OULUN AMMATTIKORKEAKOULU

Atte Kangasvieri

Ympäristökioski: SVG-työkalu

Ympäristökioski: SVG-työkalu

Atte Kangasvieri
Opinnäytetyö
Syksy 2021
Tietojenkäsittelyn tradenomi
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittelyn tradenomi

Tekijä(t): Atte Kangasvieri
Opinnäytetyön nimi: Ympäristöviisas viljelijä SVG-työkalu
Työn ohjaaja(t): Teppo Räisänen
Työn valmistumislukukausi ja -vuosi: Syksy 2021 Sivumäärä: 16 + 0

Opinnäytetyön toimeksiantajana oli Oamk:n opettaja, joka oli mukana Ympäristöviisas viljelijä hankkeessa. Opinnäytetyön aiheena oli suunnitella ja kehittää Ympäristöviisas viljelijä hankkeen Ympöristökioski sovellukselle valinta taulukkojen kuvien ylläpito työkalu.

Työssä käytettiin Python kieltä ja apuna käytettiin PyQt5 ja Python Firebase-moduuleita. PyQt5 on Python moduuli, joka auttaa käyttöliittymän kehityksessä ja Firebase-moduuli tuo toiminnot, joilla voidaan kommunikoida Firebase-tietokannan kanssa.

Opinnäytetyössä kehitetyn sovelluksen tarkoituksena oli pystyä luomaan SVG-kuva, johon voidaan lisätä valintaruutuja ja niiden kuvaus tekstejä. Tämän jälkeen valintaruudut liitetään taustakuvaan ja luodaan JSON-tiedosto, johon tallennetaan valinta ruutujen sijainnit.

ABSTRACT

Oulu University of Applied Sciences
Bachelor of Science in Computer Science

Author(s): Atte Kangasvieri
Title of thesis: Ympäristöviisas viljelijä SVG-tool
Supervisor(s): Teppo Räisänen
Term and year when the thesis was submitted: Autum 2021
Number of pages: 16 + 0

The thesis was commissioned by an Oamk teacher who was involved in the Ympäristöviisa viljelijä project. The topic of the thesis was to design and develop an Ympäristöviisa viljelijä project for an Ympäristökioski application for the selection of tables as an image maintenance tool.

The Python language was used in the work and the PyQt5 and Python Firebase modules were used. PyQt5 is a Python module that helps with UI development and the Firebase module brings functionality to communicate with the Firebase database.

The purpose of the application developed in the thesis was to be able to create a SVG image to which check boxes and their description texts can be added, after which the check boxes are attached to the background image and a json file is created in which the locations of the check boxes are stored.

SISÄLLYS

1 JOHDANTO

Ammattikorkeakoulujen rahoituksesta noin 24 % tulee TKI-työstä. Esimerkiksi Oulun Ammattikorkeakoulun kohdalla tämä tarkoittaa yli 10 miljoonaa Euroa joka vuosi. Ympäristöviisas viljelijä hanke on Oulun Ammattikorkeakoulun Tekniikka- ja Luonnonvara-alan yksikön koordinoima hanke. Hankkeessa kehitetään sähköinen työkalu, jonka avulla viljelijä voi suunnitella ja arvioida erilaisia tilan vesistö-, ilmasto- ja muita ympäristövaikutuksia pienentäviä valintoja sekä saa seurattavia ympäristötunnuslukuja. Työkalua kehitetään yhteistyössä viljelijöiden kanssa ja tavoitteena on, että työkalu jää maatilojen käyttöön hankkeen päättymisen jälkeen. (Korkeakouluille uusi rahoitusmalli 17.1.2019.)

