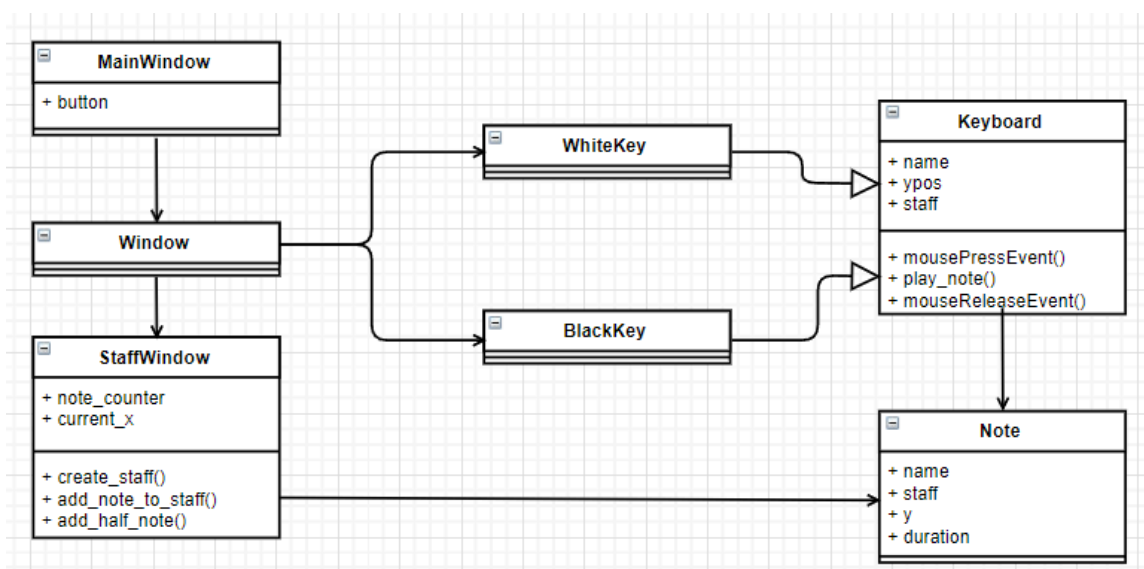
Ohjelma on käyttäjän kannalta varsin yksinkertainen. Kun käyttäjä ajaa ohjelman, ikkuna avautuu. Ikkunassa on yläpuolella nuottiviivasto ja alempana koskettimisto. Ikkunan alareunassa on "Add Rest" nappi.

Ulkoiset kirjastot

Ohjelmassa ei hyödynnetä ulkoisia kirjastoja.

Ohjelman rakenne

Alla on suuntaa antava UML-diagrammi ohjelman toiminnasta. Ohjelmassa on kuusi luokkaa. MainWindow yhdistää koskettimistoikkunan (Window) ja nuottiviivastoikkunan (StaffWindow). Window luokka luo koskettimiston WhiteKey ja BlackKey luokkien avulla. StaffWindow luo nuottiviivaston ja sen metodit hoitavat nuottien piirtymisen. WhiteKey ja BlackKey luokat perivät Keyboard luokan. Keyboard luokka sisältää esimerkiksi metodin mousePressEvent, joka tunnistaa käyttäjän painalluksen. Note luokka sisältää nuotin ominaisuuksia kuten nimen ja keston.



Algoritmit

Tämä projekti ei vaatinut monimutkaisia algoritmeja. Koodissa tarvittiin esimerkiksi hyvin yksinkertainen algoritmi, joka laskee y-koordinaatin jokaiselle nuotille. Y-koordinaatti saadaan lausekkeesta $ypos = 112 - i * 8$. Tässä luku 112 kertoo matalimman nuotin paikan, sillä nuottillistan ensimmäisen jäsenen indeksi on 0. Vierekkäisten nuottien ero on siis 8 pikseliä.

Tietorakenteet

Käytin ohjelmassa vain pythonin omia tietorakenteita kuten listoja ja sanakirjaa. Valitsin sanakirjan nuottien nimien ja äänitiedostojen yhdistämiseen. Jälkikäteen ajateltuna olisin voinut lisätä samaiseen sanakirjaan enemmänkin asioita, kuten nuotin y-koordinaatin.

Tiedostot

Ohjelma käyttää äänitiedostoja (.wav) äänten soittamiseen ja kuvia (.png) erilaisten nuottiviivaston elementtien kuvaamiseen.

Testaus

Suunnitelmassa olin listannut testaussuunnitelmaan seuraavat asiat:

- Jokainen kosketin soittaa eri äänen
- Nuotit piirtyvät oikeaan kohtaan
- Nuottien ajoitus on oikea
- tunnistaa käyttäjän painallus
- lisää tahtiviivat oikein
- printtaa oikean nuotin nimen

Kaikki näistä ehdoista toteutuvat ohjelmassa. Projektin ominaisuudet painottuvat graafisiin elementteihin, joten suuri osa testauksesta tapahtui visuaalisesti projektin edetessä. Testaustiedostossa on muutama yksikkötesti erinäisten ohjelman osien testaamista varten. Niistä kaksi testaa valkoisia ja mustia koskettimia ja yksi nuottien lisäämistä viivastolle.

