

KASINO

Tomi Panula
Tietotekniikka, 1.vuosikurssi
527127
tomi.panula@aalto.fi
26.4.2017

Yleiskuvaus projektista

Tein kasino korttipelin graafisella käyttöliittymällä. Ohjelmassa on mahdollista valita tietokonevastustajien määrä(1-9), tallentaa ja ladata pelitilanteita sekä tietenkin pelata itse peliä. Projektissa on toteutettu kaikki vaativan tason vaatimukset.

Pelin säännöt ovat seuraavat:

- Pelissä kerätään pisteitä, jotka lasketaan aina jokaisen pelikierroksen lopussa.
- Peli jatkuu kunnes joku pelaajista saavuttaa 16 pistettä.

Pelikierros :

- Jokaisen pelikierroksen alussa pakka sekoitetaan ja jakaja jakaa jokaiselle pelaajalle 4 korttia (eivät näy muille) sekä 4 korttia pöytään (näkyvät kaikille). Loput kortit jätetään pöydälle pinoon ylösalaisin.
- Jakajasta seuraava aloittaa pelaamisen. Seuraavalla pelikierroksella hän on jakaja.

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:

Esimerkki

Pöydässä on kuusi korttia:

Ruutu-6	Risti-5
Herttajätkä-11	Pataässä (1)
Risti-10	Risti-3

Pelaajalla on kädessään seuraavat kortit:

1.	2.	3.	4.
Herttakuningatar (12)	Risti-4	Ruutu-2	Patajätkä (11)

1. Pelaaja voi ottaa pöydästä Herttakuningattarella (arvo 12) seuraavat kortit:
Herttajätkä (11) + Pataässä (1) = 12
tai
Ruutu-6 + Risti-5 + Pataässä (1) = 12
(molempia ei voi ottaa, koska niissä on sama kortti (Pataässä))
2. kortilla Risti-4 voi ottaa Pataässä ja ristikolmosen
3. kortilla Ruutu-2 ei saa pöydästä mitään
4. Patajätkällä (11) saa pöydästä kortit
Ruutu-6 + Risti-5 = 11
ja
Herttajätkä (11) = 11
ja
Risti-10 + Pataässä = 11

Tilanteessa pelaajan siis kannattaa valita patajätkä, jolloin hän saa itselleen pelaamansa kortin (patajätkä) ja pöydästä ottamansa viisi korttia. Pöytään jää tällöin vain ristikolmonen jäljelle.

Mökki

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
- esim : Ruutukympillä voi siis ottaa pöydästä vaikkapa kaksi kahdeksikkoa.
- 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 :

