

Putkivirtaus

Hugo Tamm, 1020296, Energia- ja konetekniikka, 21.02.2023

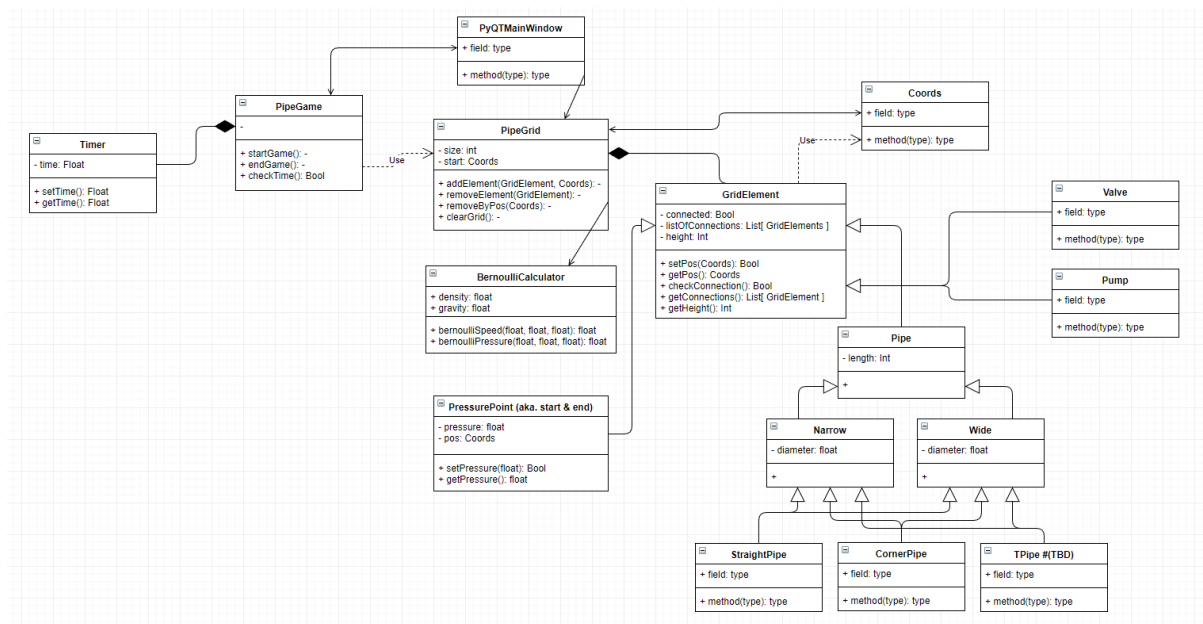
Tarkoituksena on luoda putkivirtausta simuloiva ohjelma, jossa voi asettaa alku- ja loppupään, mahdollisesti useamman pään, käyttäjän haluamaan ruudukkoon. Pääty yhdistetään suorilla putkilla, 90:n asteen mutkaputkilla ja mahdollisesti olisi myös muita variaatioita tarjolla, mm. T-putkia sekä pumppuja. Putket ovat ruudukoissa 1 metrin pituisia laskujen helpottamiseksi.

Tavoitteena on saavuttaa vaativa vaikeustaso ja ylittää se, esimerkiksi luomalla ohjelmaan myös pieni putkien yhdistely peli, jossa pelaajan täytyy yhdistää kaikki putket toisiinsa kääntämällä ne oikeinpäin tietyn aikarajan sisällä. Bernoullin lain avulla pystyy myös laskemaan putken sisällä kulkevan nesteen (tässä tilanteessa veden) nopeutta. Lisäämällä esim. Eri paksuisia putkia, voidaan lisätä vähän enemmän monipuolisuutta ja käytännön iloa ohjelmaan.

