**Programvaruteknik – utveckling och underhåll, 15 och VT14**

Intrimning av nytt API och demoutveckling för pluggbara brädspel

av

Mattias Didriksson, Anders Blomkvist och Cecilia Tyboni

Akademin för teknik och miljö

Högskolan i Gävle

S-801 76 Gävle, Sweden

Innehåll

[Inledning 1](#_Toc385366489)

[Metod 2](#_Toc385366490)

[Resultat 3](#_Toc385366491)

[Diskussion 3](#_Toc385366492)

[Referenser 5](#_Toc385366493)

[Bilagor 6](#_Toc385366494)

# Inledning

Syftet med detta projekt är att skapa brädspel med hjälp av ett skrivet API. Gameboard-API kan under arbetets gång förändras. De två brädspel som i detta projekt ska tas fram är ”Fia med knuff” och ”Solitär”.

Fia med knuff är ett brädspel där antalet spelare kan variera mellan två till fyra spelare. Genom att slå en tärning bestäms antal steg en spelare får förflytta sig åt gången runt spelplanenen som visas i bild 1.

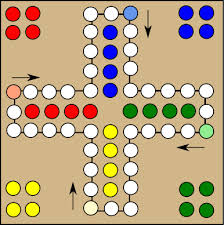


Bild 1 Spelplan för Fia med knuff

Varje spelare har fyra spelpjäser i sitt bo vid start. För att vinna måste en spelare flytta alla fyra spelpjäser ett varv runt spelplanen och därefter gå in till mål med dem. Om en spelare hamnar på samma ruta som en annan spelare redan står på kommer den andra spelarens spelpjäs att knuffas bort. Den andra spelaren får flytta tillbaka spelpjäsen till sitt bo och börja om med den spelpjäsen.

Solitär är ett brädspel för en spelare. För detta spel kan spelplanen variera och de olika varianterna kan ses i bild 2 nedan. I detta projekt används den engelska varianten av solitär.

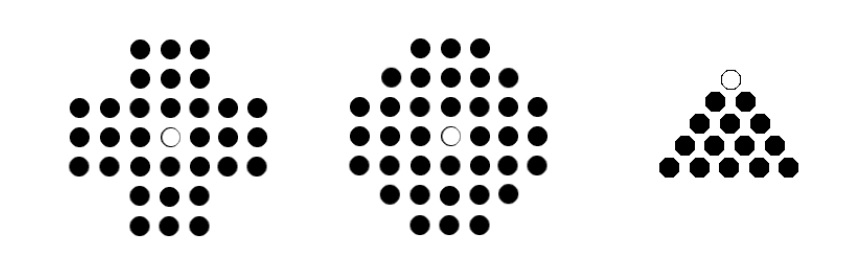


Bild 2 Olika spelplaner för Solitär, från vänster: engelsk solitär, fransk solitär och triangelsolitär

Vid spelets start är varje svart ring fylld med en kula, den vita ringen är tom. Spelet går ut på att få plocka bort så många kulor som möjligt. Det görs genom att vertikalt eller horisontellt hoppa med en kula över en annan. Den kula som man hoppar över får plockas bort från spelplanen.

I detta projekt ska minst två spelkärnor tas fram, en för Fia med knuff och en för Solitär. Ett GUI ska tas fram och detta GUI ska med mindre modifikationer vara kompatibel med andra spelkärnor.

Desingmönster som använts: Strategy, Factory

Här beskrivs bakgrunden till arbetet som har genomförts. Beroende på arbete kan det vara allt från en kort presentation till ganska så omfattande text om man har mycket bakgrundsfakta att ta hänsyn till. Finns det flera delmoment i arbetet kan det bli nödvändigt att ha underrubriker (Rubrik 2 format)

Tänk på att skapa nya stycken (som detta) i större textmassor för att få mer ”luft” i dokumentet (nya stycken skapas med ett Enter-slag). Lämpligt är att skapa nya stycken när man känner att texten delvis handlar om nya saker. Observera att det finns 2 varianter av Stycketext dels ”Stycke” som denna text (med indrag på första raden) och dels ”Stycke efter Rubrik” som inte har indrag. Stycket ovanför detta är alltså ”Stycke efter Rubrik”.

