

# Programmering i C/C++

## Räkneövning 5

# Uppgift 1

- Spara ro5skelett.tar.gz på ditt hemområde och packa upp paketet. I konsolen görs detta med kommandot

```
tar -zxvf ro5skelett.tar.gz
```

om du befinner dig i den mapp som själva tarball arkivet. I detta fall kommer arkivet att packas upp i denna mapp.

- Kompilera sedan programmet med make och dig inte om alla varningar som dyker upp i konsolen

```
t.ex. $ warning: 'First' is deprecated
```

- Kör tillslut programmet dragimag med det välkända kommandot:  
\$ ./dragimag

# Uppgift 2

- Omforma konstruktorn för DragShape att internt lagra en bitkarta-representation. Du ska alltså modifiera konstruktorn

```
DragShape::DragShape(const wxBitmap& bitmap){}
```

så att den istället lagrar en bitakarta som representerar ett spelkort utgående från färg och värde. Ett exempel är att du tar in färg och värde som inputparametrar till konstruktorn

```
DragShape::DragShape(int suit, int value){}
```

- För att skapa en bitkarta använder du dig av **wxBitmap**, t.ex. **wxBitmap myBitmap;**
- För att sedan kunna rita grafik till denna bör du först konstruera en s.k. memory device context med **wxMemoryDC** ( t.ex. **wxMemoryDC dc;** ) och sedan använda dess medlemsfunktion **SelectObject** med **myBitmap** som parameter t.ex. **dc.SelectObject(myBitmap);**

# Uppgift 2

- Andra medlemsfunktioner för wxMemoryDC som kan behövas är bl.a.
  - `GetFont()`;
  - `SetPointSize()`;
  - `SetWeight()`;
  - `DrawRectangle()`;
  - `SetTextForeground()`;
  - `DrawText()`;
- Även wxString är till nytta när ni ska skriva ut strängar. Kom ihåg att "wrappa" din sträng med wxT eller \_T, t.ex.

```
wxString cardRank(_T("HEARTS"));
```
- När ni skrivit all den text du vill ha på ditt kort ska du spara det som finns i device context (dvs ditt spelkort) som en bild i .png format, t.ex.:

```
wxString card(_T("card.png"));  
myBitmap.SaveFile(card, wxBITMAP_TYPE_PNG);
```
- Töm sedan minnet i wxMemoryDC

# Uppgift 2

- För att rita ut en grafisk representation av kortet behöver ni initialisera en wxImage:

```
wxImage image;
```

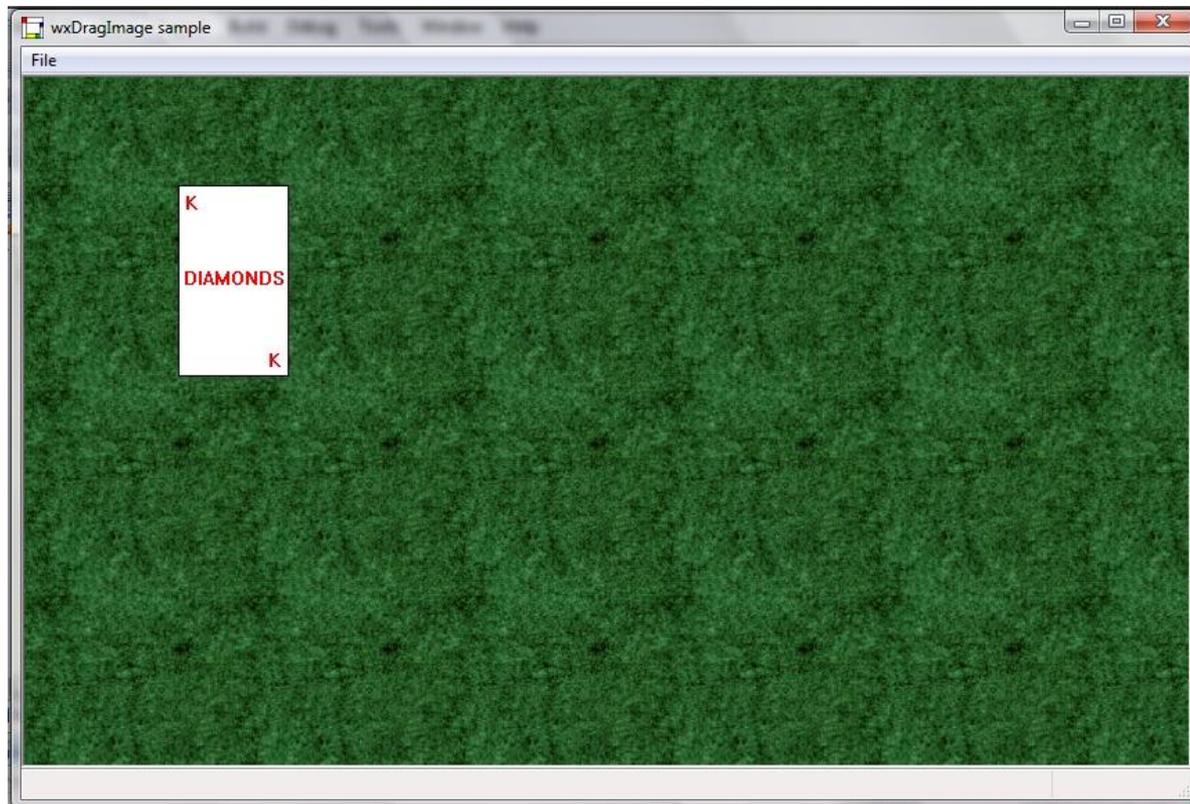
och sedan ladda det sparade kortet till image: `image.LoadFile()`. Till slut uppdaterar du `m_bitmap` på följande sätt:

```
m_bitmap = wxBitmap(image);
```

