

# Implementation av ett UNO-spel för flera användare

DAT055

Christina Meisoll - chrmeis@student.chalmers.se

Ludvig Andersson - ludvan@student.chalmers.se

Liam Mattsson - liammattsson@hotmail.com

Dag Brynildsen Tholander - Dagbrynildsentholander@gmail.com

Ebba Håkansson - ebbaha@student.chalmers.se

6 mars 2022

## Sammanfattning

Syftet med projektet var att skapa ett UNO-spel som kan spelas i realtid av användare på olika enheter. Den första delen av rapporten är till för den som vill spela spelet. Där finns en genomgång av spelets regler och poängsystem, en spelmanual och en guide för att installera och köra spelet. Efter det följer information om implementationen av spelet. Där finns UML-diagram för spelet och klassbeskrivningar för vad varje klass gör. Även en beskrivning av det valda designmönstret och hur det applicerades på applikationen finns.

Resultatet av projektet blev ett UNO-spel som kan spelas på distans. Spelet innehåller alla funktioner som finns i fysiskt UNO, och det går att spela på olika enheter via nätverkskommunikation. Hela syftet uppnåddes. Utöver det har spelet ett väl genomtänkt och bra gjort användargränssnitt.

# Användarmanual

Det här programmet är en implementation av det klassiska kortspelet UNO. Applikationen gör att UNO kan spelas digital istället för att använda en fysisk kortlek. Programmet körs på en gemensam server som spelare sedan kan ansluta sig till från olika klienter. Det gör att användarna kan spela med varandra på distans.

Under Covid-19 har alla fått lära sig att hålla avstånd från familj och vänner. I såna situationer kan det vara trevligt att spela ett spel tillsammans, på distans. UNO passar perfekt för detta då det är lätt att lära sig reglerna och passar de flesta åldrarna.

Spelet går i korthet ut på att en spelare under sin tur ska lägga ut ett kort, enligt reglerna, och sedan går turen vidare. Den som först blir av med sina kort vinner spelet. Vinnaren får ett antal poäng som baseras på de andra deltagarnas kort, så att den kan mäta sitt eget highscore.

## Spelregler

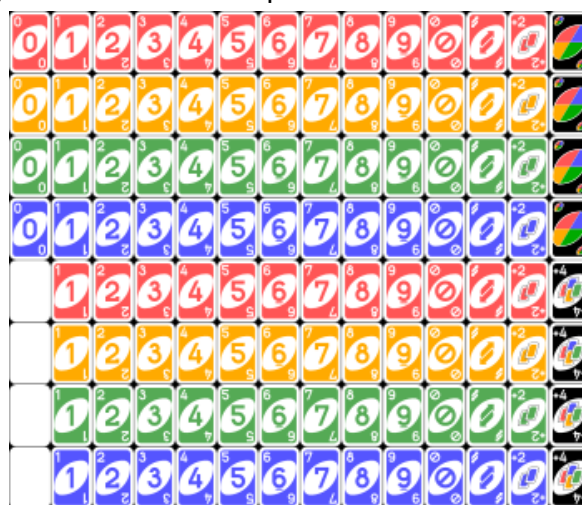
Reglerna för UNO varierar beroende på vem man frågar, men denna implementationen följer för det mesta de officiella spelreglerna.

UNO spelas med en speciell kortlek som har 108 kort vilka visas i figur 1. Grundfärgerna röd, gul, grön och blå har vardera 19 kort med valör 0-9. De har även 6 specialkort var, 2 av varje sort.

- Byt riktning-kortet med två pilar gör att spelet byter håll.
- Stopp-kortet gör att nästa spelares tur hoppas över.
- +2-kortet innebär att nästa spelare tar upp två kort och får inte spela ut den turen.

Kortleken har också två sorters svarta kort med fyra av varje sort.

- Byt färg-kortet innebär att spelaren som lägger det får välja vad nästa kort som spelas ut ska ha för färg.
- +4-kortet har samma funktion som byt färg-kortet men gör också att nästa spelare tvingas ta upp fyra kort och får inte spela ut kort den turen.



