

Y2 Projekti - Kasino

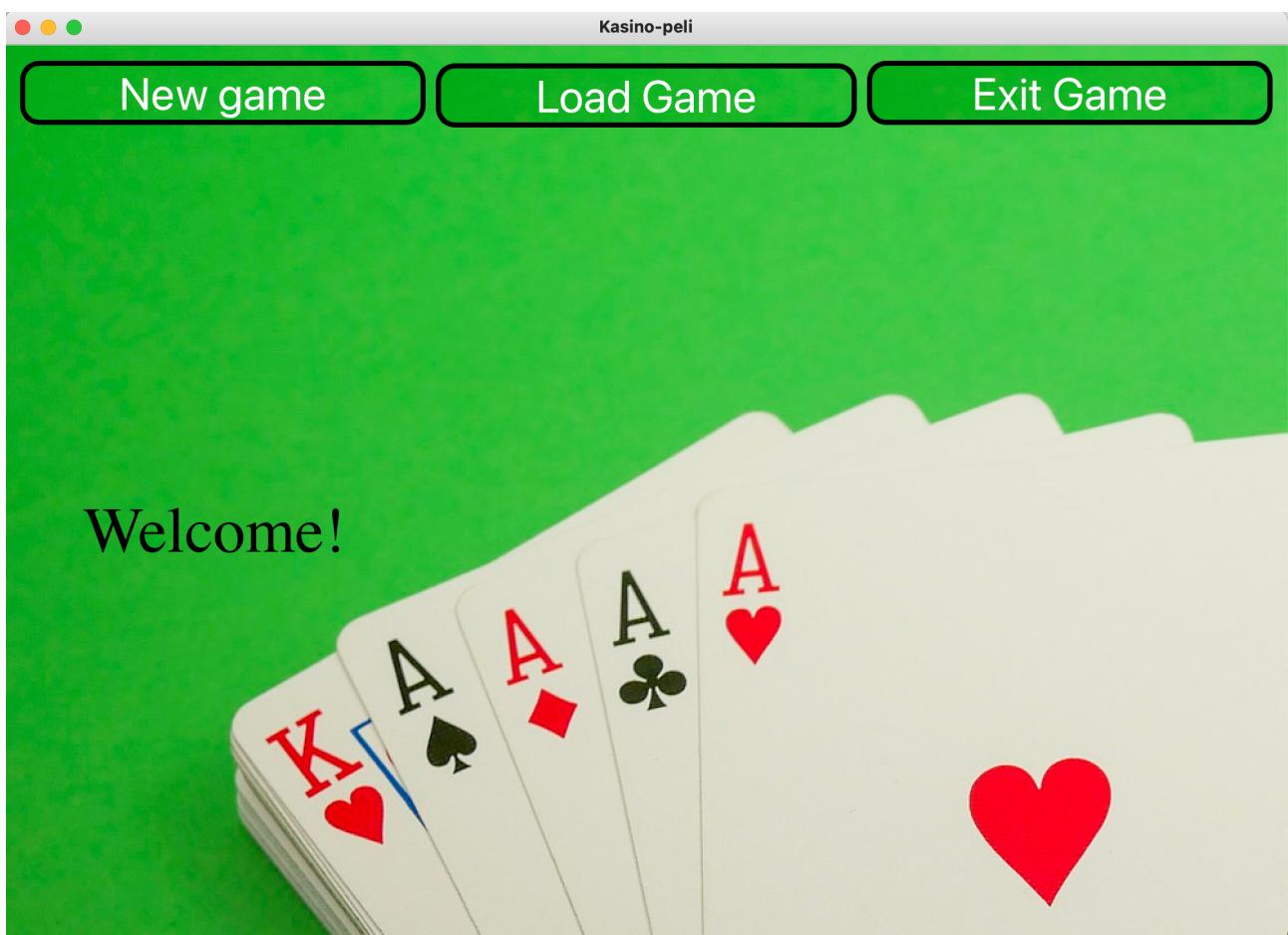
1. Henkilötiedot

Niko Somila, 710 578, Bioinformaatioteknologia, 6.5.2021