Ohjelma toteutetaan graafisella käyttöliittymällä, jolla olisi tarkoitus sekä kerätä käyttäjän antamia syötteitä sekä tulostaa tietoa samalle ikkunalle. Pääikkunassa näkyy iso ruudukko, jossa on alku- ja päätepiste, joille voi molemmille asettaa käyttäjän haluaman paineen (kPa). Ikkunassa on oma lokero putkille, joita voi asettaa ruudukkoon tai poistaa niitä sieltä klikkaamalla ruudukossa olevaa putkea. Ikkunassa on siis syöttökentät paineille, kuvalliset nappulat putkille, nappi nopeuden ja paineen laskemiselle sekä mahdollisia muita nappuloita. Kaikki nappulat antavat ohjelmalle komennon tehdä jotain.

Ohjelmaa käytetään siis ensisijaisesti putkiston sisällä virtaavan nesteen nopeuden laskemiseksi, kun tiedetään paineet. Ohjelmassa on myös pieni putkipeli, jolla voi haastaa itseensä ja/tai tuhlata aikaa. Pääikkunassa on nappi putkipelille, joka avaa uuden ikkunan, johon muodostuu pelikenttä täynnä putkia, jotka täytyy yhdistää toisiinsa kääntämällä niitä, niin ettei missään ole avoimia päitä.

Ohjelma on jaettu moneen osaan alla olevan UML-diagrammin mukaan.



Ohjelman ensisijainen tarkoitus on laskea pisteen A ja B välisessä putkistossa kulkevan veden nopeus. Tämä toteutetaan luomalla PyQt-kirjaston avulla käyttöliittymä, josta löytyy ruudukko eli *PipeGrid*. Pisteet A ja B, jotka luodaan luokalla *PressurePoint*, joiden paikkaa voi muuttaa sekä putkisto, jonka voi itse kasata eri *GridElementin* aliluokista. Pisteeseen A ja B voidaan asettaa haluttu paine yksikössä kPa (kilopascal), jonka jälkeen ohjelma laskee veden nopeuden hyväksytysti muodostetussa putkistossa *BernoulliCalculator*-luokan avulla.

GridElement-luokka on ylliluokka, josta *PressurePoint* sekä *Pipe* perivät monia metodeja, kuten sijainnin, korkeuden ja yhteydet muihin osiin. *Pipe*-luokka jakautuu vielä useampaan aliluokkaan, joista voidaan muodostaa kapeita, leveitä putkia. Näistä saadaan ainakin lähtökohtaisesti käyttöön suora sekä 90° asteen putki. Tämä ratkaisumalli pitäisi olla riittävä täyttämään keskivaikea taso, jonka päälle voi sitten rakentaa vaativan tason vaatimukset ja lisää.

Tiedon varastoiminen ja käsittely sujuu Pythonin omilla rakenteilla, joista parhaiten sopii lista. Ottaen huomioon, että putkistomme kulkee pisteestä A pisteeseen B, niin lista on täydellinen pitämään kirjaa siitä, missä järjestyksessä mikäkin osa putkesta on. Listasta voidaan helposti poimia putken ensimmäinen, viimeinen tai viides osa. Lista voidaan siis täyttää *GridElementeillä*, jotka pitävät itse sisällään myös niiden sijainnin *Coords*-luokkaa käyttäen.

Ohjelma käyttää eri *gridElementtien* kuvaajana ja mahdollisesti *PipeGridin* taustan jonkinlaista .png kuva tiedostoa, jotta ohjelma olisi helposti tulkittavissa ja mieluisan näköinen. Muunlaisille ohjelman ulkopuolisille tiedostoille ei ole varsinaista tarvetta, sillä kaikki syötteet voi käyttäjä antaa helposti käyttöliittymässä eikä ohjelma tallenna tuloksia mihinkään.

Laajennettu Bernoullin yhtälö on muotoa

$$\frac{p_1}{\rho g} + \frac{v_1^2}{2g} + z_1 + h_{loss} = \frac{p_2}{\rho g} + \frac{v_2^2}{2g} + z_2$$

Jonka voisit saattaa muotoon jossa muuttujat ovat kaikki yksikössä korkeus (m), en vielä ole selvittänyt miten.

