

# Projektisuunnitelma Kasino

Niko Somila  
710578  
Bioinformaatioteknologia  
2019, Toisen vuoden opiskelija  
25.2.2021

## Yleiskuvaus ja vaikeustaso

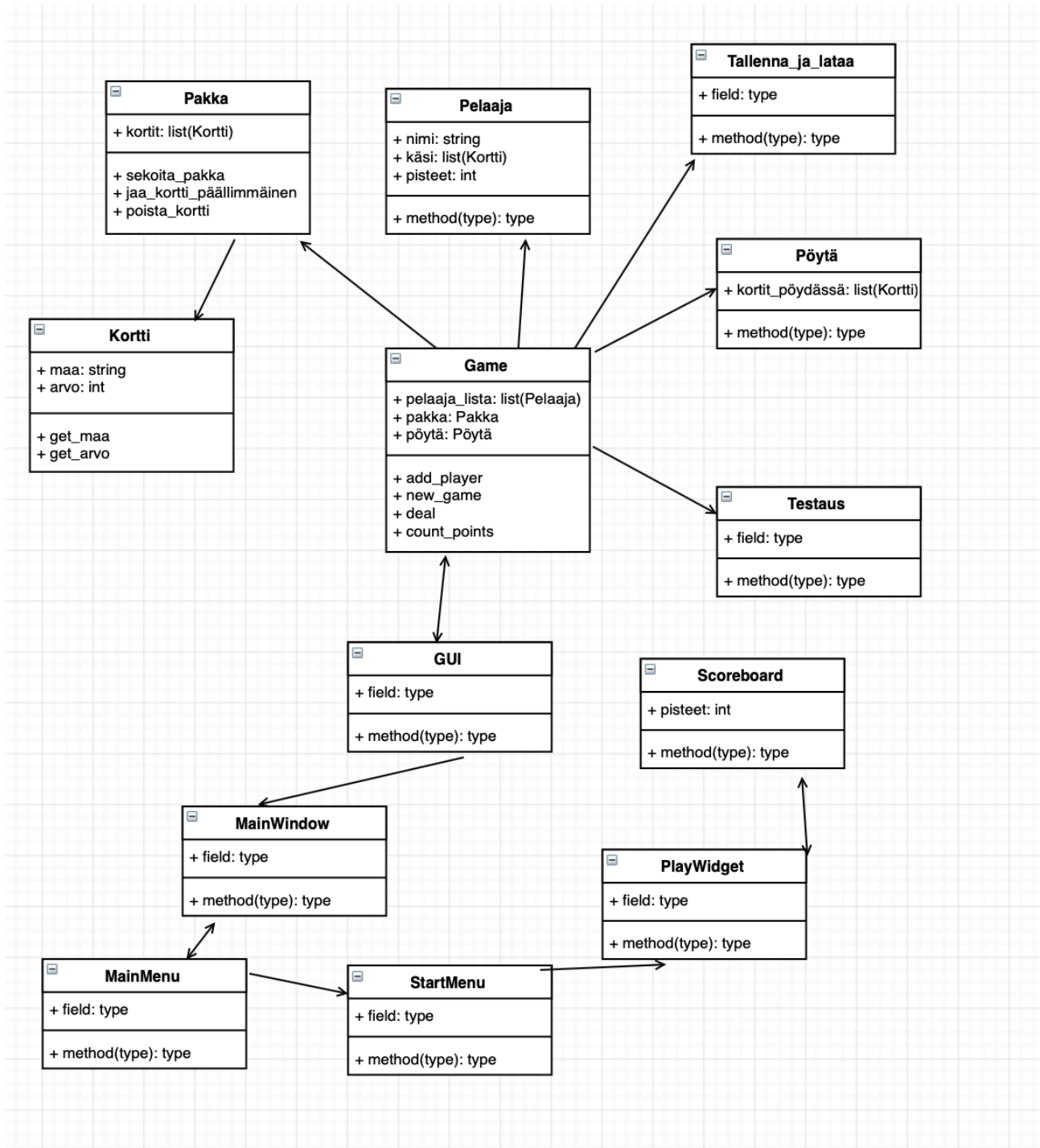
Tehtävänä on toteuttaa kasinokorttipeli ns. "pakkakasino" graafisena versiona. Itse ajattelen toteuttaa työn keskivaikean version. Ohjelma luo mahdollisuuden valita 0-N verran pelaajia ja pelissä on mahdollista tallentaa ja ladata pelitilanteita.

