

# DreamSpace

## 1. Ohjelman rakennesuunnittelu

Sisustussuunnittelu ohjelman käyttöliittymä koostuu valikosta, mistä voit ladata pohjapiirroksen kuvan sekä huonekaluista, joita voi tämän kyseisen pohjapiirroksen päälle siirtää.

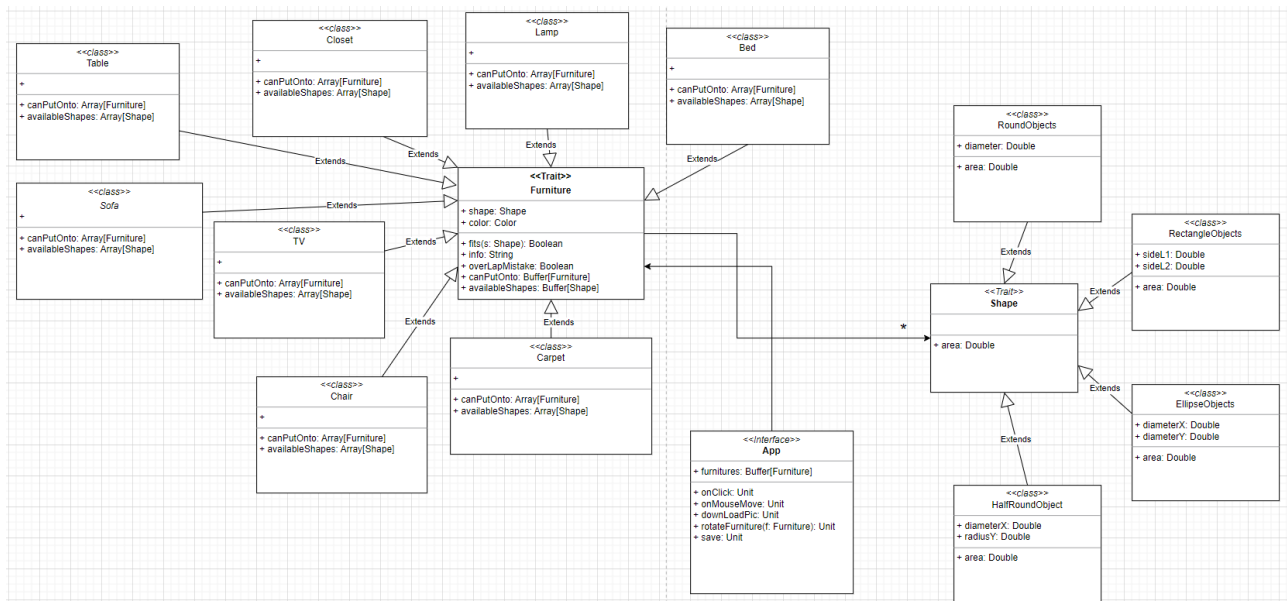
Käyttöliittymän luokkaan "App" on siis koottu lista huonekaluista, joita voit kuvaan valita sekä metodeista, joilla ohjelman käyttö luonnistuu. Luokalle annetaan siis tieto huonekaluista bufferin ("funrnitures") muodossa, johon on lueteltu kaikki käytössä olevat huonekalut, joita käyttäjä voi kuvan päälle asettaa. "App" - luokan sisältämiä metodeita ovat "onClick", "onMouseMove", "downloadPic", "rotateFurniture" ja "save". Metodit "onClick" ja "onMouseMove" määrittävät tapahtumat, mitä ohjelmassa tapahtuu, kun tietokoneen hiirellä painaa tietystä kohtaa ja miten hiiren liike vaikuttaa näytöllä oleviin tapahtumiin. Metodissa "onClick" on esimerkiksi määritelty tilanne, kun käyttäjä painaa hiiren vasenta näppäintä huonekalun kohdalla oikeanpuolella olevassa valikossa. Tällöin käyttäjä on valinnut huonekalun, jonka hän haluaa asettaa pohjapiirroksen päälle. Näytölle ponnahtaa tällöin pieni ikkuna, jossa kysytään halutut mitat kyseiselle huonekalulle sekä väri, jonka käyttäjä voi valita vapaasti. Valittuaan mitat sekä värin, hiirtä liikuttamalla voi käyttäjä siirtää huonekalun haluamalleen paikalle. Tämä tapahtuma on määritelty metodissa "onMouseMove". Metodi "downloadPic" nimensä mukaisesti lataa pohjapiirroksen kuvan tiedostoista ohjelmaan. "save"-metodi tallentaa valmiin suunnitteluprojektin laitteen tiedostoihin, joista työtä voi myöhemmin halutessaan tutkia. Metodi "rotateFurniture" vaihtaa huonekalun asentoa 45 astetta myötäpäivään.