Toinen tapa laskea nopeus paineiden avulla on

$$v = \sqrt{\frac{2(p_2 - p_1)}{\rho}}$$

Yksi sovellus tästä on pitot-putki, jolla voidaan mitata putken läpikulkevan aineen nopeutta mittaamalla paine eroa kahdessa eri pisteessä. Kaava ei välttämättä sovellu täysin tehtävään, mutta sitä voi kokeilla.

Testauksessa olisi hyvä testata:

- Virran nopeuden laskeminen
 - Nopeuden pitäisi muuttua ruudukossa olevan putkiston sekä käyttäjän antamien paineiden mukaisesti. Tätä voidaan testata laskemalla tietyn tilanteen tulos etukäteen ja tarkistamalla onko ohjelman antama tulos sama.
- Paineen laskeminen
- Ruudukossa olevan putkiston eheys
 - Jokaisen putken pään täytyy olla liitoksissa johonkin toiseen putkeen tai päätypisteeseen. Tälle voisi luoda jonkun oman funktion joka tarkistaa jokaisen putken päät, että ne ovat liitettynä.
- Nappien toiminta

- Putkien valikossa olevien nappien täytyy vaihtaa aktiivisena olevaa putken osaa, jotta ruudukkoon tulisi oikea putki painettuaan tyhjää ruutua.
- Laskuja suorittavien nappien tulisi käynnistä tarpeelliset funktiot *Bernoulli*-luokan sisällä ja tulostaa käyttöliittymään niiden tulos.
- Syötteet
 - Syötteiden formaatti ja oikeellisuus täytyy tarkistaa, ettei laskuja yritetä suorittaa merkkijonoilla tai muilla sopimattomilla arvoilla. Yksinkertainen assertRaises-komento pitäisi toimia tässä tilanteessa.
- Ruudukkoon putkien laittaminen ja poistaminen
 - Putkien täytyy ilmestyä tyhjään ruutuun kun sitä painaa ja poistua kun painaa putkea.
- Ruudukon tyhjentäminen
 - Ruudukon tulisi tyhjentyä, kun painaa clear-nappia joka tyhjentää ruudukon.
- Päätepisteiden siirtäminen
 - Pitäisi pystyä siirtämään vetämällä. Tätä vetämistä ei välttämättä pysty testaamaan yksinkertaisesti, mikäli se on edes mahdollista lisätä tällainen ominaisuus. Mutta päätepisteiden sijainnit pystytään tarkistamaan aina siirtojen jälkeen ja näin testaamaan ovatko ne oikeissa paikoissa.

Math-kirjasto saattaa koitua tarpeelliseksi laskujen muodostamisessa, mutten ohjelman toteutus onnistuu pelkällä PyQt:llä.

Ohjelman tekeminen aloitetaan siitä, että opitaan PyQt:llä tekemään sovelluksen pääikkuna sisältäen putket ja tarvittavat napit sekä ruudukko. Putkivirtauksen laskuja täytyy vähän vielä hienontaa ja varmistaa, että ne ovat oikein. Heti kun on ohjaustapaamiset hoidettu 3.3.2023 mennessä, niin aloitan ohjelmoinnin.

Ensimmäinen tavoite on saada käyttöliittymä toimimaan niin, että ruudukko ja putket ovat näkyvillä ja niitä voisi käyttää. Suunnitelmassa esitetysä UML-diagrammissa on luokkia, joita luon tarpeen mukaan. Ensin pääikkuna, sitten

ruudukko, sitten *GridElement*-luokka joka jakautuu putkiin ja päätepisteisiin. Seuraavaksi pitää varmistaa toimintakyky näille luokille, eli jonkin sortin testausta luodaan. Sitten matemaattinen osuus seuraa kun putket ovat ruudukossa ja niitä voi siirtää, lisätä ja poistaa.

Tiedonhaussa on käytössä Google, Youtube, vanhat Aallon omat virtausmekaniikan kurssimateriaalit. Esimerkiksi pari sivustoa:

<https://phys.libretexts.org/>

<https://courses.lumenlearning.com/>

<https://mycourses.aalto.fi/course/view.php?id=28999§ion=3>

Michel van Biezen - youtube kanava

The Efficient Engineer – youtube kanava