DA361A HT 2019

Laboration

OOAD & UML

Inför labben

Läs igenom fallen nedan innan du kommer till labben. Du behöver inte ha detaljkoll på dem, men det är bra om du i alla fall hunnit bekanta dig med dem innan. Ta också med dig papper och penna – på den här labben behöver du ingen dator.

Instruktioner

Innan vi startar laborationen kommer vi att köra ett upprop. Vi samlas igen kvart över elva och går igenom våra lösningar. Även här tas närvaro. Du räknas som närvarande om du är med under båda uppropen och visat dig aktiv under sessionen.

Ni arbetar i grupper om mellan tre och fem personer. För att inte fastna för länge med varje fall finns en rekommendation för hur mycket tid som bör läggas på varje fall.

Fokusera på att rita för hand under laborationen. Det går i allmänhet snabbare att skissa lösningar på papper eller whiteboard än att göra detta digitalt. På laborationen är det viktigare att ni snabbt kan skissa upp lösningar och jämföra, ändra, gå tillbaka och så vidare än att linjerna är perfekta. Om ni blir klara med era uppgifter får ni så klart lov att bygga diagrammen med ett digitalt UML-verktyg.

Spara det arbete ni gjort på laborationerna. Frågor av den här typen kommer att dyka upp på tentan, och att kunna gå tillbaka och titta på ert tidigare arbete kan hjälpa er att plugga inför den.

Fallbeskrivningar

Fall 1: AI-spelaren (30 minuter)

Rita ett tillståndsdiagram som visar hur en AI-spelare i ett spel byter strategi beroende på vad som finns i omgivningen. Reglerna för AI-spelaren är enligt nedan:

- On guard: När ingen fiende finns inom synhåll.
- Fight: Om det finns en liten fiende inom synhåll så attackeras denna.
- Run away: Används om IA-spelaren ser en stor fiende eller om AI-spelaren håller på att förlora när en fiende attackerats eller AI-spelaren attackeras.

Vilka händelser påverkar spelaren? Vilka tillstånd kan spelaren befinna sig i?

Fall 2: Niagara (75 minuter)

Vi vill göra en modell av Niagara. Utifrån nedanstående text, identifiera de substantiv som borde utgöra baser för klasser och lägg till relevanta attribut och metoder. Glöm inte relationerna!

> *Niagara är en byggnad som nyttjas av Malmö universitet. Byggnaden är byggd 1998 och ligger på tomterna Niagara 2 och Niagara 3. Den består av tre huskroppar, benämnda A, B och C. Dessa är 7, 8, respektive 13 våningar höga. Varje huskropp servas av två hissar. Hissarna (och dörrarna till byggnaden) är låsta mellan 16.30 och 7.30. Personer med behörighet identifieras med ett ID-kort och kod, som slås in på en koddosa. Hissarna i A-huset är reserverade för personal och kräver alltid identifiering.*
>
> *Under huset finns ett källarplan med ett parkeringshus, vilket drivs av Q-park. Här finns även ett omklädningsrum med dusch och låsbara skåp. Dessa skåp kräver hänglås.*
>
> *På entréplan finns tre hörsalar (benämnda B1, B2 och C) och reception med tillhörande vaktmästeri. Här finns även Restaurang Niagara, studentkåren, Drivhuset och Campusbokhandeln.*
>
> *Plan två domineras av ett café, ett studentpentry och ytor för möten.*
>
> *På plan tre till blandas student- och personalutrymmen. Datorsalarna finns i A-huset på våning tre och fyra. A-huset är i övrigt främst avsett för undervisnings- och seminarieverksamhet. På våning fem och sex huserar institutionerna för teknik och samhälle och kultur och samhälle, deras administration, samt några specialytor för forsknings- och skapandeverksamhet, exempelvis IOTAP-labbet där de eminenta herrarna Daniel Spikol och Johan Holmberg ofta sitter och arbetar.*
>
> *Plan sju och uppåt upplåts helt åt personalutrymmen. Här finns även universitetskansliet där rektor och hennes stab arbetar.*
>
> *Taket har en takterass med gräsmatta, bänkar och träd. Här finns även en större ventilationstrumma Vid bra väder är utsikten bra och från taket kan man se Öresundsbron, Köpenhamn, Barsebäck och Lund.*
>
> *Eftersom byggnaden är en offentlig byggnad råder totalt rökningsförbud på platsen.*

Börja med att läsa texten tillsammans. Identifiera de objekt som är relevanta och förkasta de objekt som är irrelevanta. Tänk på att skriva några stödord kring varför ni gör de val ni gjort – detta är bra att kunna gå tillbaka till under kursen. Identifiera därefter attribut och beteenden (som ni modellerar som metoder) hos objekten och skriv in dessa i ert diagram. Här får ni lov att använda er av er **domänkunskap** (vilket innebär att ni utnyttjar det faktum att ni känner till byggnaden i fråga) för att lägga till relevanta attribut och beteenden om information om dessa saknas* i texten. Även här ska ni motivera era beslut. Slutligen tittar ni på hur alla objekt hänger ihop och modellerar dessa relationer. Precis som ovan får ni använda er av er domänkunskap och fylla i de luckor som ni tycker finns.

* I verkligheten hade ni tagit upp dessa luckor med en kund eller stakeholder.

Fall 3: Använd en TV (75 minuter)

Vi ska modellera tre användningsfall för en TV i var sitt sekvensdiagram. Givet nedanstående klassdiagram och beskrivningar, modellera följande:

DA361A HT 2019

Starta Tvn

En användare slår på tv:n genom att trycka på Power-knappen på en fjärrkontroll. En boundaryklass Remote tar emot signalen och gör ett asynkront anrop (powerOn) till TVController. TVController anropar då metoden startLastChannel hos sig själv. Denna metod frågar ChannelList om den senast användna kanalen. Metoden sätter därefter tunern med hjälp av kanalens frekvens. Slutligen tänds skärmen, som hämtar en bildström från tunern.

Byta kanal

En användare byter kanal på tv:n genom att trycka på en sifferknapp på en fjärrkontroll. En boundaryklass Remote tar emot signalen och gör ett asynkront anrop (choose med argumentet channel:number) till TVController. TV:n är lite korkad och kan bara byta kanal genom att iterera över hela kanallistan tills rätt kanal hittats. Kanalen identifieras med ett id-nummer, vilket ska matcha sifferknappens värde. När rätt kanal har hittats, sätts tunern med hjälp av kanalens frekvens. Slutligen tänds skärmen, som hämtar en bildström från tunern.

Slå av ljudet

En användare stänger av ljudet på TV:n genom att trycka på mute-knappen på en fjärrkontroll. En boundaryklass Remote tar emot signalen och gör ett asynkront anrop (mute) till TVController. TVController skickar ett meddelande till högtalaren, som då sätter volymen till sig själv till noll.

Använd de element som vi gått igenom på föreläsningen.