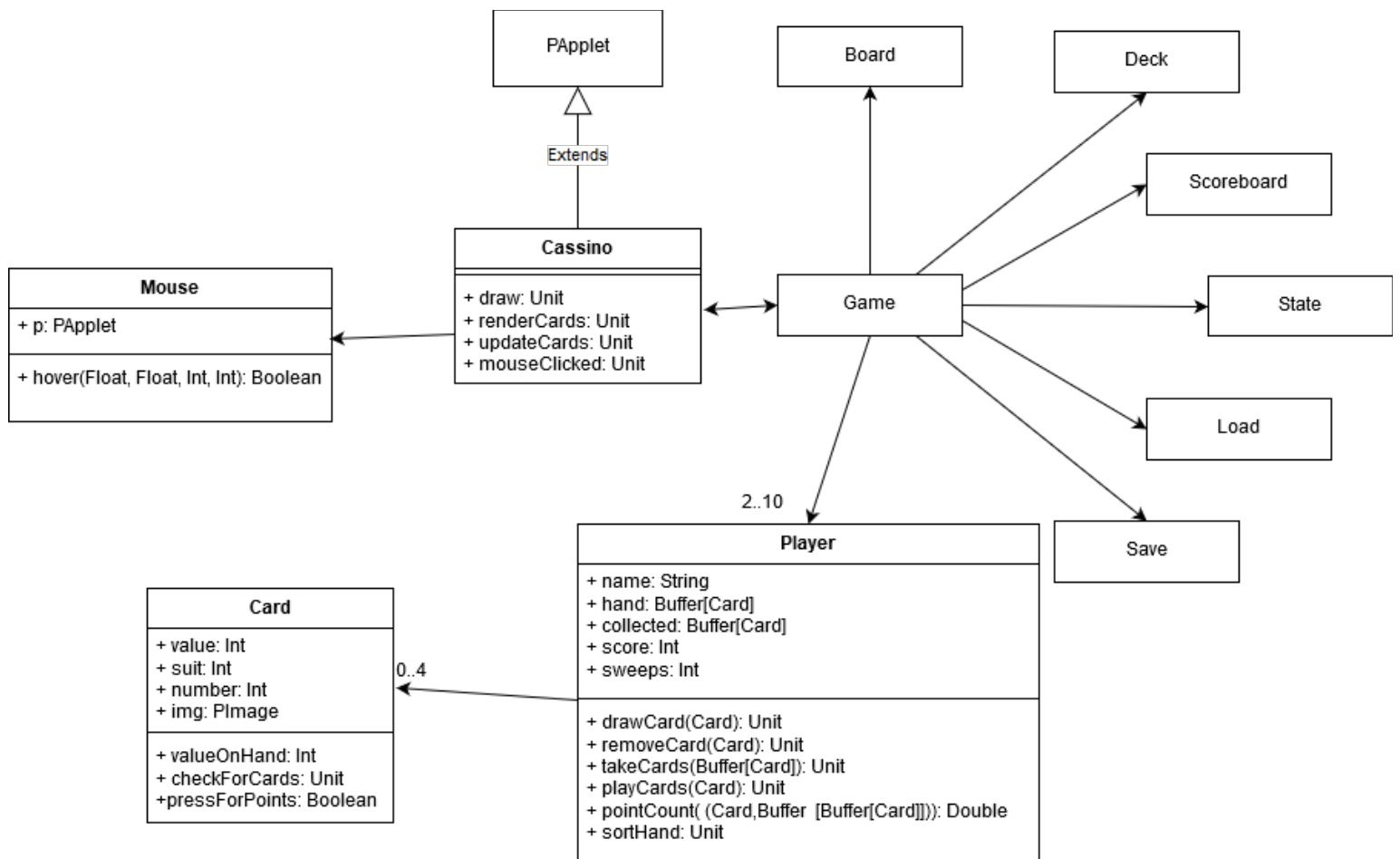
- Jokaisesta mökistä saa yhden pisteen
- Jokaisesta ässästä saa yhden pisteen
- Eniten kortteja saanut saa yhden pisteen
- Eniten patoja saanut saa 2 pistettä
- Ruutu-10 kortin omistaja saa 2 pistettä
- Pata-2 kortin omistaja saa pisteen

Käyttöohje

Kun ohjelman Casino.scala-tiedosto on ajettu Eclipsestä, niin ruudulle ilmestyy ikkuna, jossa on näkyvä alkuvalikko. Alkuvalikosta löytyy napit: "Load game", "New game", "Help" ja "Exit". Ensimmäisellä pelikerralla kannattaa lukea help napin takaa löytyvät ohjeet. Help-osiosta voi myös muuttaa tietokonevastustajien määrää(1-9). Kun käyttäjä on lukenut tarjotut vinkit niin on aika aloittaa peli.

Painamalla "New game" nappia voi aloittaa uuden pelin. Peli jakaa pelaajille ja pöytään kortit ja uusi peli alkaa aina käyttäjän vuorolla. Käyttäjä voi mahdollisuuksien mukaan joko valita ensin pöydästä kortteja joiden summa vastaa jotain omaa käsikorttiaan tai raahata yhden käsikortistaan keskelle ruutua ja päästää irti jolloin kortti menee pöytään ja on seuraavan pelaajan vuoro. Korttien loputtua pakasta sekä pelaajien käsistä, alkaa uusi kierros ja aloitusvuoro siirtyy myötöpäivään seuraavalle pelaajalle tai jos jollain pelaajalla on pisteitä 16 tai enemmän niin voittaja on löytynyt. Tällöin peli loppuu ja käyttäjä pääsee uuteen peliin painamalla oikeasta yläkulmasta löytyvää "Menu" painiketta. "Menu"-painiketta voi painaa myös missä tahansa vaiheessa peliä, jos haluaa pelin tauolle tai tutkia vaikkapa help-osiota. Menuun pääsee pelin aikana myös painamalla P-nappia näppäimistössä. TAB-näppäin avaa pistetaulukon pelin aikana. Ikkunan vasemmassa laidassa on pelihistoria, josta käyttäjä voi katsella mitä kortteja muut pelaajat pelasivat pöytään tai ottivat pöydästä.