Figur 1

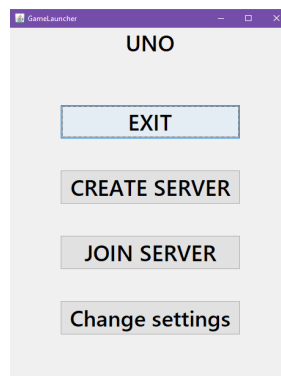
Spelaren får lägga ut ett kort med samma valör eller samma färg som det senaste lagda kortet. Det går också att lägga ett svart kort på alla andra sorters kort.

Då en spelare har ett kort kvar måste den säga UNO. I denna applikationen görs det genom att trycka på en knapp. Skulle spelaren glömma det får den ta upp tre kort i nästa runda. När någon fått slut kort vinner den personen. När spelet avslutas räknas poängen. Kort med siffror 0-9 ger lika många poäng som kortets siffra. Stopp, byt riktning och +2 räknas som 20 poäng var och båda sorters svarta kort räknas som 50 poäng var. Vinnaren får poängen för samtliga motståndarens kort.

## Spelmanual

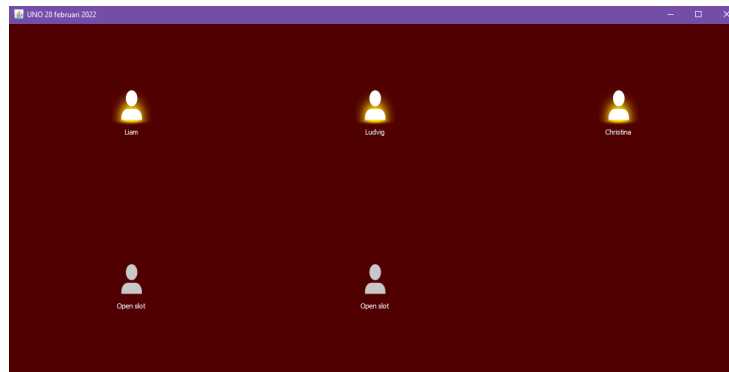
Då programmet startas öppnas en startmeny som kan ses i figur 2, där användaren kan välja mellan fyra olika alternativ.

- "EXIT" avslutar applikationen och alla servrar och spel som är öppnade på enheten.
- "Change settings" öppnar ett fönster där användaren kan skriva in sitt användarnamn och antalet spelare den vill ska delta i spel som sker på servrar skapade av användaren.
- "STARTA SERVER" skapar en UNO-server på enheten och väljer hur många deltagare ett spel på den servern ska ha. Användaren kan välja att ha det förutbestämda antalet spelare eller ändra det.
- "JOIN SERVER" gör att användaren går med som spelare i en server. Detta kan vara en server skapad av användaren själv eller av en annan spelare. Spelaren får välja mellan att ha sitt förutbestämda användarnamn eller välja ett nytt.



Figur 2

För att starta ett spel krävs att en användare startar en server. När spelare sedan går med i server flyttas de först till en lobby, se figur 3, i väntan på att rätt antal spelare ska gå med i servern(figur). I lobbyn syns användarnamnen för de andra spelarna. Då servern är full startar spelet och varje användare får ett spelfönster där de spelar UNO-spelet.(figur)



Figur 3

Spelfönstret syns i figur 4. Det visar de andra spelarnas användarnamn och indikerar vems tur det är att spela. Det visas också antalet kort som varje spelare har på handen. Bredvid det finns en knapp för att säga UNO. Under det visas det senast lagda kortet och en nedvänd korthög. Korthögen kan i vissa situationer klickas på för att ta upp kort därifrån. Längst ner i fönstret visas de kort spelaren har på handen och dessa går att klicka på för att spela ut. Om spelaren väljer ett kort som inte går att spela hörs ett varningsljud.



