

Daniel Espefält
Mail danne_190@hotmail.com
studentId: daes7720

Uppgift 4

För att kompilera och köra programmet så skriv "javac *.java" följt av "java Solly".

Programmet är uppbyggt av Klasserna Solly, Ritpanel, Deck, Foundation, Takepile, Tableau, Wastepile och Cards Bibliotek som används javax.swing, java.awt, java.util., java.io.File;

Bugg: När man flyttar ett kort från takepile till en tableau så kan andra kortet i takepile inte gå att ta och flyttas över till wastepile.

Genomgång av klasserna:

Deck

Deck är den abstrakta klassen för alla former. Attributen som Deck har är int x som är positionen i x-led för formen ,int y, int width , int height , Rectangle rec och Stack Cards.

Decks viktiga metoder

Draw ritar ut korten som finns i stacken och rektangeln.

isTopCardpressed tittar om kortet längst upp i stacken blir klickad på eller om rektangeln blir klickad på.

Abstrakta metoder är void addDeckCard(Card card) och boolean canTakeCard(Card card).

Dessa metoder är abstrakta för Decks

subklasser(Tableau, foundation, wastepile, takepile) har olika regler för att lägga till kort och ifall de kan ta kort.

Card

Card är klassen för kortobjekten. Card viktigaste attribut är ImageIcon CardImg vilket är kortets ansikte och ImageIcon downImg vilket är baksidan på kortet.

Cards viktiga metoder

Metoden boolean CardOverlap(Card othercard) vilket är en metod som tar in ett kort och tittar ifall input-kortet överlappar med kortet(this).

Metoden boolean clickCardcollsion(int clickX,int clickY är en metod för att se om kortet har blivit klickad på och clickx och clixky är musens kortinater.

Ritpanel

Ritpanel är där all interaktion med användar hanteras. Ritpanel har ganska många attribut men de viktigaste är, ArrayList <Foundation> Foundationlist , Wastepile wastepile, Takepile takepile, ArrayList <Tableau> Tableaulist , <Card> CardList vilket är alla de olika korthögarna. Card currentcard vilket är kortet som just nu används , Deck oldDeck högen som tidigare blev klickad på. Ritpanel ärver av Jpanel.

Ritpanels metoder

Mouseclicked hanterar kort överflyttningar mellan takepile och wastepil. Detta ångrar jag lite i efterhand då mousepressed och mouserleased aktiveras när mouseclicked aktiveras.

MousePressed tittar om något topkort från takepile eller något topkort från tableauerna har blivit nedtryckt och sätter currentcard till det.

MouseRelease tittar på ifall någon foundation eller tableau har fått ett kort droppat på sig och ifall de kan ta emot det. Ifall det inte stämmer så flyttas kortet tillbaka till sin gamla hög.

Initgame(boolean fixed) initierar spelet genom att ladda in alla kortbilder från Cards mappen(jag ändrade namnet på alla 10:or till T så ranken för alla kort var en char), senare så flyttar den alla kort till sin startposition. Om fixed är false blandas korten innan de delas ut.

Metoden checkWin tittar ifall storleken på alla foundationhögar är lika 13 och ändrar en global variabel(gameover) till true och ett vinstmeddelande poppar upp på skärmen.

Metoden instawin är egentligen bara en testfunktion för att kolla ifall vinstfunktion och fyller foundation högarna med respektive färg.

Solly

Solly ärver av JFrame , Solly skapar Ritpanel objektet, och knapparna JButton newBtn, fixBtn, exitBtn, instaWinBtn. newBtn reseter spelet och kallar på ritpanels metod initGame(false) , fixBtn resetar spelet och kallar på ritpanels metod initGame(true)(fixed blandning). exitBtn av slutar spelet och instaWin vinner man direkt.

Tankar kring uppgiften:

Jag borde ha tänkt mer på organiseringen och systemarkitekturen innan jag börja koda. Nu blev allt ganska stökigt och ritpanel får hantera väldigt mycket.