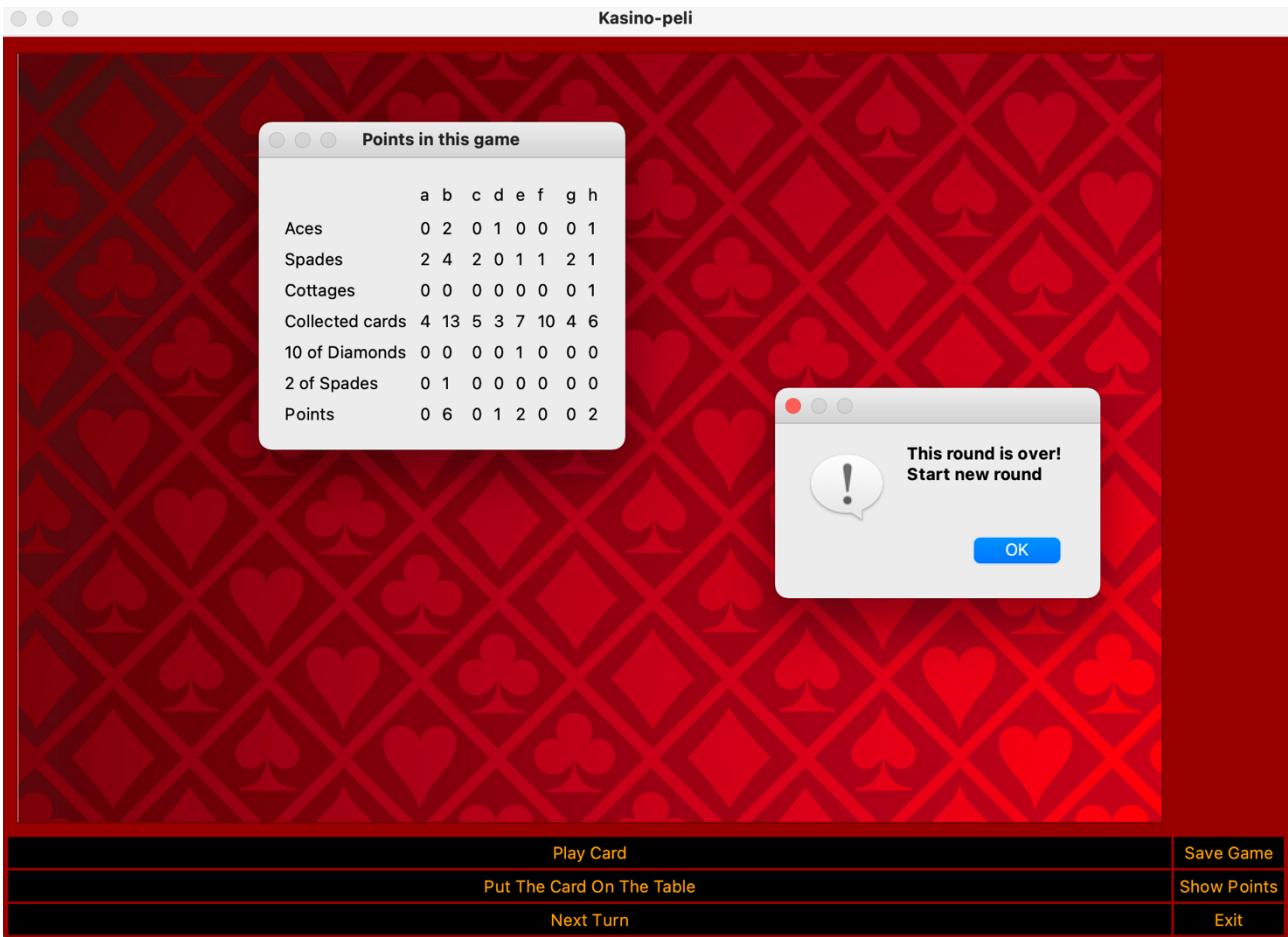
2. Yleiskuvaus

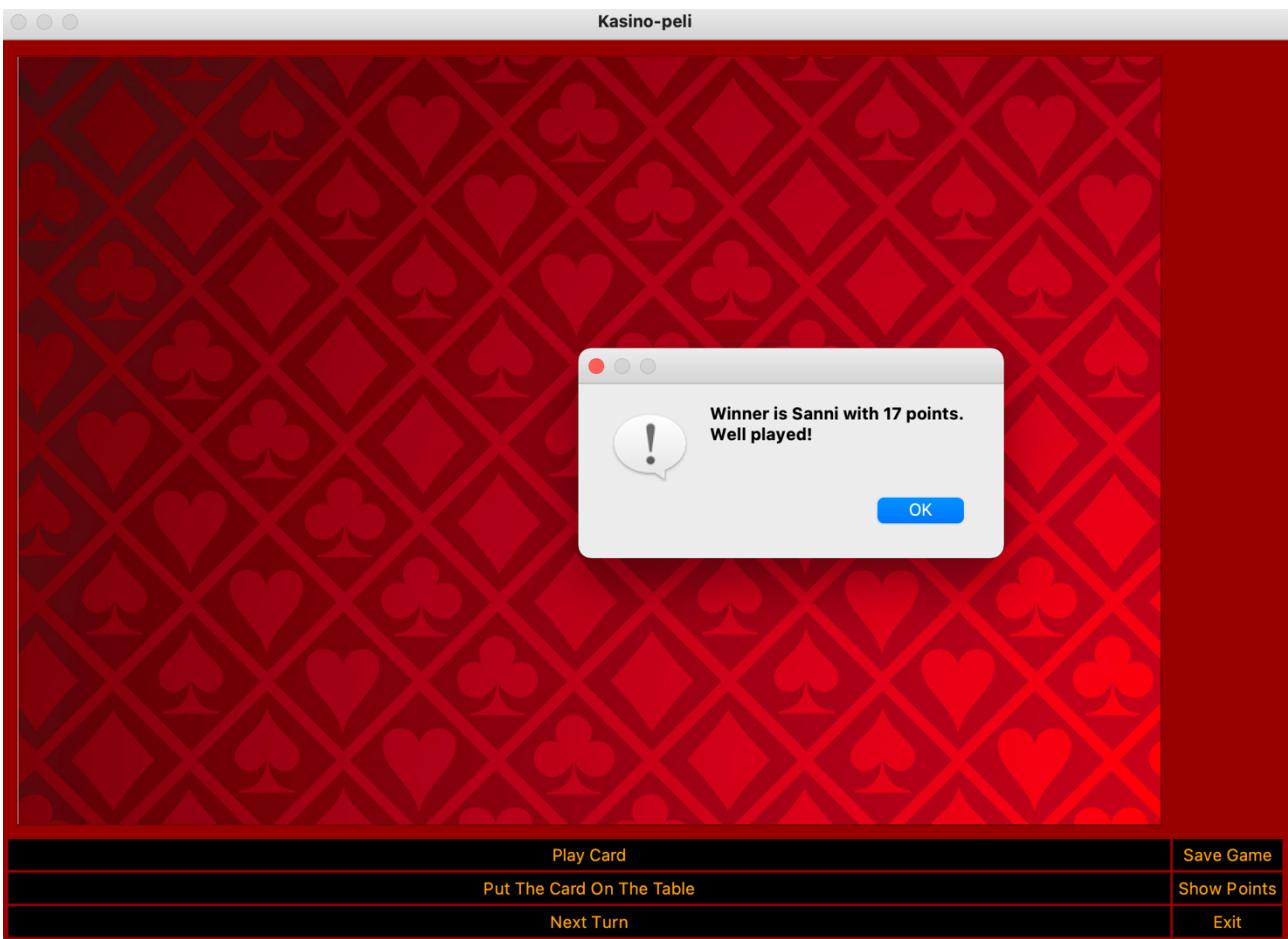
Tehtävänä on toteuttaa kasinokorttipeli ns. "pakkakasino" graafisena versiona. Itse toteutin työn keskivaikean version. Ohjelma luo mahdollisuuden valita 0-N verran pelaajia ja pelissä on mahdollista tallentaa ja ladata pelitilanteita.