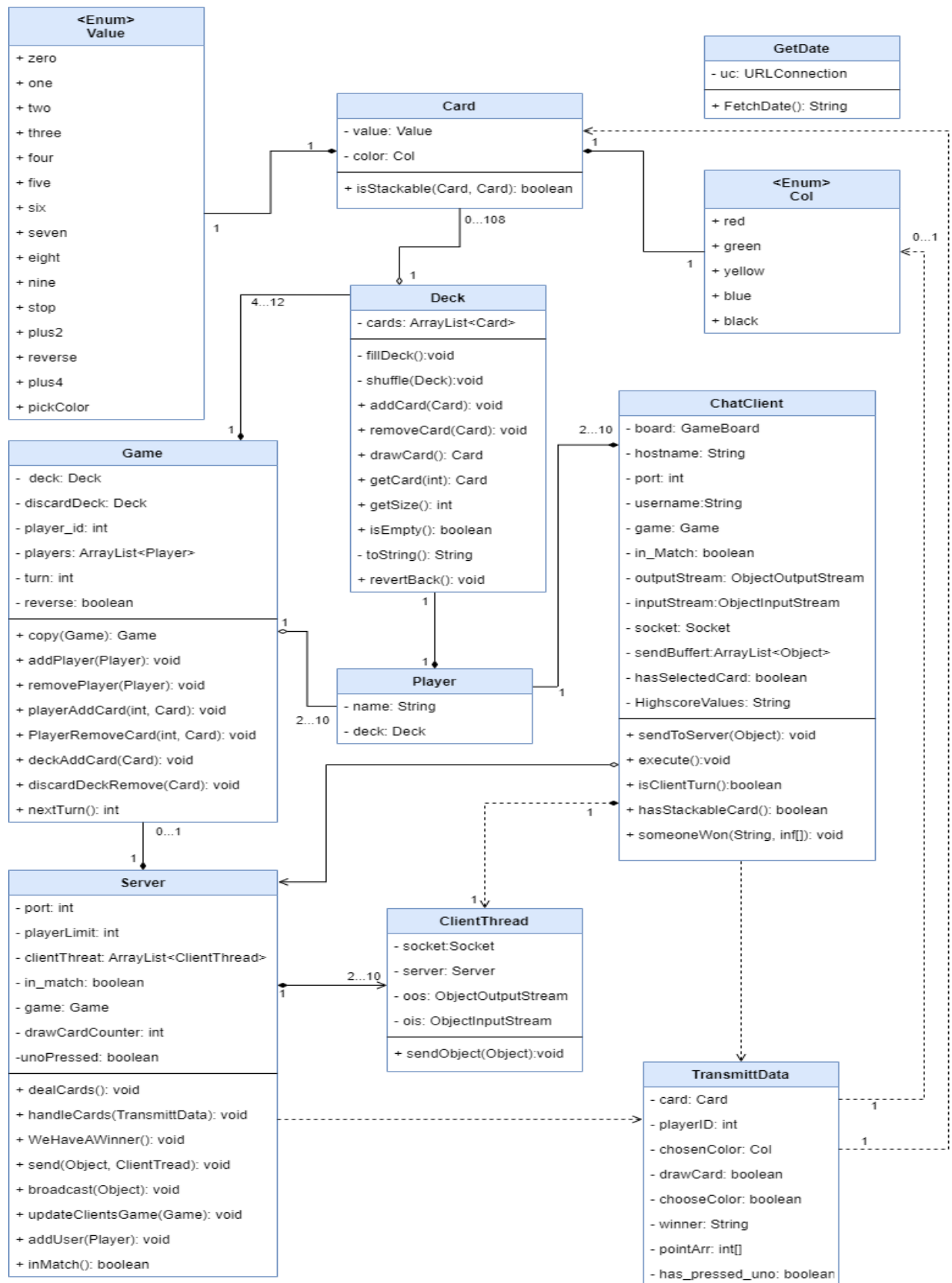
Figur 4

Turen går mellan de olika spelarna enligt reglerna tills någon har vunnit. När det händer visas ett fönster med poängen för alla som deltog i matchen. Då det fönstret stängs visas ett nytt fönster med antalet vunna matcher lokalt. När det fönstret stängs är spelet över och spelaren är åter på startmenyn.

## Installationsguide

För att köra programmet behövs filen UnoLauncher.jar som finns att hämta från [https://github.com/ludvan/Project\\_DAT055/blob/main/UnoLauncher.jar](https://github.com/ludvan/Project_DAT055/blob/main/UnoLauncher.jar). Kör den filen för att starta programmet.

