Inlämningsuppgift
Objektorienterad programmering
DA348G – VT 2012

Anders Dahlbom
anders.dahlbom@his.se
Institutionen för kommunikation och information
Högskolan i Skövde

1 Introduktion

Din uppgift är att använda objektorienterade metoder för att utveckla ett kort- eller brädspel i programmeringsspråket Java. Uppgiften är indelad i fyra steg som grovt motsvarar typiska faser som används inom mjukvaruutveckling.

- **Specifikation.** Här identifierar du det spel som du skall implementera, hur det skall fungera och vilka krav som ställs på det.
- **Prototyp.** I detta steg gör du en enkel skiss över hur ditt spel skall fungera. Denna använder du sedan för att skapa en första implementation, *quick and dirty*, av spelet.
- **Design.** I detta steg så analyserar du de krav du ställt och den prototyp du arbetat fram, för att sedan skapa en objektorienterad design för spelet.
- **Implementation.** I detta steg så implementerar du spelet enligt den design som du har tagit fram.

Målet med uppgiften är att du skall få erfarenhet i att utveckla en något större mjukvara där du använder objektorienterade metoder. För att du verkligen skall förstå principerna och fördelarna med objektorientering, så tror vi att det är viktigt att programmera mycket och brottats med detaljer och problem som olika designbeslut kan medföra. Mjukvaruutveckling är iterativt och detta innebär att både krav, design och implementation kan förändras under arbetets gång. Detta reflekteras i uppgiften genom att den innehåller två implementationsfaser som uppmuntrar till förändring.

De fyra stegen, dess leverabler samt de krav som ställs på dem beskrivs i mer detalj under rubrik 2. I slutet av kursen så kommer ditt arbete att resultera i ett fungerande spel samt en rapport som beskriver vad det är, hur det är uppbyggt och varför det är uppbyggt så.

Observera att detta är en enskild uppgift som du arbetar med efter egna förutsättningar och utefter din egen plan som du själv måste arbeta fram. Det innebär dock även att du själv skall ha skapat allt som du lämnar in. Det är tillåtet att diskutera idéer och ta hjälp av klasskamrater för att få förståelse, men i slutänden så måste allt material du lämnar in vara skapat av dig. Plagierad kod eller plagierade rapporter är inte tillåtet. Notera även att du inte får använda bildmaterial som är upphovsrättsskyddat. Om du inte vill skapa alla bilder själv så finns det t.ex. en del bildmaterial av kortlekar tillgängligt på kurshemsidan i mappen ”övrigt material/bilder”. Dessa får användas i icke-kommersiella syften.

2 Delsteg, leverabler och deadlines

Uppgiften består som nämnt av fyra steg. Varje steg har ett antal leverabler som skall lämnas in för examination vid vissa tidpunkter.

2.1 Delsteg 1 – specifikation

I denna fas så skall du identifiera och presentera det spel som du skall utveckla. De krav som ställs på det spel som väljs är att:

- det är ett kort- eller brädspel,
- spelet kräver interaktion med spelaren,
- det skall finnas ett mål med spelet samt att
- spelets utgång skall kunna påverkas av spelarens skicklighet eller erfarenhet.

Ditt val av spel bör diskuteras med din handledare innan du skickar in din specifikation. Specifikation skall presentera spelet, dess regler, vad målet med spelet är, hur användaren interagerar med spelet, samt andra krav som kan ställas på det (poängräkning, etc.). Alla egenskaper av spelet skall vara med i kravspecifikationen. Specifikationen bör innehålla figurer och text som klart och tydligt presenterar hur spelet fungerar. Om du har tillgång till en fysik version av spelet så är ett tips att ta kort på detta och använda i din beskrivning. Tänk på att inte välja ett alltför komplext spel. Det är okej att välja ett solitärspel, dvs. ett spel där en spelare spelar med sig själv utan dator- och mänskligt motstånd.