Tässä opinnäytetyössä kehittämisaiheena on kyseisen SVG-työkalun kehittäminen. Työkalun tarkoitus on helpottaa Ympäristöviisas viljelijä hankkeessa kehitetyn verkkosivun sisältämien valintaruutujen ja tekstien (ns. LumoVesi-laatikot) ylläpitoa. Työkalun pitäisi osata piirtää laatikko ja lisätä, muokata ja poistaa valintaruutuja ja niiden sisältämää tekstiä. Lisäksi työkalun pitäisi luoda JSON-tiedosto, johon määriteltäisiin valintaruutujen sijainnit React-image-mapper-kirjastoa varten, ja tämän jälkeen tallentaa laatikko ja JSON-tiedostot Firebase-tietokantaan.

Opinnäytetyön luvussa 2 esitellään työn tietoperusta. Luvussa 3 tutustutaan paremmin hankkeeseen ja kehiteltävään työkaluun. Lopuksi luvussa 4 on työhön liittyvä pohdinta.

2 KÄYTETYT TYÖKALUT JA TEKNIIKAT

2.1 Python

Python on ohjelmointi kieli, jota käytetään verkkosivujen, sovellusten, sekä tehtävien automatisointiin ja data analysointiin. Python on yleisohjelmointi kieli, joka tarkoittaa, että sitä voidaan käyttää useiden eri ohjelmien kehittämiseen eikä se ole erikoistunut mihinkään tiettyyn aiheeseen. Tämä monipuolisuus ja aloittelija ystävällisyys ovat tehneet Python kielestä yhden käytetyimmistä ohjelmointi kielistä (What Is Python Used For? A Beginner's Guide 2021).

```
1   import sys
2   import configparser
3   from PyQt5.QtCore import QRect
4   from PyQt5.QtWidgets import QApplication, QFrame, QPushButton, QMainWindow, QAction
5   from scripts.Canvas import *
6   from scripts.Save import *
7
8 > class window(QMainWindow):···
96
97  def main():
98      config = configparser.ConfigParser()
99      config.read('config\settings.ini')
100
101     app = QApplication(sys.argv)
102     ex = window(config)
103     ex.show()
104     sys.exit(app.exec_())
105
106 if __name__ == '__main__':
107     main()
```

Kuva 1. Esimerkki Python koodista.

Yllä olevassa kuvassa 1 on esimerkki python koodista. Riveillä 1–6 käytetään import komentoa, jolla ladataan Python kirjasto toisesta moduulista. Rivillä 1 ladataan systeemitason komennot. Rivillä 2 ladataan parseri INI-tiedostoille. Riveillä 3 ja 4 ladataan PyQt5 Python kirjastosta luokkia GUI-elementeille. Riveillä 5 ja 6 ladataan opinnäytetyön projektia varten tehdyt skripti tiedostot. Rivillä 8 määritellään window-luokka, joka sisältää ohjelman GUI määrittelyn. Rivillä 97 määritellään main-funktio, jonka sisällä ladataan asetukset settings-tiedostosta. Riveillä 98–104 määritellään PyQt5 kirjaton GUI applikaation luonti. Rivit 106 ja 107 testaavat onko tiedosto sovelluksen main-moduuli, jos on niin moduulin main-funktio ajetaan.

2.2 PyQt5

PyQt5 on kirjasto, joka antaa käyttöön Qt GUI-rungon Python kielen kautta. Qt itse on kirjoitettu C++ kielellä. Käytettäessä Pythonin kautta kirjastoa applikaatioiden kehittäminen on nopeampaa ja eikä uhrata juurikaan C++ tarjoamaa nopeutta. (What is PyQt5? 2021.)

```
136         self.modalNPM = QFrame(self)
137         self.modalNPM.resize(int(self.settings['display']['width']),int(self.settings['display']['height']))
138 
139         self.overlay = QFrame(self.modalNPM)
140         self.overlay.resize(int(self.settings['display']['width']),int(self.settings['display']['height']))
141         self.overlay.move(0,20)
142         self.overlay.setStyleSheet('background-color:#36373d;')
143         self.opacity_effect = QGraphicsOpacityEffect()
144         self.opacity_effect.setOpacity(0.3)
145         self.overlay.setGraphicsEffect(self.opacity_effect)
```

Kuva 2. Esimerkki PyQt5 moduulista.