Om möjligt skall ni försöka skapa referenser för ert arbete, närhelst ni kan skall ni försöka ”hänga fast” det ni påstår, utför eller diskuterar mot externa källor. Ett enkelt exempel är att under Inledning referera till en kursboks kapitel som behandlar just det som laboration handlat om.  
Två exempel:

Laborationen handlar om att installerar Linux på någon typ av hårdvara. Linux är ett operativsystem som är baserat på öppen källkod [1]. I detta fall beskrivs en installation av Linux på en mikrodator, Raspberry Pi, som är en mycket liten billig kretskortsdator [2].

Under arbetet har vi försökt att hålla oss till de riktlinjer som beskrivs i kursboken Clean Codes kapitel 2 om meningsfulla namn [3], vilket bl.a. visar sig genom att alla namn är ganska så långa och väldigt beskrivande samt att vi inte har använt några kommentarer i källkoden.

# Metod

Vid start av projektet bestämdes vilka spel som skulle skapas. Fia med knuff och Solitär har liknande spelplaner och därmed passade bra för projektet. Arbetssättet vid detta projekt har varit att jobba parallellt med de större delarna så som GIU, konsolerna för Solitär och Fia med knuff. Konsolen för Solitär skapades först för att den inte är lika komplext som fia med knuff. Dock så kan spelplanen byggas upp på liknande sätt som fia med knuff. Detta gör då att om Solitär fungerar så kan en del av grundkoden återanvändas i Fia med knuff. De större förändringar som måste göras är reglerna för programmet. Eftersom Solitär har mindre komplicerade regler.

Under metod beskrivs hur man har genomfört arbetet. Vilka olika alternativ finns? Vad valdes? Hur genomförde man det?

I ett större arbete kan metod innehålla vilket beprövat sätt man angripit arbetet på, ex. empirisk testning, undersökningar, statistiska analyser osv.. men i ett mindre arbete, som en laboration, blir det mer en beskrivning av uppställningen man använde, hur man genomförde och vilka olika vägval man funderade på och till slut tog. Även här passar det med referenser till relevant litteratur och källor.

Tänk på att kod och hänvisningar till kod och nyckelord samt filnamn i löpande text skall vara i avvikande teckensnitt. Använd Curier New för detta, ev. i fet stil.  
Exempel:

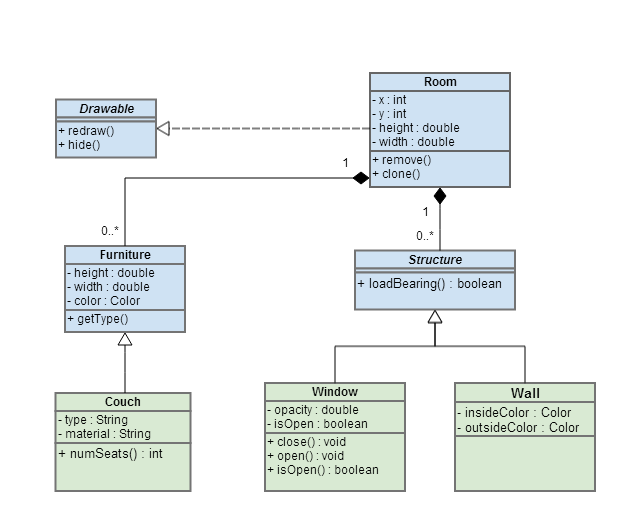
För att kontrollerar nätverksinställningarna använde vi **ipconfig /all** som ger en listning på alla satta IP adresser och MAC adresser.  
Scriptet sparades i en fil: **~home/lab1/script.sc** som senare gick att köra från BASH skalet.

# Resultat

Här kommer det som arbetet mynnar ut i. Vad har åstadkommits? Försök att undvika att lägga in subjektiva åsikter och tyckande. Beskriv och redovisa vad som producerades. I en laborationsrapport skall man här hitta ex. resultatlistningar, skärmdumpar, bilder, modelldiagram, svar på frågor, speciella kodsnutta, osv.. Naturligtvis kan vissa av dessa saker finnas i andra avsnitt, ex. under Metod som en del av genomförandet.

Kom ihåg att ge bilder ett referensnamn (Bild 1, Figur 3, Tabell 1, osv) och gärna även en mycket kort förklaring. Bilder finns med av en orsak och skall refereras till i texten.  
Ex:

Som man ser av UML-diagrammet (se Figur 1) har vi skapat en lösning som löser uppgiften på en grundläggande nivå, men p.g.a. tidsbrist inte har någon extra funktionalitet.



