

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Sten Väljako

**TÖÖRIIST ANDMETE MASSKOGUMISEKS
GEOINFOGA RIKASTATUD
PILDIPANKADEST**

bakalaureusetöö

Juhendaja: Tanel Tammet
Doktorikraad

Tallinn 2017

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Sten Väljako

22.05.2017

Annotatsioon

Antud töö tulemusena on valmistatud tööriist, millega on võimalik kaevandada andmeid erinevatest geoinfoga rikastatud pildipankadest. Kaevandamiseks kasutatud andmebaasid on Flickr ja Pokemon Go map.

Lõputöö on kirjutatud Eesti keeles ning sisaldab teksti 13 leheküljel, 5 peatükki, 4 joonist.

Abstract

Collecting worldwide crowdsourced location data.

The thesis subject is a tool which can be used to mine location data from crowdsourced image databases (Pokemon Go map, Flickr, Instagram). Main objectives were to crawl public API (coordinate based) and to reverse engineer website in order to fetch data.

The thesis is in Estonian and contains 13 pages of text, 5 chapters, 4 figures.

Lühendite ja mõistete sõnastik

API	Rakendusliides
HTTP	Hüperteksti edastusprotokoll
POST	HTTP päringu tüüp
IP	Internetiprotokoll

Sisukord

1 Sissejuhatus	8
1.1 Eesmärk ja ülesandepüstitus	8
1.2 Suured sotsiaalmeedia-pildipangad	8
1.3 Sarnased uurimused ja lahendused	8
2 Pokemon Go map	11
2.1 Andmete kaevandamine	11
2.2 Kaevuri loomine	13
3 Flickr.....	15
3.1 Andmete kaevandamine	15
3.2 Kaevuri loomine	16
3.3 Andmete kvaliteet.....	16
4 Instagram	17
5 Tarkvara kasutamine.....	18
5.1 Pokemon Go kaevandamine rakendusena	18
5.2 Pokemon Go kaevandamine teegina	18
5.3 Flickr kaevandamine rakendusena.....	19
5.4 Flickr kaevandamine teegina	19
6 Kokkuvõte	21
Kasutatud kirjandus	22
7 Lisa	23
7.1 Pokemon Go map	23
7.2 Pokemon Go vastus	24
7.3 Flickr vastus.....	25

Jooniste loetelu

Joonis 1 - Flickr cities.....	9
Joonis 2 - Sightsmap.....	9
Joonis 3 - Pokemon Go map.....	10
Joonis 4 - MAPokemon	10

1 Sissejuhatus

1.1 Eesmärk ja ülesandepüstitus

Lõputöö eesmärgiks on luua praktiline töövahend asukohapõhise info andmete masspäringuks[1]. Masspäringu tegemiseks kasutatakse pildi-fookusega sotsiaalmeedia lehekülge Flickr ja mängu Pokemon Go. Töös uuritakse masspäringute tegemiseks võimalusi ning nende põhjal luuakse kasutatav tööriist, millega on võimalik andmeid kaevandada. Töö tulemusena loodud tööriistaga kogutud andmeid on vaja huviväärsuste asukohtade analüüsiks ja erinevate fotopõhiste sotsiaalmeedia-süsteemide võrdluseks antud kontekstis.