## Käyttötapauskuvaus ja käyttöliittymän luonnos

Ohjelmaa lähdetään ajamaan main.py tiedostossa. Tähän kohtaan avataan MainWindow pääikkuna. Pääikkunaan luon kolme erilaista näppäintä (aloita peli, lataa peli ja lopeta). Aloita peli painikkeen painettua edetään pelivalikkoon, jossa lisätään pelaajia peliin, tyhjennetään vanhat pelaajat ja aloitetaan peli. Kun peli aloitetaan, siirrytään pelitilanteeseen eli tässä tapauksessa pelipöytään. Napilla aloita peli aloitetaan peli. Silloin kaikille pelaajille jaetaan 4 korttia ja pöytään jaetaan myös 4 korttia. Peli etenee niin, että kun pelaajalla on pelivuoro hän valitsee kortin/kortit jotka haluaa pelata. Korttien pelaamisen jälkeen on mahdollista joko jatkaa seuraavaan pelaajaan tai lopettaa peli. Näkymän alareunassa on myös tallenna näppäin, joka tallentaa pelin.

Avaat ohjelman ja pääikkuna avautuu. Pääikkunassa on vaihtoehdot aloita peli, lataa peli tai lopeta. Käyttäjä valitsee aloita peli, jonka jälkeen tulee ikkuna missä käyttäjä pystyy lisäämään peliin pelaajia kirjoittamalla pelaajan nimen kenttään. Kun pelaajat ovat lisätty aloitetaan peli, eli pakka sekoitetaan ja jokaiselle lisätylle pelaajalle jaetaan kortit ja pöydälle jaetaan kortit. Näin pelin alustus on suoritettu. Peliä pelaataan niin pitkään kunnes kaikki kortit ovat pelattu tai peli lopetaan tai tallennetaan.

## Ohjelman rakennesuunnitelma



## Tietorakenteet

Käytän projektissa enimmäkseen tietorakenteina listoja. Listojen käsittely on helppoa ja siihen liittyy paljon hyödyllisiä metodeja (esim. len, append, pop, remove. Niillä pidän huolen esimerkiksi pelaajan kädestä (lista Kortti-olioista), pöydän korteista (lista Kortti-olioista), pakasta (lista Kortti-olioista), pelaaja lista (lista Pelaaja-olioista). Käytän listaa, koska osaan hallita sitä hyvin ja sillä pystyy pitämään selkeästi halutun tiedon hallinnassa. Toinen vaihtoehto olisi käyttää sanakirjaa, mutta minusta siitä ei ole hyötyä tässä tehtävässä.

## Tiedostot ja tiedostoformaatit

Käytän tiedostoja projektissa pelin tilanteen tallentamiseen ja lataamiseen. Tiedostot ovat tekstitiedostoja. Tallennan tiedot tekstitiedostoon niin, että ensin tulee #-merkki ja sitten kerrotaan pelaajien määrä. Sen jälkeen aletaan käymään pelaajia läpi. Pelaajalla on tiedostossa tallennettuna nimi, pelaajan käsi eli esitetään mitä kortteja pelaajalla on, mitä kortteja pelaaja on kerännyt, pelaajan pisteet ja lopuksi pelaajan mökit. Kun kaikki pelaajat on käyty läpi, kerätään pöydässä olevat kortit. Lopuksi katsotaan vielä pakassa olevat kortit, koska sitten saadaan ladattua samanlainen pelitilanne, jos peli keskeytetään ja tallennetaan jossain vaiheessa. Alla esimerkki millaiselta tallennettu tiedosto voisi näyttää.