Kuvassa 2 riveillä 136 ja 137 luodaan modal-ikkuna ja määritetään sen koko. Rivillä 139 määritellään overlay-elementti, jonka jälkeen seuraavalla rivillä määritellään sille koko ja seuraavaksi sijainti. Rivillä 142 määritellään elementille taustaväri. Riveillä 143–145 lisätään elementtiin läpinäkyvyys efekti.

2.3 Firebase

Firebase on Googlen tarjoama palvelu, joka on keskittynyt mobiili ja web sovellusten kehitykseen. Firebase on pohjimmiltaan kokoelma työkaluja, joiden tarkoitus on tarjota sovelluskehittäjille työkaluja, jotka nopeuttavat ja helpottavat sovellusten kehitystä ja laajentamista. (What is Firebase? All secrets unlocked 2021.)

```
20  def fbPush(ref, config):
21      with open('out/'+ config['save']['name'] +'.json', 'r') as f:
22          file_contents = json.load(f)
23      key = ref.push(file_contents).key
24      oc.updateID(key)
```

Kuva 3. Esimerkki Firebase tallennuksesta.

Kuvassa 3 näytetään miten Firebaseen voidaan tallentaa JSON-tiedosto.

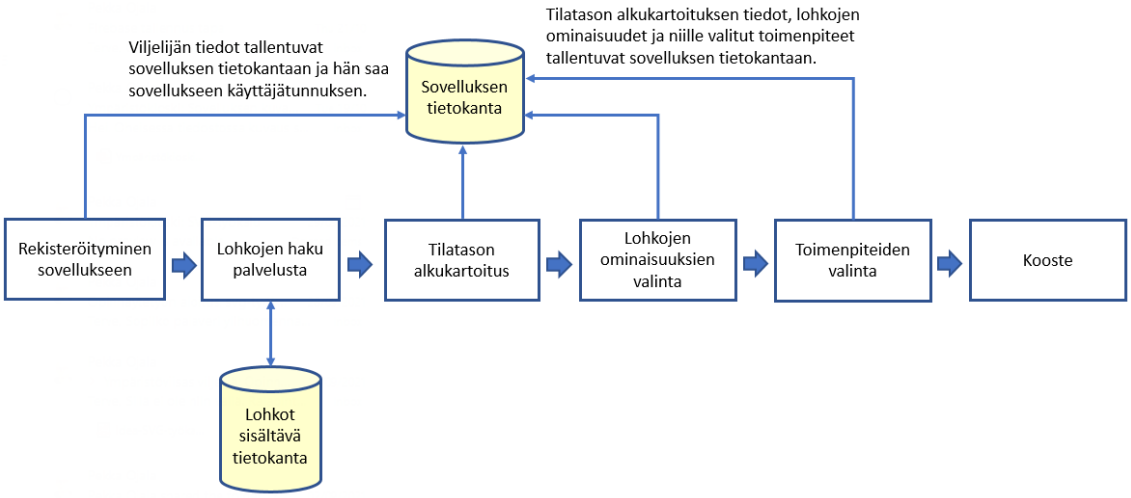
3 YMPÄRISTÖKIOSKIN TOTEUTUS

3.1 Ympäristöviisas viljelijä -hanke

Ympäristöviisas viljelijä hankkeella lisätään maatilojen välistä, erityisesti kestäviin tuotantotapoihin liittyvää yhteistyötä ja verkostoitumista sekä kehitetään maatiloille ympäristö- ja ilmastoviisaita toimintatapoja. Hanke hyödyntää uusinta tutkimustietoa sekä mallinnus- ja laskentamenetelmiä. (proagriaoulu 2021.)

Hanke koostuu seuraavista työpaketeista.