3. Käyttöohje









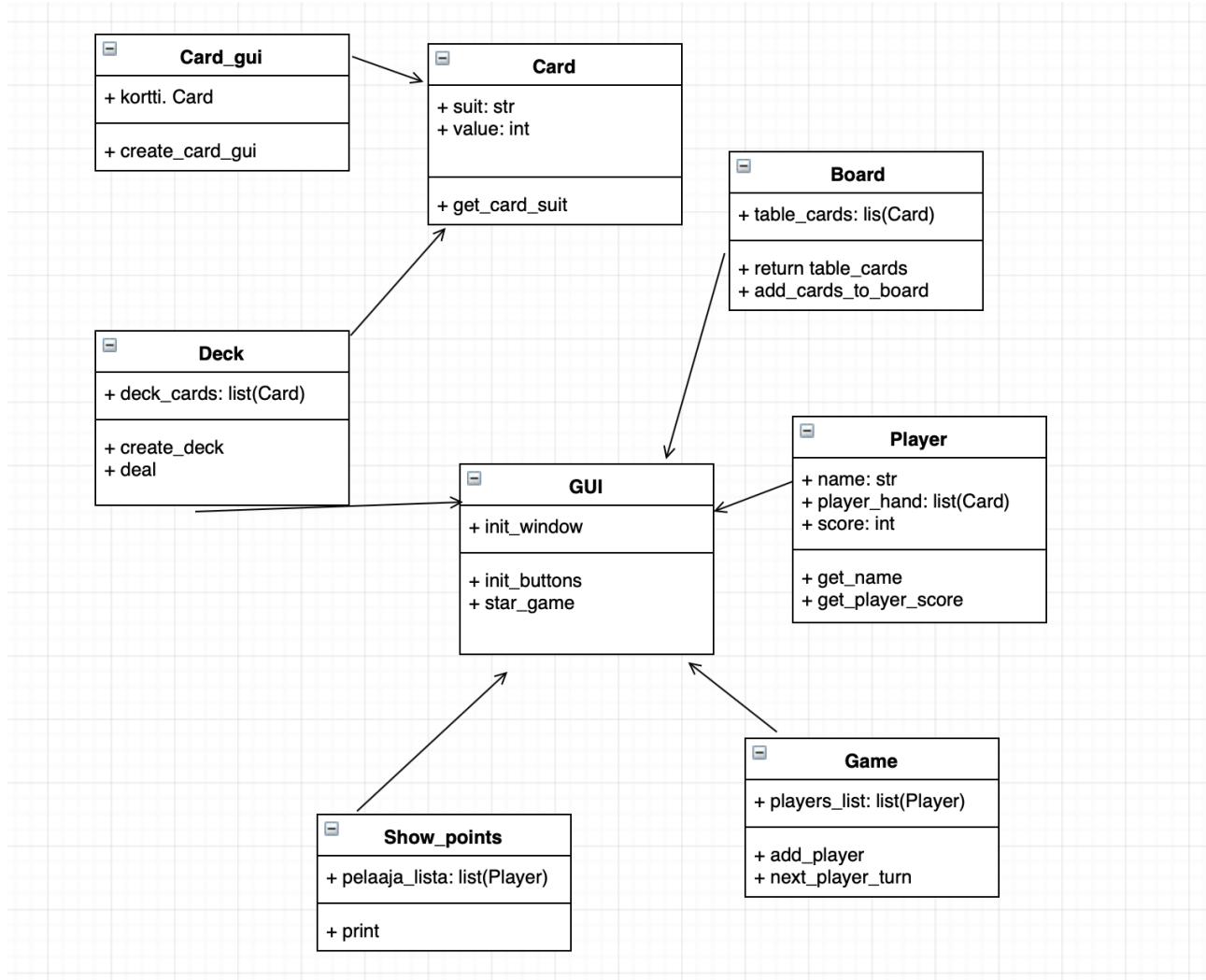
Ohjelma käynnistetään ajamalla main tiedostoa. Tiedoston ajamisen jälkeen ohjelma muodostaa MainWindow päälkkunan, jossa on valittavana kolme eri PushButtonia. Exit Game nappi lopettaa ohjelman. Load Game napilla ohjelma kysyy tallennustiedon nimeä, johon peli on tallennettu. Jos tallennustiedosta ei ole peli ilmoittaa tästä eikä avaa mitään. New Game napilla peli kysyy pelaajien määärää (2–8) jonka jälkeen pelaaja syöttää haluamansa pelaajien määrän. Jos pelaajien lukumäärä ei ole 2–8 peli pyytää antamaan oikean määrän. Seuraavaksi peli pyytää nimeämään pelaajat ja lisää pelaajat peliin tässä järjestysessä. Kun pelaajien nimet on lisätty, niin ohjelma siirtyy automaattisesti itse peliin. Tässä vaiheessa ohjelma luo pakan, sekoittaa sen ja jakaa kortit sekä pelaajille että pöytään. Pelissä vasemmasta yläkulmasta alkaen on pöytäkortit valmiiksi jaettuna. Vasemmassa alakulmassa lukee sen pelaajan nimi, sen vuoro on sekä paljonko kortteja pakassa on jäljellä. Peli näytön alakulmassa on myös pelaajan kortit näkyvillä. Sekä pelaajan, että pöydän kortteja voi klikata ja klikkauksen näkee siitä, että kortti hieman nousee. Pelaajalla voi olla samanaikaisesti vain yksi kortti valittuna, mutta pöytäkortteja voi olla useampikin valittuna. Peli näytön alapuolella on kuusi eri painiketta. Pelaaja voi vuorollaan pelata yhden kortin pöytään tai nostaa yhdellä kortilla pöydästä kortteja. Kun pelaaja on pelannut hän voi painaa Next Turn näppäintä, jolloin siirrytään seuraavaan pelaajaan. Tällöin entisen pelaajan kortit piilotetaan ja uuden pelaajan kortit näytetään. Peli ilmoittaa, jos sattuu mahdollisia virhetilanteita, eli esimerkiksi jos pelaaja ei ole pelannut vuoroaan ja painaa Next Turn tai jos pelaaja yrittää laittaa korttia pöydälle mutta ei ole valinnut kädestään yhtään korttia yms. Pelistä voi myös lähteä kesken kaiken pois Exit Game napilla. Pelin pisteen näytetään Show Points napista, joka näyttää pelaajien keräytä kortit, padat, erikoiskortit yms. ja jokaisen kierroksen loputtua pelaajille lasketaan pisteen ja näitä voi tarkkailla pistetaulukosta. Pelin

voi myön tallentaa Save Game napilla, jolloin peli pyytää tallennuksen nimeä. Nimen kirjoitettua peli tallentaa kyseisen peli tilanteen, jonka voi myöhemmin päävalikosta ladata. Kun pakasta loppuu kortit niin peli pyörii samalla lailla, mutta kortteja ei vain lisätä käteen. Kun kaikki kortit on pelattu peli ilmoittaa kierroksen loppuneen. Kun pelaaja klikkaa ok ilmoitukseen peli näyttää pistetilanteen ja uusi kierros alkaa automaattisesti. Peli loppuu siihen, kun kierroksen lopussa joku pelaajista on saanut 16 pistettä. Peli ilmoittaa tästä ja näyttää voittajan. Tämän jälkeen peliruudko tyhjennetään.