- Det är sedan i `myApp::OnInit()` du skapar ditt nya kortobjekt, bestämmer dess position på duken, fastställer "dragningsmetod" samt lägger till den till ramen (frame).
- I övrigt så får du många ledtrådar från `kortbitmap.cpp` som finns med i paketet `ro5skelett`. Du kan också få en massa goda idéer från denna sida: <http://books.google.fi/books?id=CyMsvtgnq0QC>

# Uppgift 2

BILD 1: Så här borde spelet se ut efter uppgift 2



# Uppgift 3

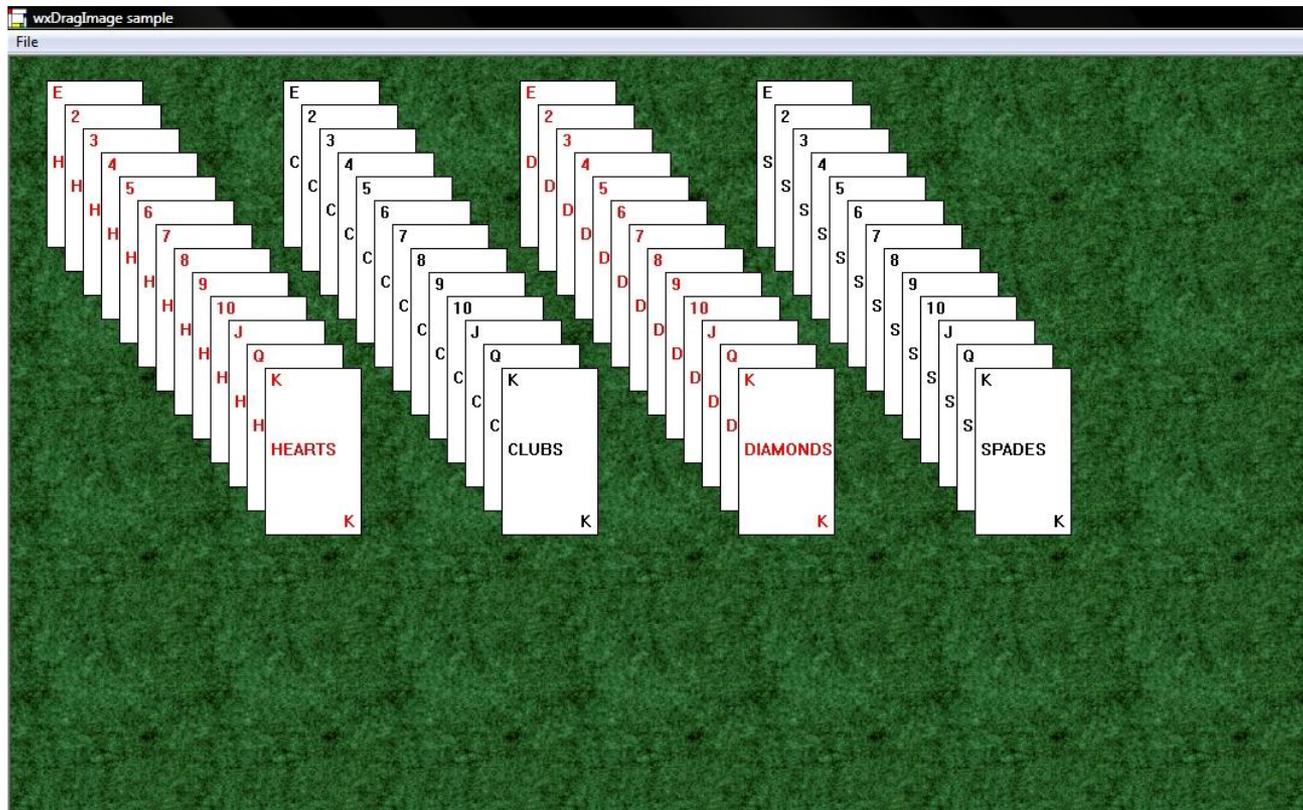
- Denna uppgift bygger långt på att lägga till den länkade lista ni skapade i tidigare räkneövningar (rö4) i detta program.
- Kopiera alltså in din kod (för bl.a. Pile och Node) i drawimag.cpp och modifiera koden för att skapa en kortpacke samt en hand.
- De två Pile-objekten, kortpacke och hand, ska sedan initieras i **MyFrame** enligt följande:

```
kortpacke = new Pile();  
hand = new Pile();
```