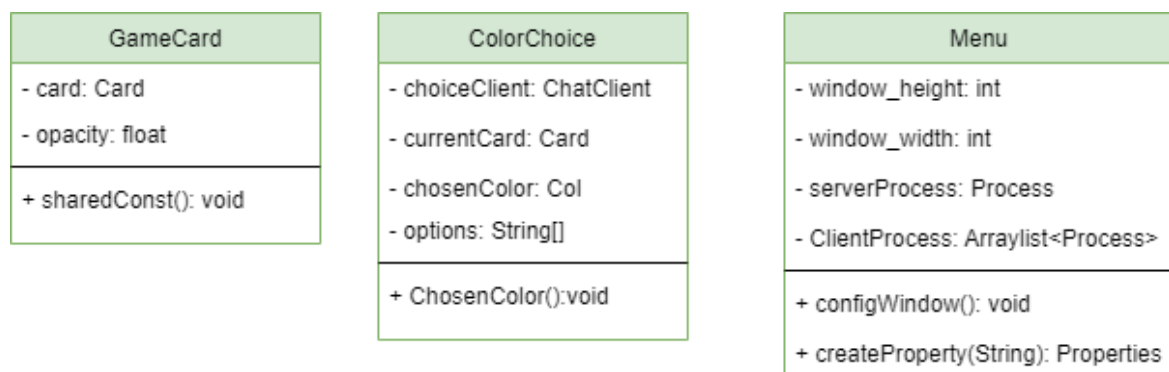
## Dokumentation av modellen



Figur 5: UML diagram för Paketet Model

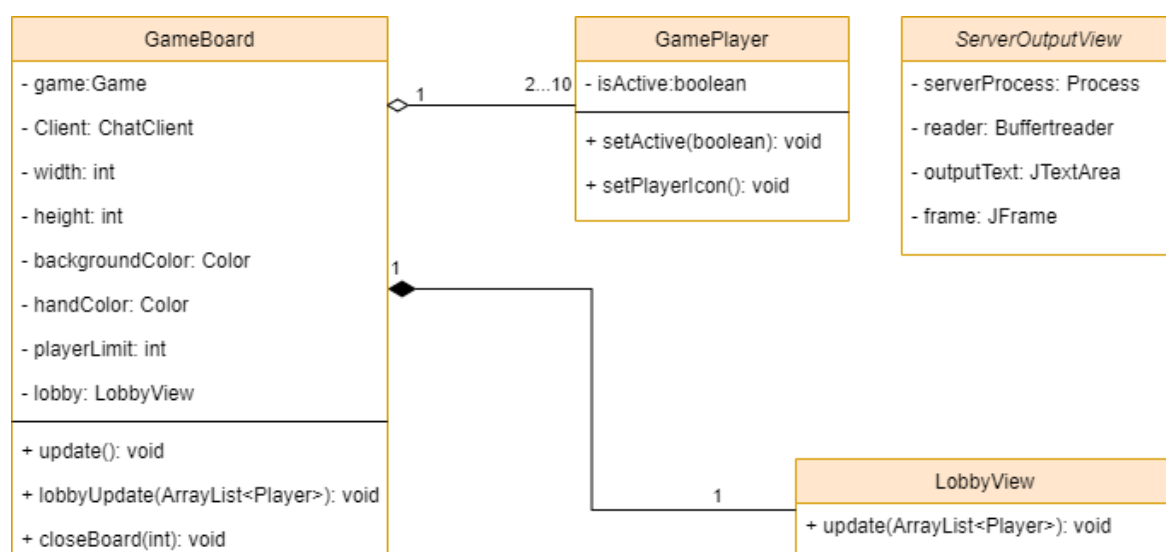
Uno Spelet består av kort som i sin tur karaktäriseras av ett värde och en färg. korten bildar sedan ett korthög. Korthögen kan vara antingen bordshögen för att dra eller lägga kort eller handkortet för spelaren. Korthögen och spelaren tillsammans ingår i spelet. Spelet själv hostas av en server. Spelaren i sin tur har var sin ChatClient, som har Samma Server som själva spelet. Spelet trådas till varje Spelare med hjälp av ClientThread. Transmittdata är vehikeln, som kan användas av både Servern och ChatClienten för att skicka data dem emellan och uppdatera spelet.