Ohjelman rakenne



Ohjelman UML-kaavio

Pelin sujuvasta etenemisestä vastaa Game-objekti, joka pitää sisällään metodeita esimerkiksi seuraavan pelaajan vuoroon siirtymiselle, uuden kierroksen alkamiselle ja kokonaan uuden pelin alustamiselle. Game-objekti sisältää myös monia tärkeitä muuttujia kuten tiedon siitä onko peli käynnissä ja listan pelissä olevista pelaajista.

Board-objekti sisältää algoritmit mahdollisten korttikombinaatioiden löytämiselle, jotka pelaaja voi ottaa pöydästä tietyllä kortilla. Card-luokasta voi tehdä yksittäisiä korttiolioita antamalla parametreiksi kortin maan, arvon ja kortin kuvan. Card-luokasta löytyy metodit mahdollisten korttikombinaatioiden löytämiselle pöydästä kyseiselle kortille. Player-luokasta saa tehtyä yksittäisiä pelaajaoliota antamalla parametreiksi nimen, pelaajan keräämät kortit ja pelaajan pisteet. Pelaaja-luokasta löytyy metodeita kortin nostamiselle ja pelaamiselle, pisteiden lisäämiselle, korttikombinaatioiden "yhteispisteiden" laskemiselle, parhaan korttikombinaation valitsemiselle sekä käden järjestämiselle.

Algoritmit

Pöydästä löytyvien korttikombinaatioiden etsintäalgoritmi etsii kaikki mahdolliset kombinaatiot mitä pöydästä voi ottaa kullakin kortilla. Algoritmi toimii rekursiivisesti aina pienentäen haluttua summaa otetun kortin arvon verran. Algoritmi lisää korttikombinaation bufferiin jos se vastaa annettua alkuperäistä summaa.

Toinen algoritmi ottaa parametriksi edellisen algoritmin tarjoaman bufferin ja tekee bufferin kombinaatioista uuden bufferin, jossa on tehty pari kortista ja sitä vastaavista kombinaatioista siten että sama kortti ei esiinny kahta kertaa yhdessä kombinaatiossa.

Kolmas algoritmi tarkistaa, jos kaikki pöydästä valitut kortit kuuluvat johonkin mahdollisista kombinaatioista kyseiselle kortille ja palauttaa tätä vastaavan boolean arvon. Jos arvo on tosi niin pelaaja voi korttia painamalla kerätä kortit pöydästä ja ansaita pisteitä.

Neljäs algoritmi järjestää pelaajan käden siten että ensimmäisenä on kortit joiden yhdessä pöytäkorttien kanssa muodostama summa on yli 16(yrittää vähentää helppojen mökkien mahdollisuutta) ja eivät ole pistekortteja(padat, ässät, patakakkonen ja ruutukymppi). Seuraavaksi tulevat kortit jotka eivät ole pistekortteja mutta muodostavat pöytäkorttien kanssa summan joka on pienempi kuin 16. Sitten tulevat padat, ässät, patakakkonen ja ruutukymppi.

Tärkeitä algoritmeja ovat myös algoritmien joita käyttämällä saa muodostettua stringistä kortin, esimerkiksi "h04" → herttanelonen ja c10 → ristikymppi. Load-olio osaa nimensä mukaisesti ladata tekstitiedostosta informaatiota peliä varten sopivasti splittaamalla tekstirivejä ja tulkitsemalla annettua dataa. Tekstitiedostoon tallennetaan esimerkiksi tiedot pelaajien korteista, pisteistä ja pakassa ja pöydässä olevista korteista.