- För att anropa kortlekens diverse funktioner (låt oss säga att kortleken har medlemsfunktionen `addCard()`) är det då tänkt att du i **MyApp** skriver: `frame->kortpacke->addCard()` ;
- Det kan också vara värt att poängtera att noderna, i detta fall, skall bestå av instanser av `DragShape` och inte av `Card` (förutsatt att du inte skriver om hela programstrukturen i drawimag.cpp).
- I denna uppgift räcker det att du ritar ut hela kortleken (handen behöver alltså inte ritas ut på duken).

# Uppgift 3

BILD 2: Exempel på en oblandad (dvs.en initierad) kortpacke



# Uppgift 4 & 5

- Börja med att blanda kortpacken innan 5 kort delas ut till handen. Blandningsfunktion kan exempelvis anropas i MyFrame efter att kortpacken initierats. T.ex.

```
kortpacke->shuffle();
```

- Modifiera sedan om MyCanvas så att klassen ritar ut korten i handen. Du kommer här att behöva initialisera en **wxClientDC**. Den skall skapas om applicationen skall rita något på fönstrets klient-yta utanför en **OnPaint** event, dvs

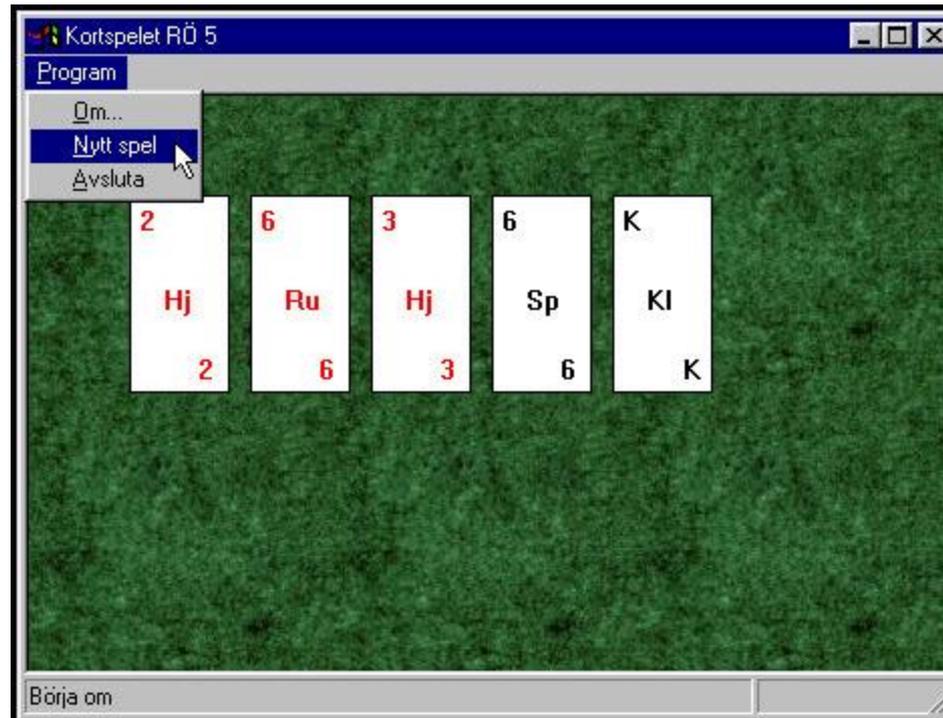
```
void MyCanvas::OnPaint(wxPaintEvent &WXUNUSED(event)) { ... }
```

- Sedan behövs ju då en funktion som delar ut korten till handen, t.ex. dealCards():

```
frame->hand->dealCard();
```

# Uppgift 4

BILD 3: Exempel efter att korten delats ut till handen.  
I din lösning ska även kortpacken visas på bordet.



# Räkneövning 5

- Här var några tips som förhoppningsvis hjälper dig på traven. Om du fortfarande har något du undrar över eller har problem med så är det fritt fram att skicka ett mail åt mig.
- LYCKA TILL MED DENNA RÄKNEÖVNING!
- **OBS!** Kom ihåg att deadline för denna uppgift är **den 22 december**. Men är det så att du av någon anledning inte hinner få rö 5 klar inom utsatt tid bör du kontakta mig i god tid **FÖRE** utgången av deadline!!