4. Ulkoiset kirjastot

Olen käyttänyt PyQt5 kirjastoa käyttöliittymän luomiseen, unittest kirjastoa yksikkötesteihiin, random kirjastoa pakan sekoittamiseen ja algoritmin luomiseen itertools kirjastoa.

5. Ohjelman rakenne



Ohjelmassa on useita perusluokkia, mutta itse pelin pyörittäminen tapahtuu käyttöliittymä luokassa (GUI). Käyttöliittymä luokasta kutsutaan toisia luokkia tarvittaessa. Korteilla on oma gui_luokkansa, jolla saadaan kortit näkyville peliin. Pisteillä myös oma gui_luokka,

jolla pistetaulukko saadaan näkyviin. Melkein jokaisessa luokassa on metodeja, millä voi muokata luokan tietoja.

Keskeiset metodit:

Card_gui: create_card_gui

- luo kortista näkyvän kuvan ohjelmaan

Card: get_card_suit, get_card_value

- palauttaa kortin maan sekä arvon

Deck: create_deck ja deal

- luo pakan sekä jakaa kortteja pakasta

Board: add_cards_to_board

- lisää kortteja pöytään

Player: get_name ja get_player_score

- palauttaa pelaajan nimen sekä pisteet

Game: add_player ja next_player_turn

- lisää peliin pelaajan ja siirtää pelaajan vuoron

Gui: init_window ja start_game

- luo pääikkunan ja peli-ikkunan

Show_points: print

- luo pistetaulukon ja näyttää sen

6. Algoritmit

Tärkein ohjelmani algoritmi on se, joka laskee, onko pelaajan korttien nosto pöydästä laillinen. Pelaaja saa nostaa pöydästä kortteja, jos kortti on samanarvoinen kuin pelaajan kortti, jos korttien summa on samanarvoinen kuin pelaajan kortti tai jos on useita kortti yhdistelmiä, joiden summa on pelaajan valitseman kortin kanssa samanarvoinen. Tämän toteutin aluksi ohjelmassa väärin, mutta loppupuolella tajusin virheeni ja tein uuden algoritmin. Algoritmi on hieman monimutkainen ja se on jaettu kolmeen osakokonaisuuteen. Algoritmi saa parametreikseen pelaajan valitseman kortin arvon sekä pelaajan valitsemat kortit pöydästä. Ensin ohjelma tekee kaikki mahdolliset yhdistelmät korteista, jotka pelaaja on valinnut ja nämä yhdistelmät voivat olla myös eripituisia. Seuraavaksi ohjelma ottaa jokaisen yhdistelmän erikseen ja summaa yhdistelmän sisällä olevien korttien arvot. Jos yhdistelmän arvo on sama kuin pelaajan valitsema käsikortti niin yhdistelmä lisätään listaan, koska se on silloin mahdollinen. Seuraavassa osakokonaisuudessa käytetään hyväksi äsknen löydettyjä yhdistelmiä. Tehdään näistä yhdistelmistä taas kaikki mahdolliset yhdistelmät ja etsitään ne, joiden korttien määrä on sama kuin pelaajan valitsemien korttien määrä. Eli siis lopputuloksen on lista korttiryhmistä, joiden pituus (korttien määrä) on sama kuin pöydästä valittujen korttien määrä. Viimeisessä osakokonaisuudessa ohjelma hyväksyy vain joukot, joissa kortti ilmenee vain kerran. Tämä suoritetaan niin, että kun kortti löydetään, se merkitään ylös ja lopuksi lisätään listaan, koska se on laillinen kortti. Ohjelman lopussa tarkastellaan laillisten korttien määrää ja jos laillisia kortteja on edes yksi, niin algoritmi on löytänyt lailliset kortit. Jos laillisia kortteja ei ole yhtään, niin silloin algoritmi on todennut, että valituista korteista ei saa laillista eikä niillä saa nostettua pöydästä kortteja.

Toinen ohjelmassa oleva pieni algoritmi on pisteiden lasku algoritmi. Eli kierroksen ja pelin lopussa lasketaan pisteet. Tässä algoritmissa kerätään pelaajan pelissä kerättyjä kortteja ja niiden perusteella annetaan pisteet.