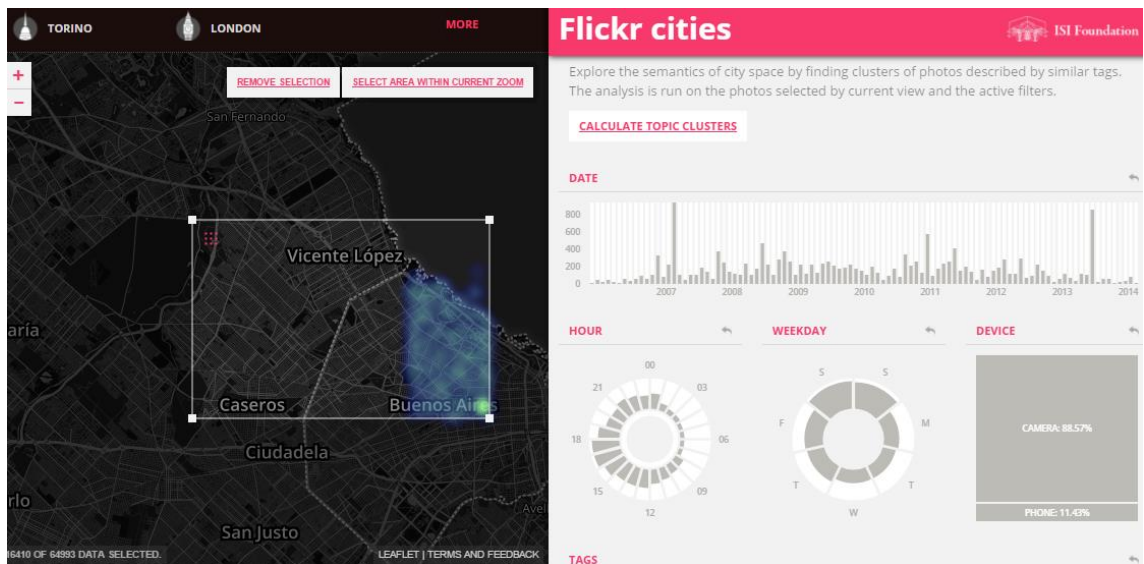
1.2 Suured sotsiaalmeedia-pildipangad

Töö tegemise hetkel suurimad sotsiaalpildi jagamise keskkonnad on Flickr ja Instagram.[2] Andmebaaside valiku juures sai otsustavaks nende suur populaarsus ja avalikult saadud andmete hulk.

Samuti Pildipangana saab käsitleda liitreaalsusmängu Pokemon Go-d: sealsed asukohad ja fotod on määratud ja laetud varasema sotsiaalse kaardipõhise mängu Ingress kasutajate poolt.[3]

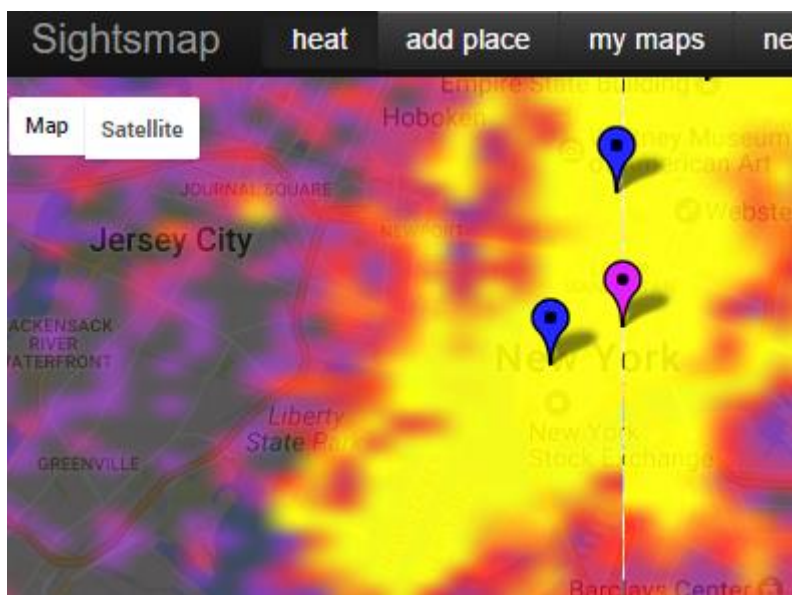
1.3 Sarnased uurimused ja lahendused

Flickr andmetes on eelnevalt kaevandatud suurlinnade kesklinnade heatmapiks[4]. Seal on esindatud järgnevad suurlinnad: Torino, London, Pariis, New York, Tokyo, San Francisco, Rooma, Veneetsia, Buenos Aires, Sidney ja Milano. Flickr cities lehel on võimalik valida linnast väike osa ja vaadata statistikat seal tehtud piltide kohta (mitu pilti on tehtud, mis kellaajal, jne.). Antud lehel töödeldakse andmeid ainult linna südamest.



Joonis 1 - Flickr cities

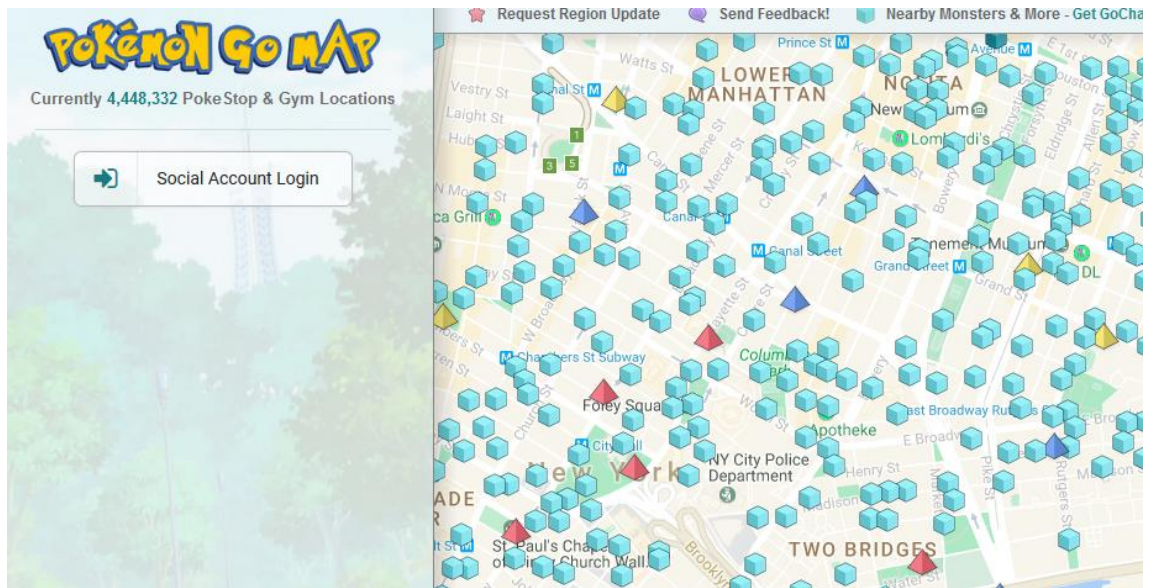
Asukohapõhiseid pildipanku on varem kaevandanud näiteks Sightsmap [5], seal on kasutatud Panoramio (praeguseks suletud), Wikipedia ja Foursquare koha-andmeid.