1. Pullonkaulat selville.
2. Tietoa vastuullisten ratkaisujen pohjaksi
3. Yhteistyöllä maataloutta kestävästi
4. Ympäristökioski vastuullisen viljelijän työkaluksi
5. Yhdessä tieto liikkeelle ja käyttöön



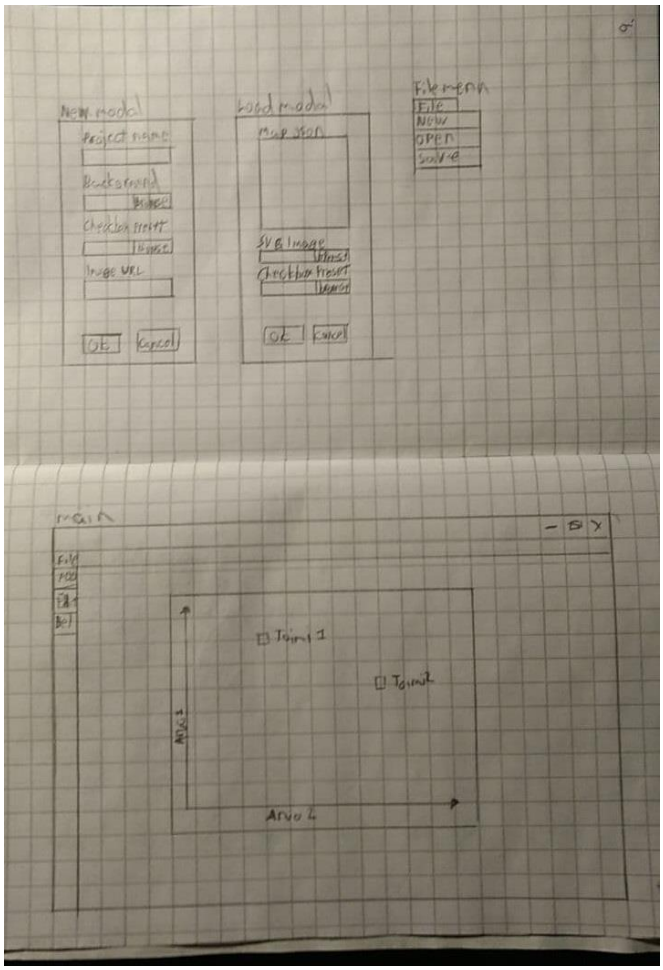
Kuva 4. Ympäristökioski toiminta kulku.

Kuvassa 4 on kuvattu Ympäristökioskin toimintaperiaate. Ensimmäiseksi sovellusta käyttävä viljelijä rekisteröityy sovellukseen, jonka jälkeen viljelijän tiedot tallentuvat tietokantaan ja hän saa sovelluksen käyttäjätunnuksen. Tämän jälkeen viljelijän omistamat lohkot (pellot ja metsät) haetaan tietokannasta. Seuraavaksi viljelijälle tehdään tilatason alkukartoitus (sijainnit ja kivennäismaan karkeus). Alkukartoituksen perusteella viljelijän lohkoista valitaan toimenpiteiden kohteet. Näille lohkoille valitaan seuraavaksi toimenpiteet. Tilaston alkukartoituksen tiedot, lohkojen ominaisuudet

ja niille valitut toimenpiteet tallentuvat sovelluksen tietokantaan. Lopuksi viljelijälle näytetään kooste valittujen toimenpiteiden mukaan. (Pekka Ojala 2021.)

3.2 Sovelluksen toteutus

Sovelluksen kehittämisen aloitin etsimällä tapoja GUI luomiseen käyttäen PyQt5 Python kirjastoa. Ensimmäiseksi tein käyttöliittymän käyttäen Qt Designer-sovellusta ja loin UI-tiedoston, mutta tulin päätökseen, että omaan tarpeeseeni sovelluksen käyttöliittymän kehittäminen oli helpompaa käsin koodaamalla kuin käyttäen ylimääräisiä työkaluja.



Kuva 5. UI suunnitelma

Käyttöliittymään lähdin koodaamaan kuvassa 5 olevien paperille suunniteltujen kuvien mukaan. Käyttöliittymän ensimmäisen version kehittämisen jälkeen aloin kehittämään SVG-taustakuvan la-

taamista tiedostosta ja renderöintiä canvakselle. Saatuani taustakuvan renderöinnin valmiiksi minulla oli hyvässä vaiheessa uusien valintaruutujen preset-kuvien lataaminen tiedostosta ja renderöinti napin painalluksella.

