

Kapitel 1

SUD - single user dungeon

1.1 Hvad vil vi?

Da vi begge er så gamle, at vi kan huske den gang der ikke var grafik i alle spil, men at de alligevel havde et gameplay der langt overstiger nutidens first-person-shooter mener vi at der er noget at komme efter. Derfor har vi besluttet at vi vil udvikle en 'single user dungeon'. Denne minder på mange måder om de forskellige MUD 'multi-user dungeons' man igennem tiden har kunne spille på diverse netværk, men med den forskel, at der kun er en 'menneskelig' bruger.

1.2 Hvordan vil vi gøre det?

Vi forestiller os, at vi har en verden der består af en række rum, hvori man kan bevæge sig rundt. I de forskellige rum, kan der være placeret forskellige objekter som man kan tage/bruge/undersøge o.s.v.. Foruden en selv, skal der selvfølgelig også være nogen at kæmpe imod. Det vil være i form af nogle væsener der bliver styret af computeren. Men hvordan vil vi gøre det programmeringsmæssigt.

De klasser som vi vil bruge i programmet, er inddelt i to kategorier:

Model-klasser er de klasser der indeholder informationer om den verden man befinder sig i. P.t. forestiller vi os, at vi har følgende hoved-klasser **Ting, Væsen, Rum**.

Styrings-klasser er de klasser der styre spilforløbet. Vi forestiller os, at spillet skal være tidsstyret. Der er et centralt ur, der løbende annoncerer at handlinger/ændringer kan foretages. I denne kategori vil følgende klasser kunne findes **Tid, Kamp, AI**

1.3 Hvad skal man kunne

Vi forestille os, at man skal kunne foretage følgende handlinger i spillet:

Slå: Det skal være muligt at angribe andre væsner i spillet.

Bevæge sig: Fra de forskellige lokationer skal det være muligt at bevæge sig til andre lokationer der er direkte forbindelse til.

Se: Få en uddybende beskrivelse af det der ses på. Eksempelvis kunne det være det rum man befinder sig i, eller en genstand man har taget op.

Tag ting: Spillet vil være udstyret med en række ting, som eksempelvis: våben, mad, o.s.v. Det skal være muligt at tage disse ting, samt at smide dem igen.

Bruge ting: Det skal selvfølgelig også være muligt at bruge de forskellige ting man har taget i spillet. Det kunne være at spise den mad man har fundet, eller tage et våben i hånde for at forbedre sine chancer for at ramme en modstander.

NOP: At foretage intet i dette træk.

Da SUD er et trækbaseret spil, er det vigtigt at man har defineret hvornår det er tilladt at foretage nye træk, samt hvornår et træk er færdigt. Vi ser to oplagte løsninger (selvom der sikkert findes mange flere). Den første er turbaseret, hvor det kun er en spiller/monster der kan foretage noget af gangen. Det vil sige, at man er nødsaget til at vente med sin næste handling indtil alle andre har foretaget deres handling. Den anden måde er en ikke turbaseret strategi, hvor det i stedet er tiden der afgør om man kan foretage en ny handling. En måde det kan gøres på, er ved at benytte observer mønstret, hvor de enkelte objekter i spillet kan abonnere på tiden. Når tidsklassen så sender en besked ud, kan de enkelte objekter så foretage en handling. Vi har valgt at bruge metode nummer 2. Grunden hertil er, at det giver mulighed for mere liv i spillet, samt at vi flytter træk-kontrollen fra den menneskelige spiller til computeren. Med mere liv i spillet tænker vi på, at det er muligt at blive overrasket imens man foreksempel prøver på at finde ud af hvilken vej man skal gå.

1.4 Hvad er vores plan for spillet

Før man går igang med udvikling af et program, er det vigtigt at man har besluttet hvornår man er tilfreds med sit produkt. I vores tilfælde har vi inddelt processen i 5 delprocesser.

- Generering af verden.

- Få en person til at bevæge sig rundt i spillet
- Kunne foretage handlinger såsom at tage og bruge ting
- Få nogle autonome monster ind i spillet.
- Kunne gennemføre en kamp.

Vi har allerede nu besluttet, at det ikke skal være muligt at gemme sit spil på et persistent lager.

Idet denne opgave er en del af evalueringen i kurset OOP, vil den selvfølgelig blive løst objekt-orienteret, samt vi vil benytte os af relevante design mønstre.