Joonis 2 - Sightsmap

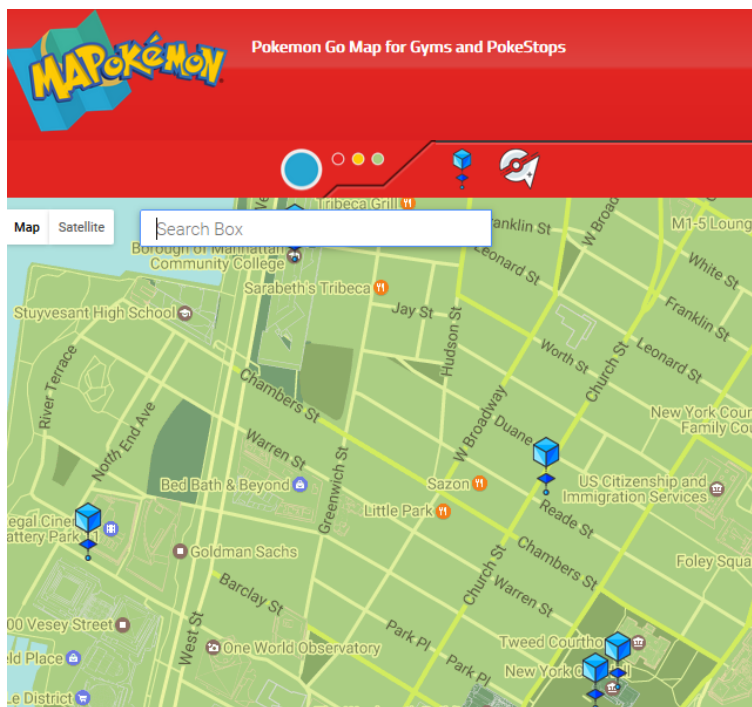
Pokemon Go avalikudest kaartidest eksisteerib:

- Pokemon Go map[6] – Kõige täiuslikum kolmanda osapoole kaart huvipunktidega.



Joonis 3 - Pokemon Go map

- MAPokemon[7] - Portaali kasutajate poolt sisestatud andmetega, andmed on väga puudulikud.



Joonis

4

-

MAPokemon

2 Pokemon Go map

Pokemon Go huvipunktide kaevandamiseks on valitud Pokemon Go map[6] kaart, kuna seal olev andmestik kõige ajakohasem ja detailsem. Lehekülg on mähitud Google mapi[8] ümber. Andmed tulevad nendepoolselt serverilt ja asukohad asetatakse kaardile.

2.1 Andmete kaevandamine

Pokemon Go map lehel puudub API. Leht teeb POST HTTP päringuid andmete saamiseks:

1. **fromlat:** 59.452919168291885
2. **tolat:** 59.46554570384721
3. **fromlng:** 24.927120136865256
4. **tolng:** 24.99419681990969
5. **fpoke:** 1
6. **fgym:** 1
7. **farm:** 0

Saadav vastus:

```
{
"214303":
  {
    "z3iafj": "MTEwMTQwODAxLjAwNA==",
    "f24sfvs": "NDYxNTU0NTMuMjUy",
    "g74jsdg": "MA==",
    "xg35": "MQ==",
    "y74hda": "MQ==",
    "zfgs62": "MjE0MzAz",
    "rgqaca": "laiakula-playground",
    "rfs21d": "Laiak Playground"},
"74121": {"z3iafj": "MTEwMTQ4MDY2LjY5Ng==", "f24sfvs": "NDYxNDUyNDUuMDI4", "g74jsdg": "MA==", "xg35": "MQ==", "y74hda": "MQ==", "zfgs62": "NzQxMjE=", "rgqaca": "muuga-elevant", "rfs21d": "Muuga Elevant"}}
```

Saadud vastuse andmed on Base64 formaadis välja arvatud asukoha nimi.

