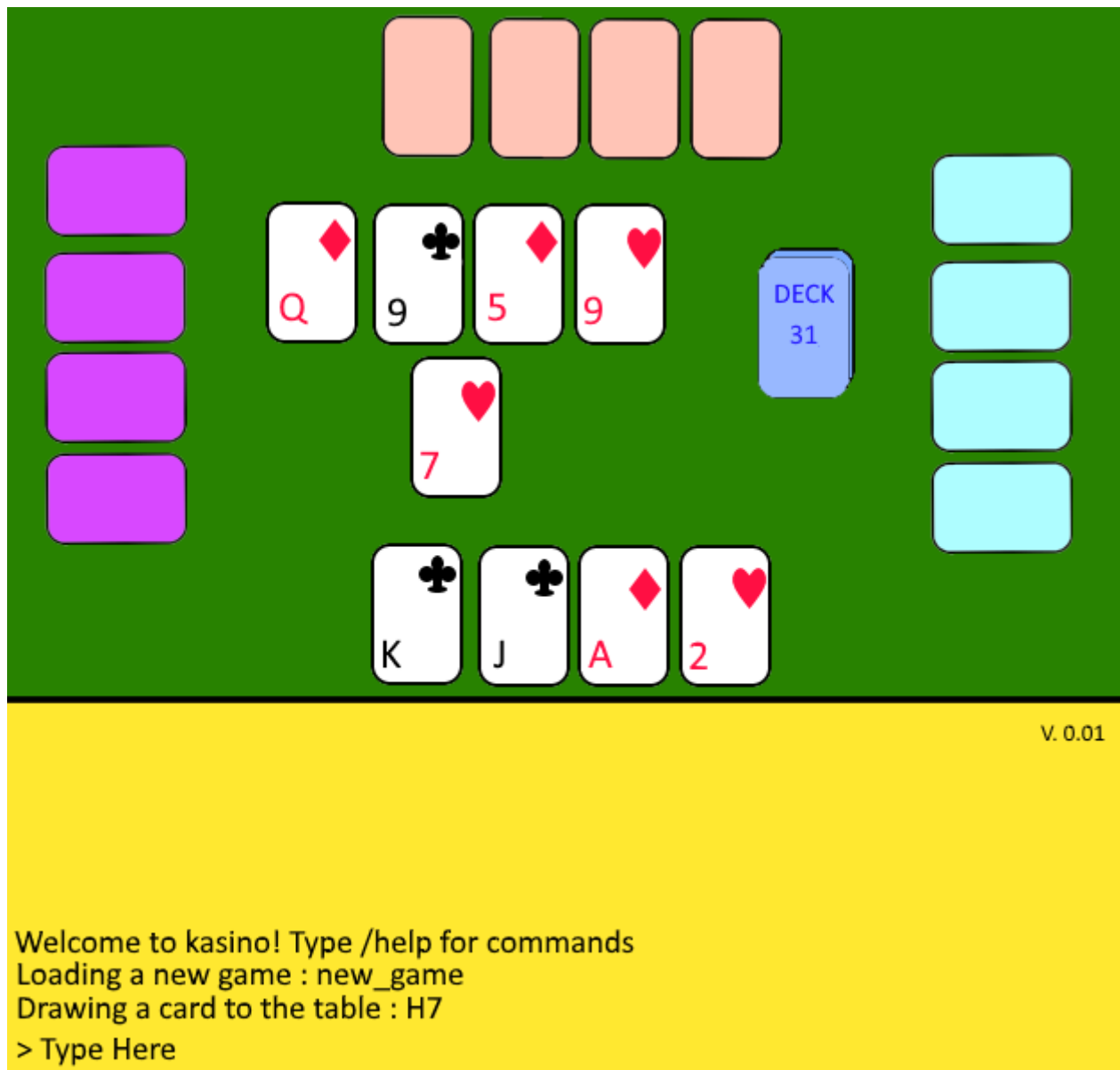


# Kasino

*Yleissuunnitelma*



*Konsepti siitä miltä pelilaudan kuuluisi näyttää valmiina*

*Lari Haapaniemi  
651624*

*Automaatio- ja Infomaatioteknologia*

*Vuosikurssi 2018*

*20.2.2019*

# Yleiskuvaus

Valitsin projektiaiheekseni kasino pelin jonka toteutuksena toimii ns. ”pakkakasino”. Projektin ideana on luoda PyQT5:lla mallinnettu korttipeli, jossa alustavasti on 2-4 pelaajaa pöydässä. Jokaiselle pelaajalle jaetaan 4 korttia kuvapuoli alaspäin, ja pöydälle 4 korttia, jotka ovat kuvapuoli ylöspäin. Loput kortit kasataan pöydälle pinoon kuvapuoli alaspäin.

Jakajana toimii aina yksi pelaaja, joka valitaan arpoen. Jakajana toiminen on vain muodollisuus, jolla pystytään määrittämään seuraava pelaaja.

Konseptinani on tehdä keskivaikea versio tästä projektista, mutta aion yrittää myös mahdollisuutta pelata tietokonetta vastaan, joka toimii min-max algoritmilla.

Varsinaiset pelin säännöt lainattuna suoraan tehtävänannosta:

## Mitä kukin pelaaja tekee vuorollaan

*Pelaaja voi kerrallaan käyttää jonkin kädessään olevista korteista: joko ottaa sillä pöydästä kortteja tai laittaa kortin pöytään. Jos pelaaja ei voi ottaa mitään pöydästä täytyy hänen laittaa jokin korteistaan pöytään.*

*Jos pelaaja ottaa pöydästä kortteja hän kerää ne itselleen pinoon. Pinon sisällöstä lasketaan korttien loputtua pisteet.*