Seuraavana vaiheena oli taustakuvan ja lisättyjen valintaruutujen liittäminen yhdeksi SVG-kuvaksi ja JSON-tiedoston luominen, johon määritettäisiin valintaruutujen sijainnit React-image-mapper-kirjastoa varten. Saatuani sovelluksen luomaan SVG-kuvan ja JSON-tiedoston tein React-image-mapper testialustan pystyyn, jotta pystyisin testaamaan toimivatko sovelluksen luomat tiedostot yhdessä.

Kun olin saanut sovelluksen luomaan SVG-kuvan ja JSON-tiedoston aloitin tekemään valintaruutujen valitsemistoimintoa ja valintaruudun kuvaustekstin muokkaamista. Kuvaustekstien muokkaamiseen tarvitsi avata käyttäjälle tekstin kirjoituskenttä. Käyttäjän painaessa Enter-painiketta tekstit muokattiin niin, että se mahtui valintaruutujen kuvausteksti kentän leveyteen.

Seuraava vaihe oli liittää Firebase-sovellukseen, jotta JSON-tiedostot voitaisiin tallentaa tietokantaan automaattisesti.

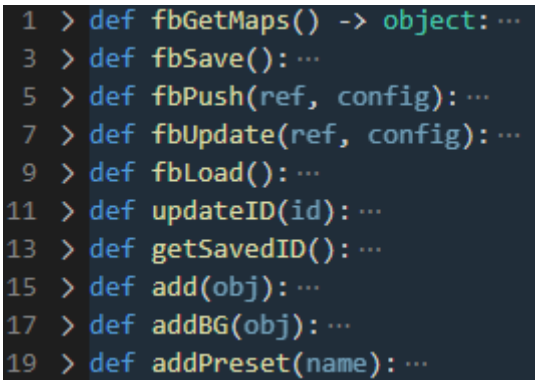
Saatuani Firebase-yhteyden toimimaan lisäsin sovellukseen modal-ikkunat uuden projektin luomiseen ja model-ikkunan vanhojen kuvien lataamiseen.

Seuraavaksi tein funktion, joka erittelee valmiista kuvasta valintaruudut niin että käyttäjä pääsee muokkaamaan niitä ladattaessa vanhaa projektia.

```
1  import configparser
2  from PyQt5.QtCore import pyqtSignal
3  from PyQt5.QtWidgets import QFrame, QGraphicsOpacityEffect, QLabel, QLineEdit, QListWidget, QPushButton, QMainWindow, QAction, QFileDialog
4  from scripts import (
5      canvas,
6      save,
7      objectController as oc,
8      Firebase as fb,
9  )
```

Kuva 6. Import esimerkki.

Kuvassa 6 rivillä 1 ladataan configparser-kirjasto, joka antaa käyttöön toiminnot INI-tiedostojen lukemiseen ja kirjoittamiseen. Rivillä 2 ladataan PyQt5.QtCore:sta signaalin käsittely moduuli. Rivillä 3 ladataan GUI widgettejä joiden avulla rakennetaan sovelluksen GUI.