Huvipunkti andmete saamiseks on järgmine päring mis samuti HTTP POST päring:

mid:214303

Kus vastuseks on:

```
{
  "poke_id":"214303",
  "poke_clicks":"10",
  "fav_count":"0",
  "been_count":"0",
  "del_count":"0",
  "gym_count":"0",
  "poke_type":"1",
  "gym_team":"0",
  "comments":"0",
  "poke_image":"https://lh3.googleusercontent.com/cDajoVbcEZT5gfuBcafpODgAzL
hxrCWh6Y3giBQ3SSLEmKowjLxCAWyRh3tWHf4WKKqNRg-jNH1NcRFhgdbY",
  "confirm":1,
  "guard_id":"0",
  "gym_last":"Unknown",
  "gym_change":"Unknown",
  "pmon_status":0,
  "pmon_data":""}
```

Andmete töötlemisel tekitas probleeme asukoha koordinaatide hankimine. Antud vastuse koordinaadid on segaseks aetud koodiga Javascriptiga pärast laadimist töödeldud.

Viimsi vallas asuva Laiaküla mänguväljaku asukoha koordinaadid peale Base64 teisendust:

```
NDYxNTU0NTMuMjUy -> 46155453.252
MTEwMTQwODA1LjAwNA== -> 110140805.004
```

Tegelikud koordinaadid:

24.921984, 59.471278

Goldsteini mälestus märgi koordinaadid Long Islandil:

```
LTEzNjMyNzIwNy4xNTY= -> -136327207.156
NzU3NTAyMzMuNjA4 -> 75750233.608
```

Tegelikud koordinaadid(käsitsi kaardil otsides):

40.901927, -73.610748

Veebilehel asuv kood mis vastuse objekte töötleb[lisa 1].

Laiaküla mänguväljaku koordinaatide jagamisel (esialgne koordinaat/tegelik koordinaat) saame kordaja, mida arvutustes kasutades:

46155453.252 / 24.921984 ~= 1851997
110140805.004 / 59.471278 ~= 1851999

Kontrollime, kas kordaja kehtib ka Goldsteini mälestusmärgi koordinaatide puhul (arvutusvea põhjustab käsitsi kaardil asukoha määramine):

-136327207.156 / 185200 = -73.610803
75750233.608 / 185200 = 40.901854

Arvestades viga, mis käsitsi asukoha leidmisel, ei ole vaja koordinaatide leidmiseks läbi töödelda Javascripti koodi.

2.2 Kaevuri loomine

Leheküljel on päringute tegemine sessiooniga seotud. Esmaselt tuleb serveris registreerida sessioon ja siis saab alustada kaevandamisega. Spämmi vastu on teenuse pakkujal kaks kontrolli: sessiooni olemasolu ja päringu päises alljärgnev võti-väärtuspaar:

```
"X-Requested-With": "XMLHttpRequest"
```

Peale päise korrektse koostamist, toimub päringutele ilma tõrgeteta kiire vastamine. Leht piirab mass päringute tegemist IP põhise kontrolliga. Edukaks andmete kaevandamiseks on vaja rakendusel magada kuni piirang on aegunud. Spämmi vastane ooteaeg suureneb 30 sekundi võrra peale korduvat piirangu tabamist. Piirangu vastu võitlemiseks oodatakse 2,5 sekundit iga asukohtade otsimise päringu tegemisel. Kui rakendus saab vastu spämmi vastase vastuse ootab rakendus aja ära ja jätkab oma tegevust.

Koordinaadid sisestatakse ristkülikute kaupa. Koordinaatide sisestamist alustatakse kõige all vasakul asetsevast ristkülikust ja lõpetatakse kõige paremal üleval asetseva ristkülikuga. Koordinaatide sisestamist ristkülikus alustatakse alumisest vasakust nurgast ja lõpetatakse ülemise parema nurgaga.

Lõplik tegevuste järjekord:

1. HTTP päring <http://www.pokemongomap.info> lehele et saada PHPSESSID
2. Alustada ristkülikutega kaardi läbi käimist
3. Kogutud asukohad läbikäija lisa andmete päringuga
(<https://www.pokemongomap.info/includes/uy22ewsd1.php>)
4. Lisa päringust pildi URLi võtmine
(<https://www.pokemongomap.info/includes/locdata.php>)
5. Uuesti tegevust korrata alates kolmandast tegevusest uues ristkülikus kuni kõik ristkülikud on läbikäidud

Pildid asuvad Google serveris ja nendele ei ole vaja sessiooni edasi anda ehk saab hilisemalt pärida.

3 Flickr

Flickr leheküljel on olemas API mida kasutada. Tööks sobilikud on päring "flickr.photos.search" erinevate parameetritega:

bbox

Või

lat
lon
radius

Bbox on sarnaselt Pokemon Go mapiga ristküliku põhine otsimine. Koordinaatide algus punkt on ristküliku vasak alumine nurk ja lõpp on ristküliku parem ülemine nurk.

Koordinaadi ja raadiuse suuruse otsimisega peab tegema rohkem päringuid, et katta ära ringide vaheline tühimik. Parim on kasutada bbox parameetrit otsingute tegemiseks.

