

INF-2101 Algoritmer

Obligatorisk oppgave 2 – You are likely to be eaten
by a Grue!

Erik Solli Teigen (ete003)

10. oktober 2012

Introduksjon

I denne oppgaven skal vi implementere en kode for å finne et objekt i en labyrint og unngå å bli spist av gruer.

Krav

I og med at vi har en gitt tid på oss, så er det ikke noe spesielt krav om at programmet skal klare å finne Toby noe det nok hadde vært om vi hadde evig tid.

Bakgrunn

Denne oppgaven er inspirert av filmen Labyrinth fra 1986. I filmen ønsker Sara broren sin Toby bort noe goblin kongen hører og tar han bort. Så angrer Sara og vil ha Toby tilbake. Goblin kongen gir henne en sjanse, hvis hun kan finne veien gjennom en labyrint til slottet på under 13 timer ellers blir Toby om til en av kongens gobliner.

Design

Designet er i grunn ganske elementært for en labyrint, enten er det gulv (tiles) eller vegger. Man må bevege seg gjennom labyrinten på «tilene» og unngå å gå i veggene. For å klare det kan man seg seg om og få tilbake info om alle retningene og det er en tile der eller vegg. Ett steg tar 3 minutter og du har som sagt fire veier å gå (nord, sør, øst og vest) og om du ikke klarer det på 13 timer (260 steg) så taper du.

Implementasjon

Da jeg begynte hadde jeg store planer og håp om at dette ikke skulle være så vanskelig å få til. Jeg fokuserte på å hele tiden vite hvor det var trygt å gå og hvor jeg hadde vært slik at jeg ikke skulle vandre i mine egne forspor. Jeg begynte med å la hele labyrinten fungere som et kart med x og y koordinater. Så lagde jeg en funksjon som brukte look() fra prekoden og den lagde ei liste og trygge koordinater. Så tok koden å sjekket rundt seg om hvilke veier som var trygge og hvilke jeg hadde vært på før og så tok et valg på hvor koden skulle gå ut ifra det. Jeg lagde og en metode som sjekket om Sara var stuck, det at alle retninger var veier hun hadde vært før eller vegger og som så kjørte en metode for å gå bakover til hun fant et punkt hvor hun kunne gå en ny vei.

Diskusjon

Tankegangen min fungerte helt fint i starten, og til syvende og sist ville den nok gjort det også. Problemet var når Sara blir stuck. Det var ikke noe problem å gå et par steg tilbake, men når det blir et større område hvor Sara hadde vært overalt så går ikke det så bra. En funksjon som kunne gå bakover i listen over hvor hun hadde vært og gådd dit og samtidig sjekket etter nye muligheter ville fungert, men jeg fant ikke helt ut hvordan jeg skulle implementert den da listen over hvor jeg hadde vært blir oppdatert samtidig som den går bakover så den blir å ende opp i en uendelig loop.

Konklusjon

Jeg føler jeg fra starten av valgte kanskje feil måte å løse oppgaven på, men da det var gjort måtte jeg kjøre på. Og jeg fikk jo til det meste for utenom en kristisk feil som nok ødelegger helhetsinntrykket. Men det var jo da ikke en uretterlig feil, men dog en feil jeg ikke fikk rettet. Jeg hadde jo et håp om å gjøre dette raskt og enkelt, men endte opp med å gjøre det hele vanskeligere enn nødvendig, noe jeg selv fikk svi for. Men alt i alt var det god og nyttig læring, på godt og vondt.