Monet ylläolevista algoritmeista käyttävät esimerkiksi scalan valmista sortBy-metodia, jota olisin halunnut käyttää vähemmän sen hitauden vuoksi. En kuitenkaan keksinyt muutakaan hyvää keinoa toteuttaa tarvittavat algoritmit joten tyydyin sortBy-metodiin.

Tietorakenteet

Käytin projektissani enimmäkseen buffer-kokoelmia, koska esimerkiksi pelaajien kädet, pöydän kortit ja pelaajien keräämät kortit ovat kaikki koko ajan muuttuvia. Muutaman napin tekstit ovat Vector-kokoelmissa, koska niitä ei tarvitse muuttaa ohjelman käynnissäolon aikana. Korttipakka on toteutettu Stack-tietorakenteella, koska se kuvaa parhaiten oikean elämän korttipakan koostumusta ja on siksi helpoin käyttää myös ohjelmoidessa korttipakkaa.

Tiedostot

Pelin tallentaminen hoituu kirjoittamalla tallennustiedostoon tarvittavat tiedot. Tämä hoituu painamalla pelin aikana "Save and exit" nappia pausevalikossa.

tallennustiedosto on tekstimuodossa ja sieltä tulisi löytyä seuraavat tiedot:

jokaiselle pelaajalle nimi, käsikortit, kerätyt kortit, pisteet, kierroksella saadut mökit.

Pelin muita tarvittavia tietoja ovat pöydässä olevat kortit, pakassa olevat kortit, kierroksen vuorohistoria, kenen vuoro on, kuka on jakaja sekä pelaajien määrä.

Tässä esimerkki pelitilanteesta tekstimuodossa:

#Player

hand: c10,h07,c12,s08

collected: d02,c02,c04,s04,s09,c09

score: 1

sweeps: 0

#CPU 1

hand: c07,d03,c05,c13

collected: h12,s07,s05,s06,h06,d08,c08,h11,s11

score: 5

sweeps: 3

#CPU 2

hand: h13,h01,s01,c01

collected: s03,h03

score: 2

sweeps: 0

#CPU 3

hand: s12,d01,s02,c06

collected:

score: 2

sweeps: 0

#GAME

board: d05,h08,c11

deck: d11,d07,d12,c03,h10,h04,s13,d13,h02,h09,d06,d09,s10,d04,d10,h05

history: Player;h08,CPU 1;h11-s11,CPU 2;d05,CPU 3;c11

turn: 1

dealer: 0

playerCount: 4

Ensin pelaajien tiedot (#pelaajan nimi aloittaa uuden pelaajan tiedot). Kortit ovat muodossa: maan lyhenne(h=hertta, s=pata, d=ruutu, c=risti) ja sen perässä kortin arvo pöydässä(esim ässä = 01). Alunperin oli tarkoitus tehdä erilainen tallennustiedostoformaatti johon olisi vaadittu että kaikki korttien nimet ovat yhtä pitkät, sen vuoksi esim 03 eikä 3. Pelaajien jälkeen tulee pelin loput tiedot #GAME

otsikon jälkeen. Historiatiedot ovat muotoa pelaajanNimi;korttienNimet-väliViivoilla-erotettuina, seuraavanPelaajanNimi;kortti-kortti-kortti jne. Ohjelma ei sisällä erillisiä asetustiedostoja.

Testaus

Ohjelmaa ei ole testattu muuta kuin pelaamalla peliä ja koittamalla etsiä bugeja rajatapauksista esim. tyhjä pöytä tai pakka.

Ohjelman tunnetut puutteet ja viat

Tietokonevastustajilla kestää todella kauan löytää paras kombinaatio pöydästä, jos kortteja on pöydässä n. 7 tai enemmän. Tällöin peli jäätyy hetkeksi, mutta tätä ei ole tapahtunut kertaakaan normaalissa pelissä, vain jos tilanne on erikseen luotu. Tilannetta parantaisi varmasti kevyemmät kombinaatioiden etsimis algoritmit.

Tallennustiedostot ei siedä virheitä kovin hyvin ja saattaa antaa virheitä, jos tekstitiedosto on virheellinen. Tätäkään ei pitäisi tapahtua normaalissa pelissä kun pelitilanne tallennetaan pelin sisältä. Latausmetodi on melko yksinkertainen, sitä voisi parantaa kestävämmän virheitä paremmin tekemällä siitä monimutkaisemman.