Ohjelmaan kirjatut huonekalut omaavat jokainen oman luokkansa. Luokkien nimet ovat "Carpet", "Chair", "TV", "Sofa", "Table", "Closet", "Lamp" ja "Bed" ja ne kuvaavat nimensä mukaisesti kutakin huonekalutyyppiä. Jokaiselle huonekalulle on kirjattu huonekalut, joiden päälle valittu huonekalu on mahdollista asettaa. Nämä huonekalut on kirjattu Array-muuttujaan ja ne voidaan tarkistaa metodista "canPutOnTo". Tämä metodi nimittäin palauttaa Arrayn, johon on kirjattu kaikki huonekalut, joihin päällekkäin asettelu on mahdollista. Maton päälle voi esimerkiksi asettaa sohvan sekä pöydän. Lisäksi huonekaluilla on metodi "availableShapes". Tämä metodi palauttaa kaikki mahdolliset muodot, joissa kyseinen huonekalu voi esiintyä. Osa huonekaluista voi nimittäin olla eri muotoisia. Esimerkiksi pöytä voi olla pyöreä, neliskanttinen tai jopa puoliympyrän muotoinen.

Jokainen huonekalu on yksilöllinen tapaus, joka on johdettu piirreluokasta "Furniture". Tähän luokkaan on kirjattu käyttäjän ohjelman kautta kirjaamat mitotukset halutulle huonekalulle sekä väri. Nämä tiedot ovat muuttujissa "shape" ja "color". Metodeita piirreluokalla on "fits", "info" sekä "overlapMistake". Metodi "fits" antaa boolean-arvon, jonka perusteella se kertoo, mahtuuko huonekalu huoneeseen käyttäjän antamalla mitoilla. Metodi "info" antaa nimensä mukaisesti tietoa huonekalusta tekstin muodossa.

Piirreluokka "Furniture" käyttää toisen piirreluokan "Shape" tietoja. Luokkaan "Shape" on määritelty luokkia, jotka kertovat eri huonekalujen muotojen vaihtoehtoista. Vaihtoehtoisia muotoja ovat pyöreät muodot, kuten "EllipseObject", "RoundObject" ja "HalfRoundObject". Lisäksi

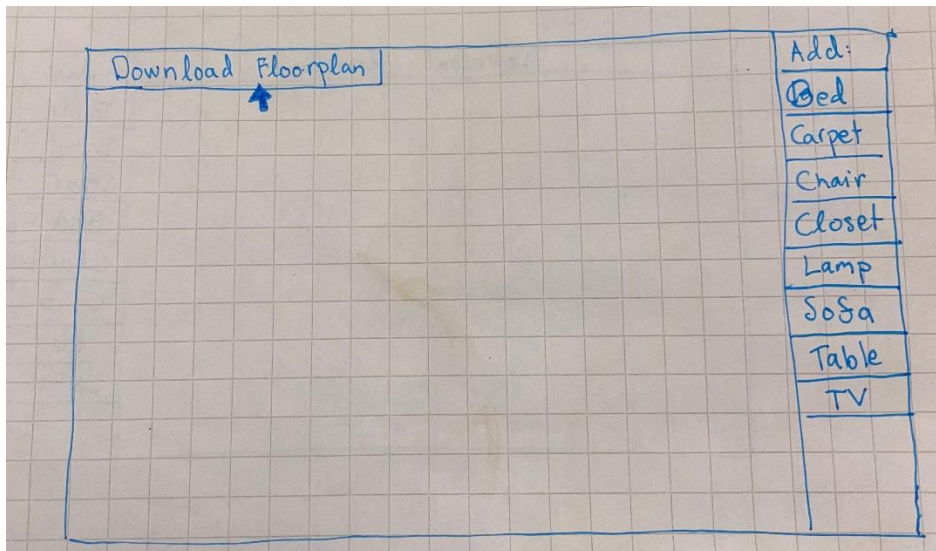
on määritelty luokka suorakulmaisille objekteille: "RectangleObject". Jokaiselle omalle muodolle on määritelty mitat. Pyöreät muodot ottavat parametreikseen halkaisijat (diameterX, diameterY) ja suorakulmaiset objektit ottavat parametreikseen sivun pituudet (sideL1, sideL2). Jokaiselle muotoluokalle on myös määritetty metodi "area". Tämä metodi kertoo kappaleen pinta-alan. Kun huonekalu siirretään pohjapiirroksen päälle, piirrelukassa "Furniture" tämä pinta-ala otetaan huomioon ja katsotaan, sopiiko kyseiset mitat huoneeseen. Tämä tapahtuma käsitellään kyseisen luokan metodissa "fits". "Shape" -piirrelukasta voi metodilla "area" nähdä myös kappaleen pinta-alan.