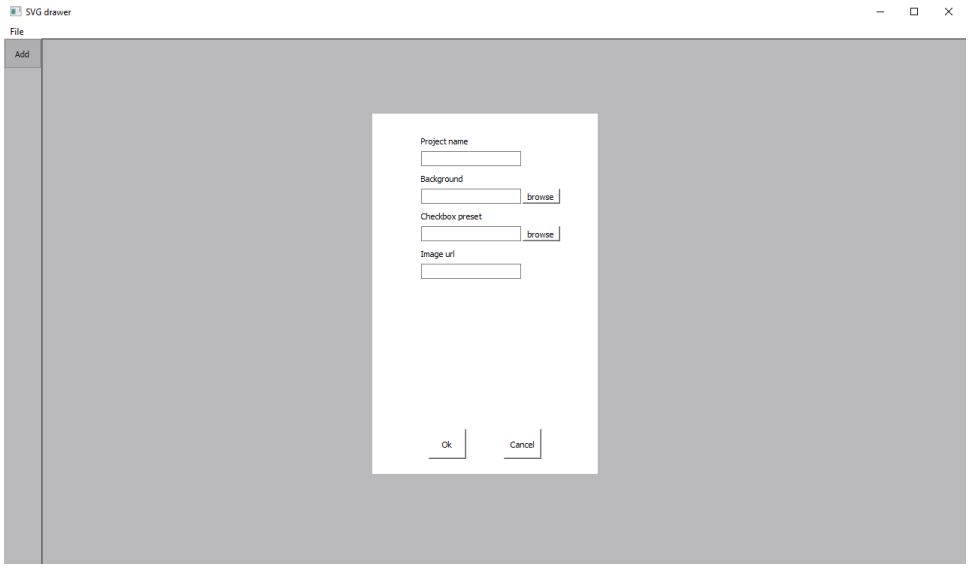
Kuva 7. Tärkeimpiä funktioita.

Kuvassa 7 on sovelluksen tärkeimpiä funktioita. Riveillä 1–9 olevat funktiot käsittelevät Firebase yhteyksiä. Rivit 11–19 ovat funktioita, jotka määritellevät sovelluksessa tehtävän projektin tietoja. Esimerkiksi funktio add(obj) lisää valintaruudun preset-arvon taulukkoon. Preset-arvoja käytetään sovelluksen toiminnassa myöhemmin.

```
23      for idx, img in enumerate(oc.elements):
24          imgFilesData[idx] = {'file': img.svg}
25          x = int(imgFilesData[idx]['file'].find('''.//*[@id='checkbox']''').get('x'))
26          y = int(imgFilesData[idx]['file'].find('''.//*[@id='checkbox']''').get('y'))
27          width= int(imgFilesData[idx]['file'].find('''.//*[@id='checkbox']''').get('width'))
28          height = int(imgFilesData[idx]['file'].find('''.//*[@id='checkbox']''').get('height'))
29          imgFilesData[idx]['file'].getroot()[0].set('id','checkbox_'+str(idx))
30          imgFilesData[idx]['root'] = transform.fromstring(ET.tostring(imgFilesData[idx]['file'].getroot(),'unicode')).getroot()
31          imgFilesData[idx]['root'].moveto(img.pos().x(), img.pos().y())
32          if not 'areas' in mapData['MAP']: mapData['MAP']['areas'] = []
33          mapData['MAP']['areas'].append({
34              'id': config['save']['idPrefix'] + str(idx) + config['save']['idSuffix'],
35              'shape': config['save']['shape'],
36              'coords': [
37                  img.pos().x()*float(config['save']['mapScaleFix'])+x,
38                  img.pos().y()*float(config['save']['mapScaleFix'])+y,
39                  (img.pos().x()+x+width)*float(config['save']['mapScaleFix']),
40                  (img.pos().y()+x+height)*float(config['save']['mapScaleFix']),
41              ],
42              'preFillColor': config['save']['preFillColor'],
43              'fillColor': config['save']['fillColor'],
44              'strokeColor': config['save']['strokeColor']
45          })
```