Leverabler:
- en specifikation för spelet i PDF format. Filen skall namnges med hjälp av ditt loginnamn på skolan följ av det delsteg som inlämningen gäller, exempelvis: a11xxxxx_spec.pdf.

Deadline: 2012-01-26, 23.55

Observera att du med fördel kan lämna in kapitel 1-2 av rapporten i detta steg.

2.2 Delsteg 2 – prototyp

I detta steg så skapar du en första implementation av det spel som du har valt. Det ställs inga krav på hur din lösning är uppbyggd. Initialt så bör du börja med att ta fram en skiss över hur du tänker lägga upp din lösning och sen bör du så fort så möjligt komma igång med kodandet. Det viktiga i detta steg är inte att: ha ett komplett spel, ha ren och bra kod, ha en implementation som är objektorienterad, utan syftet är istället att ha en implementation som fungerar (som kan spelas) och som kan användas som bas för att skapa en bra design för det spel som skall skapas. Du bör efter denna fas ha fått en god förståelse för de flesta detaljerna i spelet. *Quick and dirty* är ledorden i detta steg.

Leverabler:
- en körbar JAR-fil. Filen skall namnges med hjälp av ditt loginnamn på skolan följt av det delsteg som inlämningen gäller, exempelvis: a11xxxxx_prototyp.jar. Det är viktigt att du testar så att din fil är körbar innan du skickar in den,
- en zippad katalog med hela ditt projekt (källkod, bilder, m.m.). Filen skall namnges med hjälp av ditt loginnamn på skolan följt av det delsteg som inlämningen gäller, exempelvis: a11xxxxx_prototyp.zip samt
- en enkel skiss för hur din lösning planerades (PDF, JPG, …). Filen skall namnges med hjälp av ditt loginnamn på skolan följt av det delsteg som inlämningen gäller, exempelvis: a11xxxxx_prototyp.yyy.

Deadline: 2012-02-14, 23.55

2.3 Delsteg 3 – design

I detta steg så skall du skapa en objektorienterad design för ditt spel. Du använder dina erfarenheter från de första två stegen, du itererar och du skapar en ny design. Identifiera vad som var mindre bra med den första skissen och med den första implementationen och fundera på hur detta kan göras bättre. Du bör inte basera din design på den kod som du redan har skapat utan du bör börja om på nytt. Du bör separera funktionaliteten i lämpliga klasser. Som exempel kan nämnas att spellogiken och användargränssnittet ofta med fördel kan separeras. Tänk också minimalistiskt. Varje klass skall vara fokuserad och ha ett tydligt syfte. Tänk även på att minimera antalet beroenden mellan olika objekt.

De krav som ställs på din design är att den skall vara objektorienterad och att den skall använda sig av arv, polymorfism, inkapsling och gränssnitt.

I detta delsteg så finns det ett vägval för dig att göra. Om du är intresserad och om du tror att du hinner, så kan du välja att skapa en design för ditt spel så att det istället kan implementeras som en mobilapplikation för androidplattformen. Detta är inte något som krävs för ett högre betyg och det är något som kan komma att kräva en större insats från din sida. Det är dock mycket kul samt att det ger dig ytterligare erfarenheter. Diskutera med kursansvarig och återge det val du gör i din inlämning.

Leverabler:
- en objektorienterad design i UML samt en beskrivning av denna som presenterar designen och motiverar de designval som har gjorts. Beskrivningen bör förklara hur alla delar hänger ihop samt vad deras individuella syften är. Filerna skall lämnas in i PDF format och skall namnges med hjälp av ditt loginnamn på skolan följt av det delsteg som inlämningen gäller, exempelvis: a11xxxxx_design.pdf.

Deadline: 2012-02-26, 23.55

Observera att du med fördel kan lämna in kapitel 1-3 av rapporten i detta steg.