*Pöydässä olevien korttien määrä voi vaihdella vapaasti. Jos joku vaikkapa ottaa kaikki kortit, täytyy seuraavan laittaa jokin korteistaan tyhjään pöytään.*

*Aina käytettyään kortin, pelaaja ottaa käteensä pakasta uuden, niin että kädessä on aina 4 korttia. (Kun pöydällä oleva pakka loppuu, ei oteta enää lisää vaan pelataan niin kauan kuin kenelläkään on kortteja kädessä. Tällöin siis tietenkin alle 4 korttia on mahdollinen tilanne.)*

*Kortilla voi ottaa pöydästä yhden tai useampia samanarvoisia kortteja ja kortteja, joiden summa on yhtä suuri, kuin kortin jolla otetaan. Seuraava esimerkki selventää asiaa:*

*Jos joku saa kerralla pöydästä kaikki kortit kerralla, hän saa ns. mökin, joka merkitään muistiin.*

## Erikoiskortit

*Pelissä on muutama kortti, joiden arvo kädessä on arvokkaampi kuin pöydässä,*

*Ässät : kädessä 14, pöydässä 1*

*Ruutu-10: kädessä 16, pöydässä 10*

*Pata-2: kädessä 15, pöydässä 2*

*Muiden korttien arvot ovat samat sekä pöydässä että kädessä.*

## Pistelasku

*Kun kaikilta loppuvat kädestä kortit saa viimeksi pöydästä kortteja ottanut loput pöydässä olevat kortit.*

*Tämän jälkeen lasketaan pisteet ja lisätään ne entisiin pisteisiin.*

*Seuraavista asioista saa pisteitä:*

*Jokaisesta mökistä saa yhden pisteen*

*Jokaisesta ässästä saa yhden pisteen*

*Eniten kortteja saanut saa yhden pisteen*

*Eniten patoja saanut saa 2 pistettä*

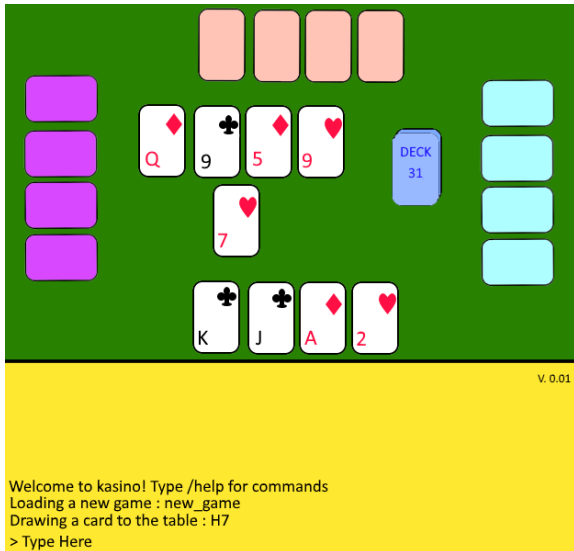
*Ruutu-10 kortin omistaja saa 2 pistettä*