Figur 2. UML-diagram av vår lösning

# Diskussion

Slutligen kommer man till diskussionen där man försöker analysera sitt resultat och koppla det till ev. frågeställningar och val gjorda under Inledning och Metod. Blev det bra eller inte? Här skall man ge uttryck för sina egna personliga reflektioner på sitt arbete. Vid mindre arbeten, typ laborationsrapporter, bör man, förutom att analysera resultatet, åtminstone reflektera över ev. val man gjorde och funderade över under Inledning. I större arbeten bör man försöka värdera sitt resultat och sätta det i perspektiv samt fundera över vad hur arbetet skulle ha kunnat göras mer omfattande eller annorlunda, med tanke på de erfarenheter som skaffats under arbetet.

Exempel:

Vårt program klarar av att hantera alla grundfunktioner som ställdes i uppgiften. Vi prioriterade den enklare lösningen då vi ville vara säkra på att få en fungerande slutprodukt. Med facit i hand inser vi att vi har kunnat lägga till funktioner för felhantering, som diskuterade i inledningen, men det har då funnits en risk att vi inte hunnit färdigt då vi haft en del problem med strukturen på genomföreandet. På det hela är vi nöjda med slutresultatet men med bättre arbetsstruktur hade vi kunna nått längre på samma tid.

Här under lägger man in de referenser man ev. har gjort i den löpande texten. Använd IEEE 2006 mall för hur man gör. Observera att Word kallar detta för en Litteraturlista (se Fliken ”Referenser” under ”Infoga källhänvisning”) men det är bara att gå in och ändra namnet senare när man infogat listan. Se exemplet nedan:

Tänk på att webbsidor generellt ses som ”sämre” referenser än publicerade böcker och vetenskapliga publikationer. Sedan finns det ju bättre och sämre webbsidor. Försök att hitta officiella sidor om ni skall använda er av webbsidor. Lyckas ni lösa ett programmeringsproblem genom att använda ett forum så är ett tips att hitta ett mer officiellt referensbibliotek där ni nu troligen förstår vad som står där, eftersom ni redan har löst problemet.

# Referenser

|  |  |
| --- | --- |
| [1] | ”Linux Knowledge Base and Tutorial,” The Linux Tutorial, [Online]. Available: http://www.linux-tutorial.info/. [Använd 30 Januari 2014]. |
| [2] | ”Raspberrypi org,” [Online]. Available: http://www.raspberrypi.org/. [Använd 2 Februari 2014]. |
| [3] | R. C. Martin, "Chapter 2: Meningful names," in *Clean Code*, Boston, MA, Prentice Hall, Pearson Education, 2012, pp. 17-30. |

# Bilagor

Sist lägger man ev. bilagor. Varje bilaga skall ha ett eget namn och finnas på en egen sida. Ofta väljer man dessutom att numrera dessa så att det lätt går att referera till bilagor i texten tidigare i dokumentet. Några exempel på bilagor är programlistningar (källkod), tabeller och bilder. Tänk dock på att bilagor är att se nästan som referenslitteratur. Relevanta bilder, tabeller, kodsnuttar osv. skall läggas direkt i rapporten medan bilagor är mer av typen bakgrundsinformation som den viktiga informationen i rapporten är hämtad från. Exempelvis kan ett arbete som handlar om en speciell sorteringsalgoritm i Java ha själva metoden som sorterar inklippt under Resultat eller Metod och hela källkoden inklusive main metoden och övrig kod för att testa sorteringsmetoden som bilaga. På det sättet behöver inte läsaren plöja igenom en massa kod för att hitta det väsentliga, sorteringsmetoden. Samtidigt kan man, om det behövs, kolla upp hela koden om det uppstår frågetecken angående genomförandet. Även bilder kan man behandla på detta sätt; relevanta bilder som är direkt kopplade till rapporten för att visa något, läggs in där saken behandlas. Men är det exempel ett arbete som jobbat med många bilder för att bevisa eller visa något kan man lägga en eller två bilder i rapporter bara för att påvisa det eftersökta och sedan lägga alla andra bilder som bilaga för att läsaren skall kunna kontrollera att arbetet gjorts på många bilder med likartade effekter.