Päätin jättää pelistä animaatiot pois kokonaan, tämä vaikeuttaa vastustajien vuorojen seuraamista, mutta ongelmaa on yritetty korvata ikkunan vasemmassa laidassa olevalla vuorohistorialla.

3 parasta ja 3 heikointa kohtaa

Itse pidän käyttäjän vuoron pelaamisen toteutuksesta. Kortin raahaaminen pöytään toimii hyvin ja korttien ottaminen pöydästä on (ainakin allekirjoittaneen mielestä) yksinkertaista ymmärtää.

Vuorohistoria vasemmassa reunassa toimii testausten perusteella hyvin ja sitä kautta on myös helppo seurata peliä.

Vastustajien dynaaminen lisääminen ohjelman ollessa käynnissä on myös hyvin onnistunut. Tietenkään pelaajat eivät ilmesty jo meneillään olevaan peliin, mutta uuteen peliin kaikki vastustajat piirtyvät ja toimivat hyvin.

Yksi heikoimmista ohjelman osista on animaatioiden puuttuminen. Olisin halunnut lisätä niitä enemmän, mutta niiden tekeminen ei mahtunut aikatauluun. Animaatioiden puuttuminen saattaa tehdä pelistä hankalasti seurattavaa, mutta vuorohistoria ja pistetaulukko(TAB) auttavat jonkin verran.

Tekoäly ei ole haluamallani tasolla. Tähän vaikuttaa paljon omat tietoni pelin parhaista taktiikoista.

Poikkeamat suunnitelmasta ja aikataulusta

Suunnitelmasta löytyvä luokkajako muistuttaa paljon lopullista luokkajakoa. Pari luokkaa on poistunut niiden tarpeettomuuden takia. Suunnitelman käyttötapakuvaus ei eroa juuri ollenkaan lopullisen pelin käytöstä.

Alkuperäinen suunnitelma oli, että numerot 0-51 vastaa jokainen yhtä korttia korttipakassa. Lopullisessa versiossa jokaista korttia vastaa yksi Card-luokan ilmentymä, joka on paljon parempi tapa kuvata korttia yksittäisen numeron sijaan.

Tekoälystä ei tullut yhtä hyvä kuin suunnittelin, mutta ehkä suunnittelin liian hyvää tekoälyä, joka laskisi mitä kortteja on jäljellä ja sitä kautta eri vaihtoehtojen todennäköisyyksiä.

Pelin tallennusformaatti muuttui täysin. Suunnitelmassa oli kerrottu ihmiselle erittäin hankalasti luettavissa olevasta formaatista, joka vaihtui helpommin luettavaksi ja ymmärrettäväksi.

Suunniteltu aikataulu ei vastannut todellisuutta. Todellisuudessa työtunnit olivat lähinnä viimeisellä kahdella viikolla toisin kuin suunnitelmassa. Tallennuslogiikan tekemiseen meni huomattavasti vähemmän aikaa kuin olin suunnitellut. Lisäominaisuudet sen sijaan veivät ylimääräistä aikaa.

Arvio lopputuloksesta

Lopputulos mielestäni hyvä. Suurimpana puutteena itse pidän animaatioiden puuttumista. Ne olisivat tehneet pelistä hienomman ja miellyttävämmän pelata. Animaatioiden korvaamiseksi pelissä on ominaisuuksia joiden avulla peliä on helpompi seurata. Muuten pelaaminen on sujuvaa ja yksinkertaista. Kaikki interaktiot käyttöliittymän kanssa toimii hyvin. Napit toimivat kuten pitääkin ja peli ei pitäisi ainakaan kaatua missään vaiheessa. Ainoastaan jos käyttäjä muokkaa tallennustiedostoa käsin voi syntyä virhetilanteita. Ulkonäöltään peli ei ole mitenkään kaunista katseltavaa, mutta sitä osiota ei myöskään ole painotettu tehtävänannossa.

Viitteet

[processing](#)

[wikipedia casino](#)