*Pata-2 kortin omistaja saa pisteen*

## Käyttöliittymän luonnos

Valitsin projektiaiheekseni kasino pelin jonka toteutan tekstikentän kautta, kuten ensimmäisellä sivulla näkyvässä kuvassa näkyy. Ohjelma saa ennalta määrättyjen käskyjen mukaan inputtinsa, jotka voidaan tulostaa ruudulle /help käskyllä. Tiedettyjä käskyjä ovat ainakin draw, play[cardfromhand, cardsfromtable], points, load\_game, save\_game ja new\_game.

Uuden pelin aloittaessa pelaajalta kysytään luultavasti useassa osassa tarvittavat tiedot, esimerkiksi pelaajien määrä, pelaajien nimet ja mahdollisesti mahdollisuus tietokonevastuksiin.



Oheisessa kuvassa nähdään myös, että jokainen kortti piirretään ruudulle PyQt:n komentojen avulla. Käytössä ei ole siis valmiita png kuvia jotka ladataan fileen, vaan tulostetaan aluksi ellipsi, johon lisätään numero ja symboli. Kortin paikka määritetään x-y kordinaatistolla pöydällä olevien korttien mukaan. Tämä ruutu tulee tulemaan luultavasti pelin ainoa ruutu, ja kenttä on luonnollisesti tyhjä jossei peliä ole ladattu. Tämä on siis tämänhetkinen suunnitelma, joka saattaa vaihtua projektin kuluessa.

## Tiedostot ja tiedostoformaatit

Ohjelman kannalta toimisi varmaan parhaiten samantyylinen ohjelmanlukutapa kuin harjoituskierroksen tehtävässä 4.4, poislukien alussa annettua datan pituutta (tällä siis tarkoitan sitä, että ohjelmassa on ennalta määrättyt lohkot, joissa on tietty sisältö). Tiedostomuoto on CSV, jossa uutta ”lohkoa” merkitään #. Esimerkkinä tästä tiedoston luku:

```
#Gamedata;Playeramount;4;#;Boardstate;Cards;5;SQ;R9;S5;H9;H7;Deck;31#;Playerdata;Player1;Joni;Cards;4;RK;RJ;SA;H2:Points;0#
```

Tässä siis luetaan ensin pelin pelaajien määrä, tämän jälkeen pöydän tilanne, ja lopuksi vuorotellen jokaisen pelaajan käsi ja pistetilanne. Filen loppuun luultavasti kannattaisi lisätä pelatut kortit, tai mahdollisesti boardstateen, jotta voidaan tarkistaa onko pelaajilla tai pöydällä olevat kortit pelattu jo. Helppo tapa tehdä tämä olisi luultavasti määrittää ennalta peliä luodessa missä järjestyksessä kortit ovat pakassa, ja poistaa jokaisen noston jälkeen yksi kortti listassa.

## Järjestelmäsuunnitelma

Testaamisessa varmaan tärkeitä asioita on katsoa miten ohjelma reagoi tuntemattomaan ohjelmalohkoon, tuntemattomaan korttiin, yli 4 pelaajaan ja yli 52 korttiin pakassa. Myöskin reagointi tuntemattomiin käskyihin tai väärinkäytetyt käskyt eivät saisi aiheuttaa ongelmaa ohjelman toiminnassa ja esimerkiksi isot kirjaimet pitäisi sallia käskyissä. Luonnollisesti monien eri pelien latausta ja tallennusta pitää testata, ja miten ne piirtyvät ohjelmaan.