Menuklassen startas med programmet. Den tar in användarens configurationer som antal spelare och username. GameCard hanterar handkortens synlighet, som skiftar med att det är spelarens tur eller ej. ColorChoice finns till för att hantera situationer när en spelare får välja färg och det valet ska visualiseras på gamebord.



Figur 6: UML för Paketet Controller

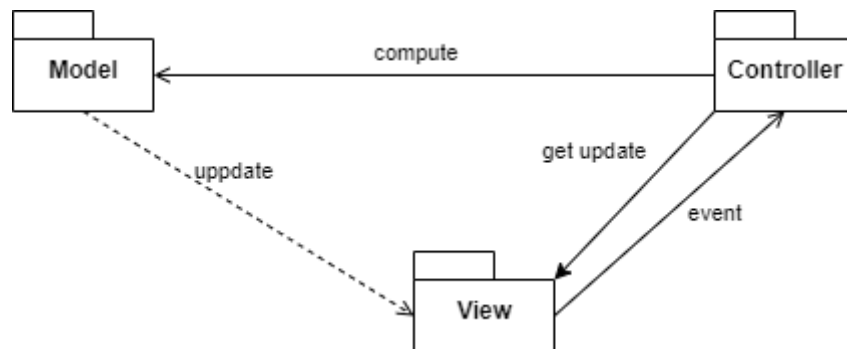
När ett spel är igång så finns ett fönster Som visar utskrifterna från Servern, som på ett verbalt sätt information över spelets tillstånd. Spelbordet visar en visuell representation av spelet. Den är individuell för varje spelare och ändras beroende om GamePlayer är aktiv eller ej. Innan spelbordet är fullt, så placeras spelare i en tillhörande lobby under väntetiden.