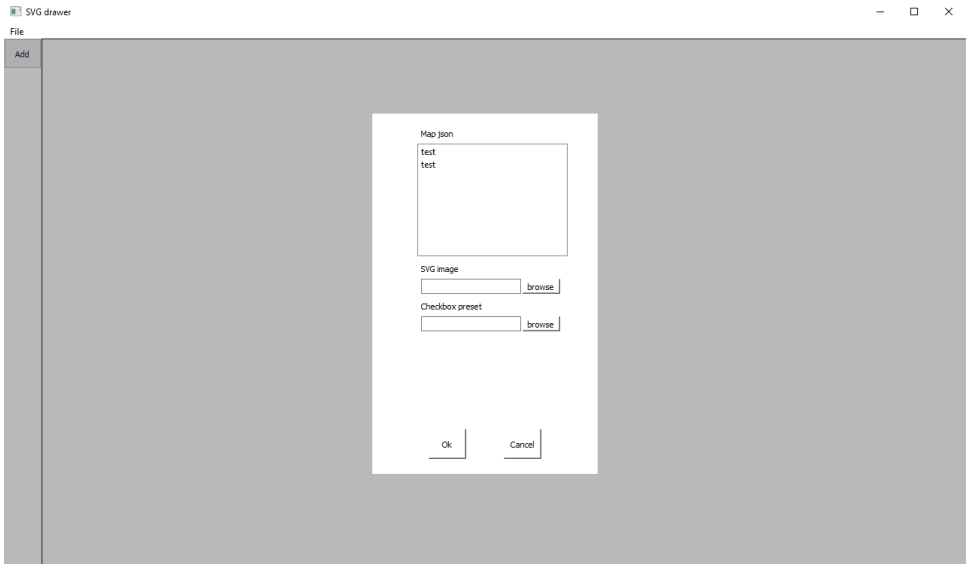
Kuva 8. Esimerkki valintaruutujen tallennus.

Kuvassa 8 rivin 23 for-silmukka käy läpi tallennetut valintaruudut. Riveillä 24–31 tallennetaan yksittäisen valintaruudun tiedot muuttujiin. Esimerkiksi tallennetaan valintaruudun x ja y koordinaatti sekä riveillä 27 ja 28 tallennetaan valintaruudun korkeus ja leveys. Rivillä 32 luodaan tyhjä karttalista. Tämän jälkeen riveillä 33–45 karttalistaan lisätään id, shape (esim. suorakulmio), coords (vasemman yläkulman koordinaatti ja oikean alakulman koordinaatti), preFillColor, fillColor (valinta ruudun taustavärit), strokeColor (valinta ruudun reunaviivan väri)



Kuva 9. Uuden projektin luonti modal.

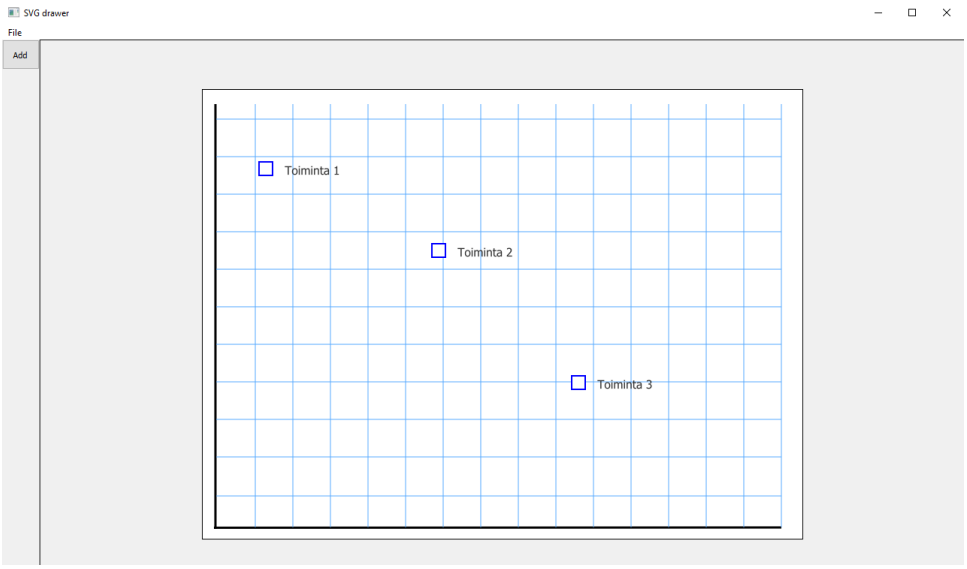
Kuvassa 9 on uuden projektin tietojen määrittely modal-ikkuna. Project name kohtaan käyttäjä määrittelee nimen projektille, joka tallennetaan JSON-karttaan. Background-valintakenttään valitaan SVG-kuva, jota käyttäjä haluaa käyttää valmiin kuvan taustakuvana. Checkbox preset-valintakenttään valitaan valintaruutujen SVG-kuva, joita lisätään taustakuvan päälle. Image url on valmiiden kuvien tallennussijainti, jonka avulla Ympäristökioski sovellus hakee SVG-työkalu sovelluksella tehdyt kuvat.