#Pelaajien määrä  
pelaajat: 2

#Pelaaja1  
käsi: p2, r7, h8, h10  
kerätyt: r8, u9, p7  
pisteet: 8  
mökit: 1

#Pelaaja2  
käsi: p1, p4, r11, u12  
kerätyt: p9, r12, h1  
pisteet: 5  
mökit: 0

#Pöytä  
Kortit: r2, p12, h9

#Pakka  
Kortit: p12, h7, r4, r5...

p=pata, r=risti, h= hertta, u=ruutu  
1=ässä, 11=jätkä, 12=kuningatar jne...

## Algoritmit

Pelissä on mukana kombinaatioiden etsintä algoritmi. Tällä tarkoitetaan sitä, että kun tietokone pelaa vuorollaan kortin niin peli tarkistaa mahdolliset kombinaatiot, joita tietokone pystyy vuorollaan saamaan ja valitsee niistä parhaan. Tämä siis tietokoneelle. Eli siis algoritmi käy läpi tietokoneella olevat kortit ja vertailee sitten pöydällä oleviin kortteihin ja näin valitsee parhaan kortin, mitä pelata. Tässä myös pitää katsoa, että tietokone ei pelaa sellaista korttia, jonka seurauksesta seuraavaa pelaaja saa helpon mökin.

Tarkistus algoritmi. Eli kun pelaaja pelaa kortin kädestä, peli tarkistaa pelatun kortin arvon ja katsoo mihin pöydän kortteihin pelattu kortti liittyy, eli tuleeko mökki ja kuinka monta korttia

pelaaja kerää itselleen. Jos kortilla ei saa pelattua kortteja pois pöydältä, niin kortti lisätään automaattisesti pöydälle. Tässä erityisesti erikoiskortit pitää ottaa huomioon.

## **Testaussuunnitelma**

Pelissä ajattelin testata ainakin seuraavia asioita: onko pakka luotu oikein, onko pelaajat lisätty oikein, onko pelaajien määrä oikea, onko pisteet laskettu oikein, testaan mökkien määrää, onko annettu kortti saman arvoinen kuin pöydässä oleva kortti (varsinkin kun kyseessä on erikoiskortti), ja myös se, että pöydän korttien summa on sama kuin annettu kortti. Testaan myös pelin sitä, että peli on tallennettu ja se pystytään lataamaan oikein.

## **Kirjastot ja muut työkalut**

Graafiseenkäyttöliittymän luomiseen käytän PyQt5- kirjastoa. Pakan sekoittamiseen käytän random-kirjastoa. Testeissä käytän unittest-kirjastoa.

## **Aikataulu**

Pyrin saamaan pelin perustoiminnon ilman käyttöliittymää maaliskuussa ja arvioisin aikaa kuluvan n. 40 tuntia. Aloitan Kortti, Pakka ja Game olioista. Yritän saada yhden olion viikossa eteenpäin ja ensisijainen tavoite on saada peli toimimaan. Kun saan pelin toimimaan lisään tallennus ja lataustoiminnot peliin. Tähän arvioin aikaa menevän noin 5 tuntia. Loppuajan käytän käyttöliittymään ja grafiikka suunnitteluun ja tähän arvioin aikaa menevän noin 50 tuntia. Teen tämän kuitenkin lopuksi. Jos saan käyttöliittymän tehtyä tarpeeksi nopeasti, niin siirryn tavoittelemaan vaikean tason tietokonepelaajien luomista, mutta jätän sen viimeiseksi.

## **Kirjallisuusviitteet ja linkit**

Katson apua ohjelmointiin monesti YouTubesta ja siellä on jotai korttipelihin liittyviä videoita. Myös sivustoja Stackoverflow ja Geegsforggeegs näyttää paljon ohjelmointia materiaalia.

## **Liitteet**