2.4 Delsteg 4 – implementation

I detta steg så börjar du om med implementationen från början där du nu använder den objektorienterade design som du tagit fram i steg 3. Ta lärdom från de fel och problem som du gjort och stött på i den första implementationen. Undvik även att återanvända kod från den första implementationen. Detta kan låta som ett slöseri, men det krävs ofta mer tid för att skriva om en applikation till att följa en ny design än att implementera den på nytt från början.

Likt i steg 3 så ställs krav på att din implementation skall vara objektorienterad och att den skall använda sig av arv, polymorfism, inkapsling och gränssnitt. Utöver detta så ställs även följande krav.

- Gränssnittet skall vara responsivt. Programmet får med andra ord inte hamna i dödlägen och det får inte vara långa svars- eller väntetider utan tydlig återkoppling till användaren.
- Varje klass skall ha ett tydligt syfte och antalet beroende bör hållas lågt.
- Metoder skall ha ett tydligt syfte och endast ett syfte och de skall hållas korta.
- Alla namn på funktioner, metoder, klasser, variabler, m.m, skall vara beskrivande och på engelska.
- Kommentarer bör finnas där det behövs, även dessa skall vara på engelska.
- Det skall gå att snabbt förstå koden genom att titta på den.

Om du i steg 3 valde att göra en androidapplikation så fullföljer du nu det valet genom att implementera din design för spelet som en androidapplikation.

Leverabler:

- en körbar JAR-fil alternativt en installerbar APK-fil (om du valt androidspåret). Filen skall namnges med hjälp av ditt loginnamn på skolan följt av det delsteg som inlämningen gäller, exempelvis: a11xxxxx_implementation.jar (.apk för android),
- en zippad katalog med hela ditt projekt (källkod, bilder, m.m.). Filen skall namnges med hjälp av ditt loginnamn på skolan följt av det delsteg som inlämningen gäller, exempelvis: a11xxxxx_implementation.zip samt
- en fullständig rapport i PDF format (denna beskrivs under rubrik 3). Filen skall namnges med hjälp av ditt loginnamn på skolan följt av ordet rapport, exempelvis: a11xxxxx_rapport.pdf.

Deadline: 2012-03-25, 23.55

3 Rapport

Dina resultat från samtliga delsteg i uppgiften skall redovisas i en rapport som skall följa den mall som finns tillgänglig på kursens hemsida. Rapporten skall lämnas in i PDF format och skall ha följande innehåll och struktur.

- Kapitel 1: introduktion
  - Här introducerar du uppgiften och det spel som skall skapas samt att du ger en överblick över hur rapporten är strukturerad.
- Kapitel 2: kravspecifikation
  - Här analyserar du spelet och presenterar alla krav som ställs på det. Du bör ta figurer till din hjälp för att presentera spelet och för att förtydliga de krav som ställs.
- Kapitel 3: design
  - Här presenterar, motiverar och förklarar du den slutliga design som du har arbetat fram. Kapitlet skall innehålla klassdiagram som presenterar din lösning. Andra typer av diagram är också tillåtet (flödesscheman, m.m.). Det är viktigt att du presenterar och motiverar hur alla komponenter hänger ihop samt syftet med varje enskild komponent. Diskutera även algoritmer och algoritmval där sådant är relevant.
- Kapitel 4: körexempel och analys
  - Här presenterar du din färdiga produkt och analyserar den utifrån de krav som ställts och den design som du använt. Du bör bland annat diskutera frågor som: har alla kraven uppfyllts, var designen lämplig, vad var mindre bra med din lösning, vad var riktigt bra med din lösning och vad kunde/borde ha gjorts annorlunda och i så fall hur?
- Kapitel 5: slutsats
  - Här sammanfattar du rapporten (uppgiften, spelet, lösningen) och de erfarenheter som du fått när du arbetat med uppgiften.