7. Tietorakenteet

Ohjelmassa käytin pääsääntöisesti tietorakenteina listoja. Listat ovat dynaamisia tietorakenteita, koska niiden muisti muuttuu. Listat olivat minusta sopiva ratkaisu ohjelmaan, koska niitä on helppo luoda, niihin on helppo lisätä sekä poistaa alkioita, listojen pituutta on helppo tarkistaa ja niitä on helppo alustaa. Toinen tietorakenne, joka olisi voinut sopia ohjelmaan olisi ollut sanakirja, mutta en käyttänyt niitä ohjelmassa, koska en kokenut tarvitsemani sitä enkä osaa käyttää niitä yhtä hyvin kuin listoja. Toinen mahdollinen tietorakenne olisi myös ollut monikko (tuple), mutta en kokenut sitä hyödylliseksi tässä ohjelmassa.

8. Tiedostot

Ohjelmassa käytän tiedostoformaatteja pelin tallentamiseen. Tiedostoformaatit ovat tekstitiedostoja. Tiedostoon tallennan pelin tilanteen eli pelaajat, pelaajien pisteet, pelaajien korttit, pelaajien vuorot, siirrot yms. sekä pakan ja pöytäkortit. Pelin pystyy lataamaan samaan tilanteeseen kuin tallennushetkellä lukemalla tiedostoa. Tiedostolle pitää

tallentamisen yhteydessä antaa nimi ja pelin lataamisen hetkellä ohjelma etsii tiedostoja pelaajan antamalla tiedoston nimellä. Tiedoston ensimmäisellä rivillä on yksi numero, joka kertoo pelaajan vuoronumeron. Seuraavalla rivillä on #-merkki. Seuraavilla riveillä on kaikkien pelaajien tiedot järjestysessä ja tietojen välissä on välilyönti merkki. Näitä rivejä on siis yhtä monta kuin on pelaajia pelissä, ja lukemisen helpottamiseksi pelaajien tiedot loppuvat, kun vastaan tulee #-merkki. Rivit, joissa on pelaajien tietoja menevät seuraavasti, ensin tulee pelaajan nimi, sitten pelaajan kerättyjen korttien määrä, pelaajan pisteet, pelaajan mökit, pelaajan kerätyt padat, pelaajan kerätyt ässät, pelaajan mahdollinen ruutu 10, pelaajan mahdollinen pata 2, onko pelaaja nostanut pöydästä, onko pelaaja pelannut vuoronsa ja lopuksi #-merkki. Kaikki muut ovat lukuarvoja paitsi se, että onko pelaajan vuoro pelattu, joka on joko True tai False. #-merkin jälkeen kirjoitetaan pelaajan kädessä olevien korttien tiedot niin, että ensin tulee maa ja sitten arvo. Kaikki tiedot siis erotellaan välilyönnein. #-merkin jälkeen tiedostossa on pöydässä olevien korttien tiedot, jos niitä pöydässä oli tallennushetkellä. Jokaisen kortin tiedot ovat omalla rivillään ja menevät samoin kuin pelaajan korttien tapauksessa eli ensin maa ja sitten arvo ja erotetaan tiedot välilyönnein. Pöydän kortit loppuvat, kun uudella rivillä on #-merkki. Lopuksi tiedostossa on pakan kortit kirjoitettu samoin kuin pöydän kortit, eli jokaisen omalla rivillään, ensin maa ja sitten arvo ja tiedot erotettu välilyönnillä. Tiedoston lopettaa #-merkki.

```
1 | 0
2 | #
3 | Matti 9 0 2 3 1 1 0 1 True # D 13 C 10 C 8 S 13
4 | Teppo 0 0 0 0 0 0 0 False # D 1 C 5 S 10 D 3
5 | #
6 | D 12
7 | H 5
8 | S 9
9 | C 11
10 | #
11 | H 2
12 | D 9
13 | H 7
14 | S 8
15 | C 7
16 | C 4
17 | D 4
18 | H 1
19 | D 5
20 | H 11
21 | S 4
22 | C 9
23 | C 1
24 | D 7
25 | H 6
26 | D 11
27 | H 13
28 | C 6
29 | H 12
30 | H 3
31 | S 12
32 | H 4
33 | D 2
34 | C 2
35 | C 13
36 | D 8
37 | S 7
```

0	H 3
1	S 12
2	H 4
3	D 2
4	C 2
5	C 13
6	D 8
7	S 7
8	S 11
9	S 3
0	H 8
1	S 2
2	#

Ylemmät kuvat ovat esimerkki tallennustiedostoja.