Spämmi vastu võitlemise piiranguid ei kohanud, strateegiat selle vastu pole vaja.

3.1 Andmete kaevandamine

Esmalt on vaja endale registreerida API võti. Flickr'i lehel on võimalus saada ka ajutine Explore API lehe võti, mis aegub iga 24 tunni tagant[8].

Päring:

```
https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=9299f677ec20bcb3c4f5565b53a04ab0&bbox=59.434177%2C24.734417%2C60.448761%2C25.843343&per_page=5&format=json&nojsoncallback=1
```

Vastus:

```
{ "photos": { "page": 1, "pages": 10, "perpage": 5, "total": 46,
  "photo": [
    { "id": "31944453866", "owner": "25071770@N05", "secret": "c504e8dfe0",
      "server": "557", "farm": 1, "title": "KNA_2068", "ispublic": 1, "isfriend":
      0, "isfamily": 0 },
    { "id": "29234840944", "owner": "92895471@N08", "secret": "5f74944247",
      "server": "8002", "farm": 9, "title": "#instagrammers #instatravel #igers
      #instagood #instagram #instapic #instamood #instalikes #picoftheday
      #bestoftheday #vscocam #photooftheday #photogrid #photoshoot #instaphoto
      #instadaily #iphone #iphonegram #iphoneonly #iphoneography #travelgram
      #travel", "ispublic": 1, "isfriend": 0, "isfamily": 0 },
    { "id": "26843020140", "owner": "97958986@N00", "secret": "fca89cb260",
      "server": "7118", "farm": 8, "title": "Handmade art at #Lalehjin #Hamedan
      #Iran #mustseeiran #tourism #hamgardi #discoveriran by #Note4", "ispublic":
      1, "isfriend": 0, "isfamily": 0 },
    { "id": "10344633046", "owner": "100336835@N02", "secret":
      "2ec5554596", "server": "7341", "farm": 8, "title": "Helsinki", "ispublic":
      1, "isfriend": 0, "isfamily": 0 },
    { "id": "6370699163", "owner": "16899467@N03", "secret": "c345ed61ee",
      "server": "6106", "farm": 7, "title": "dsc_5573.jpg", "ispublic": 1,
      "isfriend": 0, "isfamily": 0 }
  ] }, "stat": "ok" }
```

Et vastu tulnud andmete sees puudub pildi asukoht, tuleb otsida piisavalt väiksest kastist, mis tagastab soovitava tulemuse.

3.2 Kaevuri loomine

Päringute lihtsustamiseks otsustasin kasutada teeki, mis on tehtud Javas Flickr'i jaoks[9-10]. Võimalikult täpsete andmete saamiseks kasutan laiuskraadi ja pikkuskraadi samme 0.0001 ~ 11meetrit. Spämmi vastaseid võtteid et täheldanud.

3.3 Andmete kvaliteet

Meta andmeid tuleb päringu vastuses vähe tagasi. Et saavutada olukord, kus võimalik näha isikute trajektoore on vaja päringutesse tuua lisa parameetreid, nt: pildi loomise aeg või ülesse laadimise aeg. Neid andmeid vastuses tagasi ei tule, aga on võimalik luua päringuid, mis nende tingimuste järgi otsivad. Ehk iga päevaselt tuleb teha sama asukoha peale mitmeid päringuid. Kasu kogutud anmetest oleks inimestele, kes võõras kohas võimalikult väikse ajaga tahavad näha võimalikult palju, optimaalne teekond kureeritud vaatamisväärsuste vahel.

4 Instagram

Instagramil on samuti olemas API, mille läbi on võimalik suhelda. Võtme registreerimine testkeskkonda on kiire ja lihtne, aga valmistab probleeme. Nimelt on piiranguks 500 päringut minutis ühe võtme peale. Arvestades andmete täpsuse vajalikkust, ei ole mõistlik ühe arvutiga kaevandada. Rohkemate API võtmete genereerimise/OAUTH autentimise automatiseerimisega saaks päringuid automatiseerida, aga piltide kvaliteet ei ole sobilik analüüsiks. Keskmiselt 21st pildist viiel ei ole subjektiks inimene.

5 Tarkvara kasutamine

5.1 Pokemon Go kaevandamine rakendusena

Rakenduse käivitamine:

```
java -jar final-1.0-SNAPSHOT.jar -p <start longitude> <start latitude> <end longitude> <end latitude>  
<valikuline: otsitava kasti suurus -ss=0.1> <valikuline: määrab kas pildi url'i on vaja laadida -noPicture>
```

Näite päring:

```
java -jar final-1.0-SNAPSHOT.jar -p 24.664225 24.671654 59.391616 59.397297 -ss=0.01
```

Vastuse osa sellele päringule lisas 7.2.