Picture 1: UML-kaavio

## 2. Käyttötapauskuvaus

Kun käyttäjä avaa ohjelman, esille tulee ruudun vasemmalle valikko, missä huonekaluja on lueteltu nimeltä. Ruudun yläaidassa on teksti "Download floor plan". Klikkaamalla tätä tekstiä käyttäjä voi ladata asunnon pohjapiirustuksen tiedostoistaan. Tämä tapahtuma käynnistää luokan "App" metodeista "onClick" ja "downloadPic".



Picture 2: Käyttöliittymä ennen pohjapiirroksen lisäämistä.

Kun kuva on ladattu, voi käyttäjä valita valikosta haluamansa huonekalun, antaa sille muodon, mitat sekä siirtää sen pohjapiirroksen päälle haluttuun sijaintiin. Mitat syötettyä, ohjelma käyttää "Shape" -piirreluokan metodia "area", ja ohjelma laskee huonekalun pinta-alan. "Furniture" -piirreluokan metodi "fits" tarkistaa, että huonekalu sopii huoneeseen annetuilla mitoilla. Kun käyttäjä on valinnut huonekalun metodi "onMouseMove" reagoi siihen niin, että huonekalua voi siirtää hiiren liikkeen avulla. Kun huonekalu on siirretty sen haluttuun paikkaan voi klikkaamalla hiirtä sen asettaa paikoilleen. Kun huonekalu on paikoillaan voit klikata sitä, jolloin tulee pieni valikko, missä lukee "rotate" ja "move". Jos käyttäjä valitsee ruudun "rotate" huonekalu kääntyy 45 astetta myötäpäivään. Tällöin ohjelma kutsuu luokan "App" metodia "rotateFurniture". Jos taas klikkaa ruutua "move" voi huonekalua uudelleen siirtää, kuten aikaisemmin. Kun huonekalua ei voi siirtää tietyn huonekalun päälle, huomataan tämä virhe "Furniture" -piirreluokan metodissa "overlapMistake" ja ohjelma palauttaa virheilmoituksen tapahtumasta. Käyttäjän tallentaessa sisustussuunnitelman, ohjelma käynnistää luokan "App" metodin "save". Tällöin suunnitelma voidaan tallentaa koneelle haluttuun tiedostoon.

### 3. Algoritmit

Ohjelman käyttämät algoritmit perustuvat pinta-alojen mittaamiseen. Suorakulmaiset objektit ottavat mittaussparametreikseen sivujen pituudet ja kertomalla nämä yhteen saadaan pinta-ala. Pyöreille objekteille annetaan mitoiksi leveydet halkaisijoiden muodossa. Tämä valinta on tehty selkeyden ja helpomman hahmottamisen vuoksi. Metodi, joka laskee pyöreiden kappaleiden pinta-alan, niin että se puolittaa halkaisijan pituuden (eli ottaa säteen) ja laskee ympyrän tai ellipsin pinta-alan sille kuuluvalla geometrisella kaavalla. Poikkeus on tietysti puoliympyrät, jotka ottavat halkaisijan lisäksi esineen säteen.