9. Testaus

Ohjelmassa testasin samoja asioita kuin suunnitelmassa. Eli yksikkötestauksen avulla kokeilin ohjelman perusosia kuten onko pakka luotu oikein, lisätäänkö pelaajat oikein peliin, luodaanko kortit oikein, onnistuuko pelaajille korttien jako, onnistuuko pelaajien vuoronvaihdot ja kerääkö pelaajat esimerkiksi erikoiskortit oikein. Ohjelma läpäisee kaikki nämä. Kuitenkaan käyttöliittymää ei pystynyt yksikkötestauksilla testaamaan. Testasin käyttöliittymä tiedostossa olevia asioita paljon printtaamalla sopivii väleihin erilaisia tekstejä, jos halusin selvittää, mennäänkö ohjelmassa tiettyjen if-lauseiden läpi ja tietyissä tilanteissa erilaisia listojen pituksia tai alkiota, jolloin näin oliko niissä tiedot haluttuja. Testauksen suunnittelussa ei mielestäni ollut olennaisia aukkoja.

10. Ohjelman tunnetut puutteet ja viat

Ohjelmassa pelaajan kädessä olevat kortit toimivat niin, että kun valitset yhden kortin ja valitset kortin oikealla puolella olevan kortin, niin ohjelma automaattisesti valitsee uuden kortin ja vanha kortti palautuu ei valituksi automaattisesti niin kuin pitääkin. Toisaalta kun valitsee kortin vasemmalta puolelta, niin korttia ei pysty valita, vaan ensin pelaajan pitää ei valita alkuperäinen kortti ja sen jälkeen valita uusi kortti. Tämän korjaisin niin, että kun pelaaja valitsee uuden kortin niin ohjelma ensin epävalitsee kaikki pelaajan kortit ja sen jälkeen valitsee pelaajan valitseman kortin, jolloin ohjelma suorittaa pehmeän korttien valinnan.

Toinen vika on se, että kun peliä pelataan, peli tallenetaan ja ladataan uudestaan niin joskus pakan korttien määrää kuvaava widget näyttää pakan korttien lukumäärän epäselväänä, luultavasti sen takia, että sen alle on jänyt toinen sama widget, mutta erikokoisenä. Tämän

voisi korjata sillä, että loisin widgetin omassa funktiossa enkä aivan alussa, jolloin voisim kutsua sitä lataamisen yhteydessä ja se varmasti luotaisiin silloin ensimmäistä kertaa.

Yksi vika on myös se, että pelin alussa kysytään pelaajien lukumäärästä ja tätä ikkunaa ei voi lopettaa muuten kuin asettamalla sallitun lukumäärään. Eli ei pysty painaa cancel nappia tai lopettaa ruksista. Tämä johtuu siitä, että olen pistänyt while-loopin joka kysyy pelaajien lukumäärästä siihen asti kunnes se on ohjelman mielestä oikein. Tämän voisi korjata sillä että, ottaisi while-loopin pois ja laittaisi normaalim if-lauseen, joka tarkistaa sen, että pelaaja itse painaa ok nappia, jolloin peli menee eteenpäin. Sama ongelma on pelaajien lisäyksessä, mutta se ei ole mielestäni niin paha vika, koska silloin pelaaja on jo halunnut aloittaa pelin tiettyllä pelaaja määrellä.

11. 3 parasta ja 3 heikointa kohtaa

Mielestäni ohjelman parhaita puolia on sen ulkoasu, johon olen erityisen tyytyväinen. Olen ottanut huomioon ulkoasua ja nappuloita suunnittelessa monia pieniä asioita huomioon, jotka mielestäni tekevät pelistä miellyttävän näköisen. Näitä ovat mm. nappuloiden sijoittelu, nappuloiden värit sekä fonttit. Kun esimerkiksi vie hiiren nappulan luokse niin hiirestä tulee PointingHandCursor, joka mielestäni on hieno lisä. Mielestäni myös suunnittelini peli-ikkunan hienon näköiseksi, koska nappulat ovat selkeästi, pakan tiedot näkevät, pelaajan vuoron näkee ja korttien valinnat näkee selvästi. Olen myös tyytyväinen pisteiden laskuun ja siihen, että pistetaulukko on oma käyttöliittymänsä.

Heikkoina kohtina pidän sitä, että peli ei ole niin pelattava realistisesti. Eli esimerkisi jos haluaisi pelata peliä ystävien kanssa, niin pelivuoron vaihtuessa kaikki näkevät seuraavan kortit. Tämä ei ole pelin kuluksi hyvä, että muut pelaajat näkevät toisen kortit. Tämän voisi korjata niin, että asettaisi aina pelin aluksi kortit piiloon, niin että näkyisi vain kortin takapuoli. Lisäisin Show Cards napin, jolla omat kortit saisi näkyville ja silloin muut eivät näkisi heti omia kortteja. Toinen heikko kohta on pelin loppu. Itsellä käy niin, että pelin lopputta peli ilmoittaa voittajan, mutta sen jälkeen pelissä on vain tyhjä näyttö. Tähän kohtaan voisim laittaa napit Play Again ja Exit Game, jolloin pelaaja voi itse päätää aloittaako pelin uudestaan vai lopettaako pelin.