5.2 Pokemon Go kaevandmine teegina

Sisend parameetrid:

```
startLon - otsitava ristküliku pikkuskraadi vasak alumine nurk  
startLat - otsitava ristküliku laiuskraadi vasak alumine nurk  
endLon - otsitava ristküliku pikkuskraadi parem ülemine nurk  
endLat - otsitava ristküliku laiuskraadi parem ülemine nurk
```

Lisa valikulised parameetrid:

getPicture – Vaatab kas teha igale pildile eraldi päring.

searchSize – Määrab kui suure sammuga muudetakse koordinaate.

Kaevandamine toimub läbi meetodi välja kutse:

```
fun scanArea(startLon: BigDecimal, startLat: BigDecimal, endLon: BigDecimal,  
endLat: BigDecimal,  
searchSize: String = "0.1", getPicture: Boolean = true):  
MutableList<PokeStop>
```

Pildi URLi saab hiljem kätte kasutades meetodit:

```
fun getPicture(pokeStop: PokeStop)
```

5.3 Flickr kaevandamine rakendusena

Rakenduse käivitamine:

```
java -jar final-1.0-SNAPSHOT.jar -f <start longitude> <start latitude> <end longitude> <end latitude> <apiKey> <apiSharedSecret> <valikuline: mis kuupäevast otsitakse -from=2017-05-26> <valikuline: mis kuupäevani otsitakse -from=2017-05-26> <valikuline: kas alla laadida kõik lehed -allPages> <valikuline: otsitava kasti suurus -ss=0.0001>
```

Näite päring:

```
java -jar final-1.0-SNAPSHOT.jar -f 24.744491 24.745598 59.437139 59.437584 e007e57a6407e12fd87508dc74fd19a0 secret
```

Vastuse osa sellele päringule lisas 7.3.

5.4 Flickr kaevandamine teegina

Sisend parameetrid:

```
startLon - otsitava ristküliku pikkuskraadi vasak alumine nurk  
startLat - otsitava ristküliku laiuskraadi vasak alumine nurk  
endLon - otsitava ristküliku pikkuskraadi parem ülemine nurk  
endLat - otsitava ristküliku laiuskraadi parem ülemine nurk  
apiKey - API võti  
sharedSecret - API võtme salasõna
```

Lisa valikulised parameetrid:

from – Mis kuupäevast alates pildid peavad tehtud olema.

to – Mis kuupäevani otsida.

allPages – Laeb alla kõik lehed mida leidis.

searchSize – Määrab kui suure sammuga muudetakse koordinaate.

Kaevandamine toimub läbi meetodi välja kutse:

```
fun scanArea(startLon: BigDecimal, startLat: BigDecimal, endLon: BigDecimal,
endLat: BigDecimal, apiKey: String = "",
    sharedSecret: String = "", from: Date? = null, to: Date? = null,
allPages: Boolean = false,
    searchSize: String = "0.0001"): MutableList<FlickrPageWrapper>
```

6 Kokkuvõte

Töö tulemusena loodi tööriist/teek[1], mis kaevandab asukoha põhiselt Flickr andmebaasist piltide, meta andmeid ja Pokemon Go populaarseima kolmanda osapoole kaardilt huvipunkte.

Edasi arendusena lõpetaks Instagrami kaevuri ja prooviks leida moodust kuidas filtreerida kasulike andmeid piltide seast mis avalikud. Samuti lisada Pokemon Go kaevurile lisa strateegiaid spämmi päringute tegemise vastu.

Kasutatud kirjandus

[1] Väljako, Sten [WWW] Valminud rakenduse lähtekood
<https://gitlab.com/kjeasr/pfscrapeminer> (25.05.2017)

[1] Nimekiri sotsiaal võrgustikest [WWW]
https://en.wikipedia.org/wiki/List_of_social_networking_websites (22.05.2017)

[2] MEYER, R. (2006). The Curious Mystery of the Map in Pokémon Go [WWW]
<https://www.theatlantic.com/technology/archive/2016/07/where-did-pokemon-go-get-its-map/490799/> (22.05.2017)

[3] Flickr Cities [WWW] <http://www.datainterfaces.org/projects/flickr/#> (22.05.2017)

[4] Sightsmap [WWW] www.sightsmap.com (22.05.2017)

[5] Pokemon Go map [WWW] <https://www.pokemongomap.info/> (22.05.2017)

[6] MAPokemon [WWW] <https://mapokemon.com/> (22.05.2017)

[7] [WWW] maps.google.com (22.05.2017)

[8] Flickr [WWW] <https://www.flickr.com/services/api/explore/flickr.photos.search>
(22.05.2017)

[9] Flickr4Java [WWW] <https://github.com/boncey/Flickr4Java> (22.05.2017)

[10] Maven Flickr4Java [WWW]
<https://mvnrepository.com/artifact/com.flickr4java/flickr4java/2.16> (22.05.2017)

[11] Gradle [WWW] <https://gradle.org/> (22.05.2017)

[12] Kotlin [WWW] <https://kotlinlang.org/> (22.05.2017)