4 Inlämningar

Alla inlämningar sker under uppgifter på kurshemsidan i SCIO. Observera att alla deadlines och krav är hårda. Om du t.ex. lämnar in en JAR fil som inte är exekverbar så får du underkänt.

5 Examinationskriterier

Inlämningsuppgiften examinerar följande kursmål:

- kunna använda objektorienterade designprinciper och hjälpmedel,
- självständigt identifiera, formulera och lösa programmeringsproblem,
- tolka och skapa objektorienterade program, använda tillgängliga klassbibliotek, använda konstruktioner för händelsebaserade program och använda konstruktioner för flertrådade program, i programmeringsspråket Java samt
- skriftligt presentera och redogöra för krav och lösning på ett programmeringsproblem med ett korrekt och sakligt språk.

För att få godkänt på uppgiften så krävs det att samtliga kursmål bedöms som godkända.

Uppgiften bedöms med betygen U, 3, 4 och 5. För betyget 3 så måste spelet och rapporten som lämnas in i steg 4 uppfylla ett antal grundläggande kriterier. För betygen 4 och 5 så används även att ett antal utökade kriterier med avseende på process, produkt och kod vid bedömning.

5.1 Grundläggande kriterier

För att få godkänt så måste ett fungerande kort- eller brädspel ha lämnats in där:

- det krävs interaktion med spelaren,
- det skall finnas ett mål med spelet samt att
- spelets utgång skall kunna påverkas av spelarens skicklighet eller erfarenhet.

Design och implementation måste:

- vara objektorienterad samt
- använda sig av arv, polymorfism, inkapsling och gränssnitt.

Rapporten måste:

- tydligt presentera krav på ett programmeringsproblem,
- innehålla klassdiagram i UML samt beskrivningar av samtliga sådana,
- diskutera och motivera designval,
- presentera och analysera den lösning som skapats,
- innehålla en röd tråd som kopplar ihop problem, krav, design och lösning samt
- vara, i stora drag, fri från stavnings- och meningsbyggnadsfel och skall följa föreskrifterna i studieverkstans *Enkla skrivregler* som finns att läsa på följande adress*:*
  http://www.his.se/student/stod-i-dina-studier/studieverkstan/sprakstod/enkla-regler/

5.2 Utökade kriterier

Uppgiften har ett stort fokus på den process som genomgås. Om du är delaktig i handledningar, lämnar in samtliga deluppgifter i tid, om du reflekterar över och diskuterar de problem du stöter på och de val du gör och om du drar lärdom av detta, så kan detta kompensera för brister i den slutgiltiga produkten och den kod som den är uppbyggd av. Omvänt så kräver ett lågt resultat på processkriteriet mer av den slutliga produkten och koden.

Processkriterier

- Analys och reflektion av resultat.
- Självständighet i utvecklingen.

- Förmågan att möta deadlines.
- Förbättringar mellan versioner och insikt gällande möjliga sådana.

Produktkriterier

- Funktionalitet, kvalitet och komplexitet hos spelet.
- Kvaliten på implementationen (stabilitet, korrekthet, kompletthet, buggfrihet).
- Återkoppling mot kravspecifikationen.

Kod- och designkriterier

- Användandet av objektorientering.
- Användandet av Javas klassbibliotek (alternativt androids motsvarigheter).
- Koden är läsbar, lättförståelig och välstrukturerad.
- Sund namngivning.
- Fokus, syfte och beroenden av och mellan klasser och metoder.
- Kommentering.

6 Omexamination

Om du inte får godkänt på den slutliga inlämningen så kommer du att erbjudas ytterligare ett inlämningstillfälle. De leverabler som skall lämnas in då är de samma som i steg 4.

Om du missar eller får underkänt på ett delsteg (1-3) så kan du fortfarande fortsätta till nästa steg. Detta betyder att det är möjligt att ignorera steg 1-3 och endast lämna in det som krävs i steg 4. Observera dock att detta ställer höga krav på produkten och koden eftersom processen har ignorerats.