12. Poikkeamat suunnitelmosta

Tein koodin eri tavalla kuin suunnitelmassa olin miettinyt. Aloitin samalla lailla luomalla perusoliota. Suunnitelmosta poiketen käyttöliittymä pitäisi ottaa myös mukaan aikaisessa vaiheessa. Suunnitelmosta poiketen käyttöliittymäkoodissa meni enemmän aikaa kuin olin ajatellut. Tämä johtui siitä, että se oli haastavampi kuin olin ajatellut. Alkuperäisestä suunnitelmosta poiketen en myöskään tehnyt koodia tasaisesti, vaan jatin koodin tekemisen jossain välissä vähemmälle muiden kurssien kuormittavuuden takia ja aloin taas koodaamaan ennenmän palautusjan lähestyessä.

13. Toteutunut työjärjestys ja aikataulu

Aluksi loin perusoliot peliin eli Game, Pakka, Kortti, Pelaaja yms. Seuraavaksi aloin luomaan käyttöliittymää, koska peliä ei muuten pystynyt kokeilemaan kunnolla. Käyttöliittymässä loin ensin MainWindow näytön, johon lisäsin alkunäytön napit. Seuraavaksi loin pelinäytön ja lisäsin sinne kortit. Sen jälkeen loin pelin mekanismit. Viimeisinä asioina loin tallennuksen ja latauksem, sekä pisteiden laskemisen ja pisteiden

näyttämisen. Suunnitelmasta poikettiin siinä, että käyttöliittymä piti lisätä peliin jo alkuvaiheissa.

14. Arvio lopputuloksesta

Omasta mielestäni ohjelma on toteutettu hyvin. Se toteuttaa kaikki sille annetut kriteerit ja peli toimii muutenkin hyvin. Pelissä ei mielestäni ole mitään bugeja, tai en ainakaan itse ole niitä löytänyt ja itse pelin pelaamisen ei liity ongelmia. Käyttöliittymä on toteutettu hienosti ja mukaan on otettua pieniä yksityiskohtia parantamaan kokonaiskuvaaa. Puutteita ohjelman massassa kuitenkin on joitain. Ne eivät vaikuta pelin pelattavuuteen, mutta ne poistamalla peli toimisi vieläkin sujuvammin. Puutteita ohjelman massassa on esimerkiksi se, että pelaajan kortit eivät vasemmalle klikattaessa automaattisesti laske pelaajan ensimmäiseksi klikattua korttia. Tämän korjaisin niin, että kun pelaaja valitsee uuden kortin niin ohjelma ensin epävalitsee kaikki pelaajan kortit ja sen jälkeen valitsee pelaajan valitseman kortin, jolloin ohjelma suorittaa pehmeän korttien valinnan. Toinen vika on se, että pelin lopussa peliä ei voi heti aloittaa uudestaan vaan, ohjelma pitää ensin sulkea ja vasta sitten aloittaa peli uudestaan ajamalla ohjelma uudestaan. Tämän ongelman voisi korjata niin, että lisäisi loppuun pelin uudelleen aloittavan napin. Omasta mielestä myös itse koodin olisi voinut kirjoittaa järkevämmän, niin että pelistä vastaavat ominaisuudet ovat omassa luokassaan. Tällä hetkellä käyttöliittymä luokassa on suurin osa koodista ja sitä voi olla raskasta lukea ulkopuolisena. Korjaisin asian siis niin, että tekisin uuden luokan, missä suoritan pelin kannalta tärkeät asiat ja pitäisin käyttöliittymä koodin erikseen. Näin yksikkötestaus tulisi myös mahdolliseksi. Ohjelmaa voisi parantaa monellakin tavalla ja jos aikataulu olisi ollut erilainen olisin nämä muutokset ohjelman tehty. Yksi lisä mitä olisin halunnut lisätä olisi ollut pelimusiikki, jonka pelaaja voi itse laittaa päälle halutessaan. Toinen lisäys on myös se, että pelaaja pystyy vaihtamaan pelin taustaa muutamasta vaihtoehdosta. Olisin myös halunnut lisätä pelin loppuun pistetaulukon, joka lisää pelin taustalle taulukon, jossa on kaikki pelissä olleet pelaajat ja heidän pisteesä asetettuna suuruusjärjestykseen.

15. Viitteet

<https://plus.cs.aalto.fi/y2/2021/>

<https://doc.qt.io>

<https://www.youtube.com>

<https://stackoverflow.com>

<https://pixabay.com>

https://www.tutorialspoint.com/pyqt/pyqt_qmessagebox.htm

<https://pythonprogramminglanguage.com/pyqt5-message-box/>

16. Liitteet