7 Lisa

7.1 Pokemon Go map

```
eval((function(){
    var p='abcdefghijklmnopqrstuvwxyz';var
    s=[72,90,85,94,87,82,71,88,75,80,65,76,81,66,79,70,60,86,74,89];var
    e=[];for(var d=0;d<s.length;d++)e[s[d]]=d+1;var b=[];for(var
    k=0;k<arguments.length;k++){var z=arguments[k].split('~');for(var h=z.length-
    1;h>=0;h--){var u=null;var f=z[h];var g=null;var l=0;var x=f.length;var
    m;for(var q=0;q<x;q++){var j=f.charCodeAt(q);var w=e[j];if(w){u=(w-
    1)*94+f.charCodeAt(q+1)-32;m=q;q++;}else if(j==96){u=94*(s.length-
    32+f.charCodeAt(q+1))+f.charCodeAt(q+2)-
    32;m=q;q+=2;}else{continue;}if(g==null)g=[];if(m>1)g.push(f.substring(l,m));g
    .push(z[u+1]);l=q+1;}if(g!=null){if(l<x)g.push(f.substring(l));z[h]=g.join('
    ');}}b.push(z[0]);}var r=b.join('');var a=[92,126,39,96,10,42].concat(s);var
    v=String.fromCharCode(64);for(var
    d=0;d<a.length;d++)r=r.split(v+p.charAt(d)).join(String.fromCharCode(a[d]));r
    eturn r.split(v+'!').join(v);})(function(){var
    H*=["pokemongomap.info","",":","split","/","://","href","location",";","lentg
    h",".", "error","gym_team","g74jsdg","spadding","stringpad","confirm","xgxc35H
    +lat","z3iafjH+lng","f24sfvsH+type","y74hdaH+id","zfgs62","cleantitle","rgqac
    aH+title","rfs21d"];function _H-
    (H.26,H.6@q,H.59){if(!H.26){return};H)37=thisH(7]H\ 'H&5)][1H\ 'H&4)][0H\ '
    H&2)][0];H)04=H.26H(H&8)];for(H)48=0;H.48@w _H-H(9)];H.48++){H)15=_H-
    [H.48];if(H,== H.37){return};if(!H.6@q){continue};H,= H.37H(H&10)+
    H,);if(H,H(9)]== 2&& !H,[1]){return}};throw (H.59|| H*[11])}{_H-
    )(H*[0],1,H*[1]);H%12H$H#H!13H 16H$H#H!17H 18H$H#H!19H 20H$H#H!21H
    22H$H#H!23H 24H$H#H!25H 26H/H%27]];H%28H/H%29]];H%18H/Number(H%18]])/
    jqueryscrollzoom;H%20H/Number(H%20]])/
    jqueryscrollzoom})(~]];H%~]](H%~H$1~15]H\ '14~]]=
    $H(~itemH(~3]])(H*[~]H(~[H*[~var H._~_0x13C@v3~","poke_~H.15~0x13D04~_0x13D~]=
    '));
```

7.2 Pokemon Go vastus

```
[{
  "longitude": 59.396696,
  "latitude": 24.670201,
  "type": 0,
  "unknown1": 1,
  "unknown2": 1,
  "id": 1083498,
  "tag": "outdoor-chess",
  "name": "Outdoor Chess",
  "picture":
  "https://lh5.ggpht.com/rwX_HTWsk04LiA6qH0mgzLM4F8XJauxTSu9mOLLNNtEuG3w873yEzt
  EduhTXzw7n1BRjQNuH0gm5pia86rs"
}, {
  "longitude": 59.401338,
  "latitude": 24.671639,
  "type": 1,
  "unknown1": 2,
  "unknown2": 2,
  "id": 593747,
  "tag": "manni-raamatukogu",
  "name": "M\\u00e4nni Raamatukogu",
  "picture":
  https://lh3.googleusercontent.com/f9QhLlHSMYRaogY53CZR6hNNqVb-
  hrzjd8uL68p3kaRb915gWmYn_NXZWG4kIgL9nh1EViz-qI7L6aSvmZgT
}, ... {
  "longitude": 59.395151,
  "latitude": 24.665516,
  "type": 0,
  "unknown1": 1,
  "unknown2": 1,
  "id": 160642,
  "tag": "ttu-puhkeala-plaan",
  "name": "TT\\u00dc Puhkeala Plaan",
  "picture":
  "https://lh3.ggpht.com/51Yq2XUALxaiwMCenKu8RjuU2t8OncrYoqQmQkNNNSOM-
  tIYgyXNAIC3W81qL7CPc6rX_FM_koRZFLy1EmWv"
}, {
  "longitude": 59.399219,
  "latitude": 24.669758,
  "type": 0,
  "unknown1": 1,
  "unknown2": 1,
  "id": 22505,
  "tag": "libra-wall-art",
  "name": "Libra Wall Art",
  "picture": "https://lh3.ggpht.com/S8www-
  P5PhASKwBYcQkkLIUfkm0xnWlMjTmgsdfSqD0he16rQV5uZgqAzMxwu7uLm1kKTD9M7BYF1S5E7d8
  "
}, {
```



```

        "longitude": 59.399359,
        "latitude": 24.673741,
        "type": 0,
        "unknown1": 1,
        "unknown2": 1,
        "id": 1098898,
        "tag": "manni-outdoor-gym",
        "name": "M\u00e4nni Outdoor Gym",
        "picture":
"https://lh3.ggpht.com/wSHi185sK4At8lqJ0Y7IW7viHuSJHA59FU1AVBOYblam-
157x2f6u5OU1k_o2ziHuq2gyxtVtvJ_YnVqUPZw"
    }
]