Kuva 10. Vanhan projektin lataus modal.

Kuvassa 10 on tehdyn projektin avaamis-modal-ikkuna. Map JSON-lista on Firebase-tietokantaan tallennettujen kartta tiedostojen lista. SVG image valintakenttään valitaan valmis kuva, jota käyttäjä

haluaa muokata. Checkbox preset valintakenttään valitaan valintaruutujen SVG-kuva, joita lisätään taustakuvan päälle.



Kuva 11. Esimerkki SVG projektista.

Kuvassa 11 on esimerkki projektista, johon on lisätty kolme valintaruudun preset-tiedostoa taulukko taustakuvan päälle ja määritetty valintaruutujen kuvaustekstit. Valintaruutuja voidaan liikuttaa hiirellä klikkaamalla ja vetämällä. Tupla klikkaus avaa kuvaustekstien muokkauskentän.

4 POHDINTA

Opinnäytetyössä suunnittelin ja kehitin SVG-ylläpito sovelluksen Ympäristöviisas viljelijä hankkeen Ympäristökioski sovellukselle. Ylläpitotyökalun tarkoitus oli luoda SVG-kuva, johon sovelluksen käyttäjä pystyy lisäämään valintaruutuja ja niiden kuvaustekstejä. Sovellus tallensi Firebase-tietokantaan JSON-tiedoston, jossa oli kuvan valintaruutujen koordinaatit.

Opinnäytetyössä sovelluksen ohjelmointi eteni hyvin. Sain tehtyä sovitussa ajassa toimivan sovelluksen. Kommunikointi työn toimeksiantajan kanssa toimivat hyvin. Opinnäytetyön etenemistä auttoi viikoittain pidetyt palaverit ohjaavan opettajan kanssa ja kävimme läpi, miten olin viikon aikana edennyt ja asetimme seuraavan viikon tavoitteet.

Tehtävien suunnitteluun olisin voinut käyttää enemmän aikaa. Tätä työtä tehdessä en käyttänyt paljon aikaa suunnitteluun ja etenin aika lailla vain viikko kerralla.

Työtä tehdessä opin Python kielen toimintatapoja ja PyQt5 moduulia käyttämällä opin tekemään visuaalisen käyttöliittymän. Työssä oli myös mukana Firebase jota olin jo aikaisemmin käyttänyt koulun projekteissa. Opinnäytetyötä tehdessä pääsin kokemaan miten Firebase toimii Python kielen kanssa.

Haasteita tuotti raportin kirjoittaminen, mutta työn myötä sain raportin myös etenemään.

5 LÄHDELUETTELO

Proagriaoulu 2020. Hakupäivä 3.10.2021. https://www.proagriaoulu.fi/fi/ymparistoviisas-viljelija/

Pekka Ojala 2021: Ympäristökioski: Sovelluksen kuvaus. Yksityinen sähköpostiviesti 19/10/2021. Viestin saaja: Atte Kangasvieri.

What is PyQt5? Hakupäivä 08.11.2021 https://build-system.fman.io/pyqt5-tutorial

What is Firebase? All secrets unlocked Hakupäivä 08.11.2021 https://blog.back4app.com/firebase/

What Is Python Used For? A Beginner's Guide Hakupäivä 08.11.2021 https://www.coursera.org/articles/what-is-python-used-for-a-beginners-guide-to-using-python

Korkeakouluille uusi rahoitusmalli 17.1.2019 https://okm.fi/-/korkeakouluille-uusi-rahoitusmalli