

Seminarie 2 - Space Invaders

Objektorienterad analys

1. Finn objekten (leta substantiv)
2. Klassificera objekten (namn, ansvar, samarbeten)
3. Beskriv relationer (Arv eller association)
4. Identifiera aktörer och användningsfall

Utförligare beskrivning av stegen

1. Ett förslag till hur man ska finna objekt är att läsa kravspecifikationen och notera ofta förekommande substantiv.
2. Alla objekt som identifierats i steg 1 ska klassificeras. Detta innebär att bestämma vilka klasser som objekten är instanser av. Beskriv varje klass kortfattat med:
 - klassens namn – välj namn med omsorg! Välj engelska namn som substantiv i singularis.
 - ansvar – operationer som kan utföras på eller av objekt av klassen ifråga.
 - samarbetspartners – vilka andra klasser som klassen samarbetar med för att utföra sina åtaganden.

Vilka andra objekt/klasser ett objekt behöver känna till kan upptäckas genom att ställa frågor som "varifrån får objektet denna information", "till vem ska objektet leverera denna information". Ibland kommer kunskaper att representeras av en datamedlem i objekten ifråga, ibland i form av utdata från anrop av metoder i andra objekt.

3. I detta steg bestäms de relationer som finns mellan klasserna i systemet. En relation är antingen arv (x är en specialisering av y), aggregation/komposition (x består av komponenten y) eller association (x kan använda en y).
4. Ett användningsfall är en interaktion som kan inträffa under systemets exekvering. Syftet med att identifiera aktörer och användningsfall och utföra olika scenarier för användningsfall är att få en djupare insikt om samarbetet mellan objekt. Under denna aktivitet kan man till exempel upptäcka nya aktörer och användningsfall, nya objekt/klasser, nya samarbetspartners och nya relationer mellan klasser och objekt. Omvänt gäller att varje objekt/klass/ansvar/samarbete ska ingå i något användningsfall för att dess existens ska vara motiverad.

Scenario

Det nystartade svenska spelföretaget Oskho Games (en hyllning till tidigare kursledare Oskar Holmström) vill lansera en riktig succé. Företaget vill skapa en modern version av det klassiska spelet Space Invaders. De har själva en bra idé av vad som ska vara i spelet, men har dålig koll på hur de ska implementera de olika delarna i spelet - speciellt representationen av alla objekt. Oskho Games vänder sig därför till er, 76:an Consulting, för experthjälp.

Mail från Oskho Games

Hej 76:an Consulting,

Tack för att ni kan hjälpa oss. Här är en beskrivning av vad vi vill ska vara med i spelet:

Space Invaders är ett spel med olika flygande objekt. För spelaren är det viktigaste objektet spelarskeppet. Den är det enda försvaret mot alla space invaders. Det finns endast ett spelarskepp och det är spelaren som styr det. (I framtiden ska det vara möjligt att expandera till två eller fler spelarskepp.) Spelarskeppet har en viss mängd liv, en position i spelet, kan förflytta sig och kan skjuta olika projektiler. Spelarskeppet har dessutom en spelpoäng som ökar när fiender skjuts ned.

I spelet finns det också flera olika typer av fiender. Det finns både "vanliga" fiender, som är lätta att stoppa, och bossar. Vid lanseringen av spelet kan det räcka med att det finns två stycken olika typer av vanliga fiender och en boss. Det som skiljer vanliga fiender från varandra är att de har olika mängd liv och kan skjuta olika projektiler. Detsamma gäller för bossar. Bossar har dessutom unik funktionalitet, att de kan skapa nya vanliga fiender för att lansera attacker på spelarskeppet. Vanliga fiender rör sig likadant, fast bossar kan ha ett unikt rörelsemönster för att göra dem svårare att träffa.

I spelet finns det dessutom asteroider som agerar hinder. Dessa är hinder som står stilla mellan spelarskeppet och fienderna. (Iallafall i den första versionen, i framtiden kan de börja flyga runt som riktiga asteroider.) När en asteroid har träffats av ett antal projektiler så ska den försvinna.

Det finns från början två olika projektiler i spelet: laser och atomic blaster. Om en projektil träffar spelarskeppet eller en fiende ska dess liv minska. Om spelarskeppet kolliderar med en asteroid eller ett fiendeskepp ska spelarskeppets liv också minska.

Power-Ups är en otroligt viktig del för att ge spelet variation. När spelarskeppet skjutit ned ett antal fiender kan en power-up falla ned för skärmen. Om den kolliderar med spelarskeppet så ska power-Up:en ha funktionalitet som ökar spelarskeppets liv. (I framtiden kan det finnas fler power-ups som gör andra saker än ökar liv.)

Utöver allt detta behöver det finnas två olika banor i spelet. En bana har en unik uppsättning av fiender, bossar och asteroider.

Vi på Oskho Games har hört mycket gott om er och håller tummarna att arbetet går bra!

Med vänliga hälsningar,
Oskho Games

UML

Exemplet nedan visar hur de viktigaste relationerna ritas med UML.

- **Vehicle** är en basklass med en skyddad datamedlem **color** och en *polymorf* funktion **speed**. Pilen från **Car** till **Vehicle** visar att bilen ärver(specialiserar) fordonet.
- **Car** är en klass härledd från **Vehicle** där 1 **Car** har 4 **Wheel** som privat datamedlem. Diamantpilen från **Wheel** till **Car** visar att hjulen är en komponent som bygger upp bilen och att det behövs 4st hjul för varje 1st bil. Ibland syns detta förhållande med en fylld diamant och kallas då komposition och används när komponenten aldrig existerar självständigt.
- **Driver** har en publik funktion **drive** för att köra en **Car**. Strecket från **Car** till **Driver** visar att föraren kan använda (köra) en bil.