```

7.3 Flickr vastus

```

[ {
    "startLongitude": "24.744491",
    "endLongitude": "24.744591",
    "startLatitude": "59.437139",
    "endLatitude": "59.437239",
    "result": [ {
        "id": "28544131340",
        "owner": {
            "id": "96599214@N07",
            "username": "",
            "ignored": false,
            "contact": false,
            "friend": false,
            "family": false,
            "admin": false,
            "pro": false,
            "iconFarm": 0,
            "iconServer": 0,
            "photosCount": 0,
            "bandwidthMax": 0,
            "bandwidthUsed": 0,
            "filesizeMax": 0,
            "revContact": false,
            "revFriend": false,
            "revFamily": false
        },
        "secret": "c1123be64e",
        "farm": "9",
        "server": "8825",
        "favorite": false,
    }
    ]
}

```

```

    "license": "",
    "primary": false,
    "title": "WP_20160807_13_42_36_Pro",
    "publicFlag": true,
    "friendFlag": false,
    "familyFlag": false,
    "comments": 0,
    "views": -1,
    "rotation": -1,
    "notes": [],
    "tags": [],
    "urls": [],
    "iconServer": "",
    "iconFarm": "",
    "url":
"https://flickr.com/photos/96599214@N07/28544131340",
    "originalFormat": "jpg",
    "originalSecret": "",
    "placeId": "",
    "media": "",
    "mediaStatus": "",
    "pathAlias": "",
    "originalWidth": 0,
    "originalHeight": 0,
    "hasPeople": false
  }, ... {
    "id": "15092079773",
    "owner": {
      "id": "67151500@N00",
      "username": "",
      "ignored": false,
      "contact": false,
      "friend": false,
      "family": false,
      "admin": false,
      "pro": false,
      "iconFarm": 0,
      "iconServer": 0,
      "photosCount": 0,
      "bandwidthMax": 0,
      "bandwidthUsed": 0,
      "filesizeMax": 0,
      "revContact": false,
      "revFriend": false,
      "revFamily": false
    },
    "secret": "a715824403",
    "farm": "4",
    "server": "3939",
    "favorite": false,
    "license": "",

```

```

    "primary": false,
    "title": "IMG_8169",
    "publicFlag": true,
    "friendFlag": false,
    "familyFlag": false,
    "comments": 0,
    "views": -1,
    "rotation": -1,
    "notes": [],
    "tags": [],
    "urls": [],
    "iconServer": "",
    "iconFarm": "",
    "url":
"https://flickr.com/photos/67151500@N00/15092079773",
    "originalFormat": "jpg",
    "originalSecret": "",
    "placeId": "",
    "media": "",
    "mediaStatus": "",
    "pathAlias": "",
    "originalWidth": 0,
    "originalHeight": 0,
    "hasPeople": false
}, {
    "id": "2540522715",
    "owner": {
        "id": "26577336@N05",
        "username": "",
        "ignored": false,
        "contact": false,
        "friend": false,
        "family": false,
        "admin": false,
        "pro": false,
        "iconFarm": 0,
        "iconServer": 0,
        "photosCount": 0,
        "bandwidthMax": 0,
        "bandwidthUsed": 0,
        "filesizeMax": 0,
        "revContact": false,
        "revFriend": false,
        "revFamily": false
    },
    "secret": "abf038bfde",
    "farm": "3",
    "server": "2053",
    "favorite": false,
    "license": "",
    "primary": false,

```

```
    "title": "EST070823_60043",
    "publicFlag": true,
    "friendFlag": false,
    "familyFlag": false,
    "comments": 0,
    "views": -1,
    "rotation": -1,
    "notes": [],
    "tags": [],
    "urls": [],
    "iconServer": "",
    "iconFarm": "",
    "url":
"https://flickr.com/photos/26577336@N05/2540522715",
    "originalFormat": "jpg",
    "originalSecret": "",
    "placeId": "",
    "media": "",
    "mediaStatus": "",
    "pathAlias": "",
    "originalWidth": 0,
    "originalHeight": 0,
    "hasPeople": false
  }
]
}
```