Ohjelman tunnetut puutteet ja viat

Ohjelman pitäisi toimia niillä osa-alueilla, jotka projektin tehtävänannossa listattiin. Yksi pieni vika esiintyy heti ohjelman ajaessa, kun tyhjä nuottiviivastoikkuna avautuu hetkeksi ennen varsinaisen pääikkunan avautumista. Ongelma johtuu siitä, etten projektin aluksi suunnitellut yhdistää nuottiviivasto- ja koskettimistoikkunaa. Tämän takia huomasin vian vasta myöhemmin, enkä enää viitsinyt käyttää aikaa pienen ohjelman korjaamiseen.

Jos olisin voinut käyttää enemmän aikaa projektin tekemiseen, olisin lisännyt ainakin eri tahtilajin, tyhjennys ominaisuuden ja erilaisia äänitiedostoja. Myös jonkinlainen indikaattori nykyisen nuotin kestosta voisi olla hyödyllinen. Tällä hetkellä voi olla vaikea tietää tuleeko nuotista puolinuotti vai $\frac{3}{4}$ -nuotti. Jälkikäteen ajateltuna olisin myös mielelläni tehnyt pianon, jota voi soittaa tietokoneen näppäimistöllä.

3 parasta ja 3 heikointa kohtaa

Mielestäni projektini onnistunein ominaisuus on pidempien nuottien piirtyminen

oikein, vaikka ne eivät mahtuisikaan tahtiin kokonaan. Käytin paljon aikaa ominaisuuden suunnittelemiseen paperilla. Kävin läpi jokaisen mahdollisuuden ja luokittelin ne kategorioihin sen perusteella, kuinka kaukana ensimmäinen piirtyvä osa nuotista on tahtiviivasta.

Mielestäni nuottien äänet ovat yksi ohjelman heikoista kohdista. Ääni kuuluu vain hiiren painalluksen verran. Tämä tarkoittaa sitä, että hiiren painike tulee pitää pohjassa pidempään, vaikka käyttäjä haluaisi soittaa vain neljäsosanuotin. Olisin voinut tehdä soittamisesta sujuvampaa esimerkiksi määrittämällä nuotille minimipituuden, jolloin lyhyelläkin painalluksella kuuluisi järkevän pituinen ääni.

Ohjelmassa ei myöskään ole montaa yksikkötestiä. Koin haastavaksi keksiä testejä liittyen graafisten elementtien testaukseen. Lisäksi koodissa esiintyy jonkin verran ns. "maagisia lukuja". Suuri osa niistä liittyy jotenkin nuottien piirtymiseen.

Poikkeamat suunnitelmasta

Projektini noudattaa alkuperäistä suunnitelmaa melko hyvin. Suurin ero suunnitelmaan on sävellajiominaisuuden puute. Koin ominaisuuden melko turhaksi, enkä ollut varma miten olisin halunnut toteuttaa idean. Alkuperäisen suunnitelman mukaan tämä ominaisuus oli kategoriassa "jos jää aikaa", joten en pitänyt sitä kovin keskeisenä ominaisuutena.

Eripituisten nuottien implementointi vei enemmän aikaa kun odotin. En ollut suunnitelmassani huomionnut sitä, että pidempiä nuotteja tulee pystyä soittamaan myös silloin, kun tahdissa ei ole nuotille enää tilaa. Tämän takia lisäsin ominaisuuden, joka yhdistää nuotin osat kaarella tahtiviivan yli. Toinen uusi ominaisuus on tauon lisääminen napin avulla.

Toteutunut työjärjestys ja aikataulu

Olisin halunnut käyttää projektiin enemmän aikaa. Aluksi tein projektia säännöllisesti eteenpäin, mutta jälkeenpäin koodin kirjoittaminen keskittyi lähinnä välipalautusten ympärille. Ensin lähdin työstämään koskettimistoa, jonka jälkeen lisäsin nuottiviivastoikkunan. Nuottien piirtymiseen liittyi monta asiaa, kuten oikea sijainti viivastolla ja nuotin osien piirtäminen. Eri elementtien sijoittuminen oikein toisiinsa nähden vaati paljon testaamista.

Viimeiselle viikolle jätin kuvien hyödyntämisen ja äänten implementoimisen. En saanut tiedostoja toimimaan muuten, kuin laittamalla tiedoston koko polun koodiin. Äänen implementoinnin kanssa kohtasin pieniä ongelmia. Koodin kirjoittaminen ei ollut vaikeaa, mutta en ensin tiennyt olinko kirjoittanut sen oikein, sillä en kuullut ääniä omasta tietokoneestani. Yllätyksekseni kuitenkin huomasin äänten toimivan kun kopioin koodin pöytäkoneeni Pycharmiin. Useamman päivän jälkeen ongelman syyksi paljastui väärä python tulkki.

Arvio lopputuloksesta

Kokonaisuudessaan koen projektin onnistuneen melko hyvin. Sain selvitettyä