Pinta-alojen algoritmit kirjoitettuna:

Suorakulmaiset objektit:

$$A = sideL1 \cdot sideL2$$

Ympyränmuotoiset objektit:

$$A = \pi \cdot \left( \frac{diameter}{2} \right)^2$$

Ellipsin muotoiset objektit:

$$A = \pi \cdot \left( \frac{diameterX}{2} \right) \cdot \left( \frac{diameterY}{2} \right)$$

Puoliympyrän muotoiset objektit:

$$A = \frac{\pi \cdot \left( \frac{diameterX}{2} \right) \cdot radiusY}{2}$$

Kun ohjelma tarkistaa mahtuuko jokin tietty esine huoneeseen se tarkistaa, onko annetut mitat sopivia. Riippumatta valitusta huonekalun muodosta, ohjelma ottaa ylös annetut y- ja x-suuntaiset mitat. Näistä mitoista luodaan suorakulmio, jonka avulla skaalataan huonekalua kuvaan ja voidaan tarkistaa sopiiko huonekalu tilaan. Kyseisen tapahtuman algoritmin käsitellään "Furniture" luokan metodissa "fits" ja se on muotoa:

*sideL1 < roomSide1 && sideL2 < roomSide2*

#### 4. Tietorakenteet

Ohjelmassa tarvittavan tiedon varastointiin käytetään Array-kokoelmia. Kokoelmat, mitä ohjelmassa käytetään eivät muuta kokoaan ohjelman käytön aikana. Tämän vuoksi Arrayt ovat selkeämpikäyttöisiä kuin Bufferit tässä ohjelmassa. Arrayt ovat kuitenkin joustavampia käytössä kuin vektorit, joten sen vuoksi ohjelmaan on valittu Arrayt vektoreiden sijaan.

#### 5. Tiedostot

Ohjelma käsittelee erilaisia kuvatiedostoja. Nämä kuvatiedostot voivat olla jossain graafisessa muodossa kuten png tai jpg tiedostona. Ohjelma lukee tiedoston ja asettaa sen näytölle niin, että sen päälle voi siirtää huonekaluja. Kun työ on valmis, tiedoston voi tallentaa uutena tiedostona. Tämä tiedosto on binääritiedosto. Tiedot kaikissa ladatuissa tiedostoissa ja ladattavissa tiedostoissa on esitetty graafisesti kuvan muodossa.

#### 6. Aikataulu

Karkea arvio työtunneista ja toteutettavista asioista viikkojen aikana:

12.2.-26.2. Muodostetaan projektille luokat ja metodit alustavasti valmiiksi omiin tiedostoihin. Metodeita ei tarvitse tässä kohtaa vielä toteuttaa. Aika-arvio on +4h. Projektia tehdään niin paljon eteenpäin kuin näinä viikkoina ehtii. Syynä pieneen työmäärään tässä kohtaa on tenttiviikko.

26.2.-11.3. Työarvio on +12h. Muodostetaan metodeita ja ainakin käyttöliittymä-luokan "Shape" ja "Furniture" toteutus lähes valmis.

11.3.-25.3. Työarvio +12h. Valmiina ovat ainakin kaikki huonekalujen luokat ja käyttöliittymä on muodostettu.

25.3.-8.4. Työarvio +12h. Tavoitteena olisi, että ohjelman koodi olisi tähän mennessä siinä vaiheessa, että tehtävänä olisi vain debugata.

8.4.-22.4. Työarvio 16h. Debuggausta ja viimeistelyä. Jos ollaan myöhässä aikataulussa tässä kohtaa on pakko kiriä. Jos työ on hyvällä mallilla voidaan suunnitella lisäyksiä projektin vaikeustasolta "vaikea".

22.4.-25.4. Työarvio +4h. Työ palautusvalmiina.

## 7. Testaussuunnitelma

Testatessa ohjelman koodia voidaan käyttää esim erilaisia Stubb- tai Dummy-olioita ja REPL:ä voidaan joidenkin metodien toimintaa testatessa käyttää. "Shape"-piirreluokassa esiintyvät algoritmit voi testata REPL:ssä, sillä matemaattiset laskutoimitukset on helppo tarkistaa joillain itse valituilla arvoilla.

Kun testataan piirreluokkaa "Furniture" voidaan luoda Stubb- tai Dummy-olioita huonekaluista tai luoda ensin kokonainen valmis huonekaluluokka, jota testauksissa käytetään. Samaa käytäntöä voidaan käyttää "App" -luokan testauksessa. Graafisen käyttöliittymän testaus voidaan aloittaa kunnolla, vaikka esimerkiksi silloin, kun huonekaluluokkien toteutukset ovat valmiita. Tällöin valmiita luokkia voi hyödyntää testauksessa.