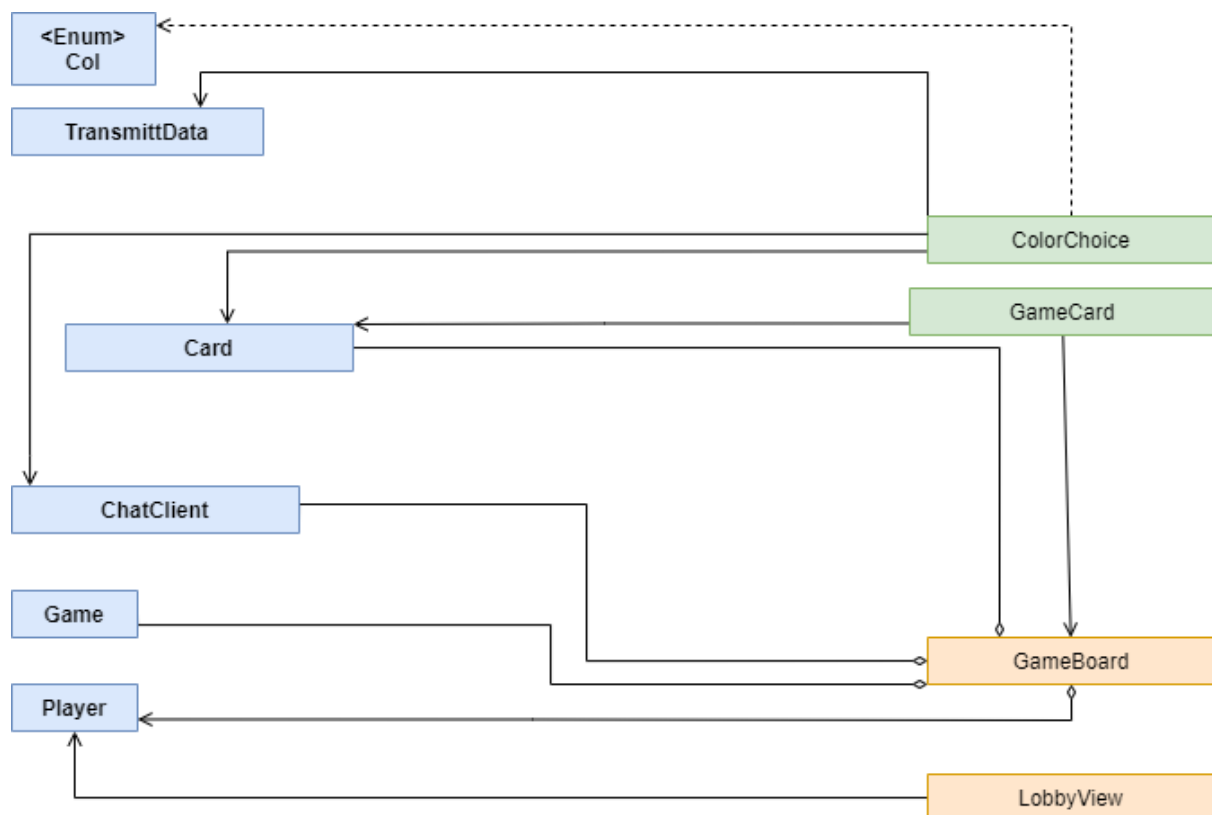
Figur 7:Model för View

## Modellbeskrivning

Programmet är strukturerat enligt MVC-modellen. Det innebär att klasserna är indelade i tre grupper, model, view, och controller. Klasserna i model kategorin sköter applikationens logik och datahantering, view-klasserna innehåller allt användargränssnitt, och controller-klasserna tar hand om händelser i programmet.



Figur 8: MVC Model



Figur 9: vår implementation



I UNO-applikationen är view-klasserna de klasser som sköter det användaren ser och interagerar med. Där finns klasserna som används för att rita upp spelbrädet och lobbyn. Det finns även en klass för att visa serverns utskrifter.

De klasser som tillhör model-kategorin är de som sköter all spellogik och nätverkskommunikation. Till kategorin hör de klasser som bygger upp kortleken. Det innefattar klasser för kort, kortfärg, kortvärde, och kortlek. Klassen som innehåller själva spelet och klassen för en spelare finns också där. En viktig del av projektet är kommunikationen, och kod för det finns i olika klasser inom model-kategorin. Det finns klass för klient, server och för klient trådar som körs hos servern. Klassen för server innehåller också mycket av spellogiken.

Den största controller-klassen i spelet är klassen för huvudmenyn. Den skapar den grafiska aspekten av menyn, men framför allt utför den handlingarna som användaren väljer att göra. Det är meny-klassen som skapar servrar och klienter vilket gör att spelet kan starta. Även klassen för att visa ett kort tillhör controller-kategorin. Den reagerar och ändrar utseendet på kortet då användaren interagerar med det. Klassen för att ge användaren möjlighet att välja kortfärg finns också här.

## Klassbeskrivningar

### **Card:**

Innehåller färg och värde för ett UNO-kort.

### **ChatClient:**

Låter en klient ta emot och skicka data via ClientThread.

### **ClientThread:**

Tråd som hanterar kommunikation mellan server och klient.

### **Col:**

Enum för tillåtna färger på kort.

### **Deck:**

Hanterar en samling kort. Används för spelhög, draghög, och spelares hand.

### **Game:**

Innehåller information om spelets tillstånd, dvs en lista på spelare, en spelhög, en draghög samt vems tur det är.

### **GetDate:**

Hämtar dagens datum i sverige.

### **Player:**

Innehåller spelarens namn och kortlek

### **Server:**

Startar ett spel, hanterar data som spelare skickar in, och skickar vidare data till andra spelare via ClientThread.

### **TransmitData:**

Används för att skicka olika sorters data i samma paket.

### **Value:**

Enum för tillåtna värden på kort.

**ColorChoice:**

Öppnar ett fönster för att välja färg när spelare har spelat ett svart kort.

**GameCard:**

En knapp i form av ett kort som spelaren kan interagera med.

**Menu:**

Startmeny för att skapa server, gå med i en server, ändra inställningar.

**Gameboard:**

Visar spelplanen/lobby för spelaren och tillåter spelaren att spela/dra kort.

**GamePlayer:**

Statusikon + namn för spelare i matchen.

**LobbyView:**

Visar vilka spelare som är i en lobby innan matchstart samt hur många platser det finns kvar.

**ServerOutputView:**

Ger information om serverns tillstånd till den som startat servern.