

SUPER SEJT KODEPROJEKT

TIDSFORDRIV TIL ODA

Del 1

Basal character creator

FREDERIK ETTRUP LARSEN

29. juni 2019

1 Input håndtering

Når man sådan skal til at lave et super sejt kodeprojekt som det her, så er det ret vigtigt at man ved hvordan man skal håndtere input. Med håndtere input skal der forstås to ting, først og fremmest skal man vide hvordan man overhovedet kan læse noget af det som brugeren gør. Men, desuden skal man også kunne håndtere hvis brugeren gør noget som de ikke bør! I C er den nemmeste måde at håndtere input nok "scanf" functionen. Hvordan virker den spørger du? Simpelt, "scanf" udnytter en af de smarte features i C, nemlig pointers. Man skal bare fortælle den hvad den skal læse, og så giver man den adresse som den skal gemme værdien på. Det smarte er at man kan lade en variabel ændre værdi ved at lade den være et input til en funktion som ikke returnerer noget. Det er især smart hvis man skal deale med strenge (som egentligt bare er arrays af chars).

Et typisk eksempel på hvordan man kan bruge scanf er som følger:

```
#include <stdio.h> // Virker ikke hvis den ikke er der

int main(){
    char string[20];
    int n;
    scanf("%s", string); // bemærk, her giver vi bare string som input,
    //da den allerede er en pointer
    scanf("%d", &n); // Her giver vi &n, som er adressen på n
    //altså en pointer til n

    printf("%d, %s\n", n, string);

    return 0;
}
```

1.1 Opgaven

Den allerførste del af d&d 5e karakter håndterings programmet er at kunne skabe karakterer, så de første par opgaver kommer til at dreje sig om det. Det allerførste som programmet skal kunne er som følger:

Når programmet åbner, så bliver man bedt om navnet på sin karakter (det må ikke være længere end 100 tegn), så spørger den om race, så klasse og til sidst baggrund. Til alle tre skal den give en nummereret liste med alle mulighederne i player's handbook, brugeren taster da et tal svarende til den mulighed de ønsker at vælge.

Til sidst skal der printes en besked, hvor der står:

"Du spiller som [navn], en [race] [klasse], med baggrunden [baggrund]"

Programmet skal desuden bede brugeren om at taste om, hvis de vælger en mulighed som ikke er tilgængelig, og det skal så vidt muligt være muligt at have special tegn som æ, è eller î hvis man gerne vil have et fjollet navn.

1.2 Extra points

Gør sådan at terminal vinduet bliver ryddet hvergang den skal til at printe en ny menu, så det hele bliver lidt pænere, hvordan man gør det afhænger af hvordan du compiler og kører koden, så hvor svært det er ved jeg ikke, men det er helt klart noget du bliver bedt om før eller siden :)

2 Abilities

Abilities er en af de vigtigste ting i hele spillet, det er dem som på mange måder definerer hvem ens karakter egentligt er, og af den grund er det aldeles vigtigt, at programmet kan finde ud af at håndtere abilities. Så hvordan virker abilities?

Alle karakterer har seks abilities, men det kommer vi mere ind på senere. Alle abilities kan som udgangspunkt ligge mellem 1 og 30, men det er meget sjældent at komme over 20 i en ability, og det skal virkeligt gå galt for at få en ability på mindre end 5, ideen er lidt at en gennemsnitlig person har 10 i alle abilities.

2.1 Opgaven

Du skal nu udvide dit program fra sidst. Nu når man har valgt race, klasse og baggrund skal der laves en "statblock" til karakteren. Hver ability findes ved at rulle tre terninger, en 4-sidet terning (1d4), en 6-sidet terning (1d6) og en 8-sidet terning (1d8). Hvis man ruller to 1'ere, så må man rulle dem om, men hvis man ruller tre 1'ere, så må man kun rulle to af dem om, man kan dog kun omrulle to 1'ere en gang per ability.

Hele den ballade skal man igennem seks gange, og man har nu seks numre. Hvis der ikke er nogen af disse numre som er 15 eller over, så skal den næst laveste ability ændres til at være 15.

I den her opgave er der et par store ting at deale med, først og fremmest skal man selvfølgelig have en måde at generere sine tilfældige tal på. Desuden, skal man have en måde at sortere i de seks abilities for at finde den næst laveste.

Den skal så til sidst displaye alle de seks abilities, og karakterens navn race og baggrund som før.

3 Abilities men mere

Som sagt er der 6 abilities, navnligt, strength, dexterity, constitution, intelligence, wisdom og charisma. Værdien af hver stat svarer til hvor ”god” man er til den ting. For eksempel kan det være at en super buff body builder har en strength på 18, hvor en socialt akavet nørd måske har en charisma på 8, men en intelligence på 20. De forskellige classes i spillet er alle sammen bygget op omkring en eller to af de her abilities, en kriger i pladerustning har nok høj strength, mens en tyv der lusker rundt i mørket nok har høj dexterity. Ud over hvad man ruller, bliver ens abilities også påvirket af hvilken race man har valgt. En halv-ork er foreksempel stærkere end en hobbit, men en hobbit er tilgængelig mere adræt end en halv-ork.

3.1 Opgaven

Så i sidste opgave lavede du forhåbenligt en ting som kan lave nogle værdier, men! Det er ikke nok i sig selv, programmet skal kunne tildele tallene til de forskellige abilities.

At opstille en mere formel måde at gøre det her på er lidt mærkeligt, for på mange måder, er en del af det sjove i spillet at lave en karakter som måske ikke er super optimeret, men bare for det her projekts skyld, laver vi noget som nok oftest kommer til at give noget som ikke helt er skrald.

Først skal du udvide race vælger menuen til så man også kan vælge subrace til alle de racer hvor det er relevant.

På side 45 i Players Handbook er der en tabel med alle klasserne, og deres ”primary ability”. Med udgangspunkt i dette skal abilities fordeles. Først skal ens primary ability være så høj som muligt. I et par tilfælde er der to primary abilities, i de tilfælde vil man gerne have at den første er højest, og at den anden er næst højest. I fighter’en står der ”strength or dexterity” i det tilfælde skal der genereres to statblocks, en til hver primary ability. Som brugeren så skal vælge i mellem.

Om man har to primary abilities eller ej, så skal den næste være constitution (den er bare generelt rar at have så høj som muligt, da den afgør hvor mange liv man har, og derved hvor mange tæv man kan tage).

De sidste tre eller fire skal fordeles ud fra karakterens baggrund som det ses i tabel 1. Tag listen af abilities som der allerede er, og tildel de sidste så de følger den her blok, hvis en eller flere af abilities her allerede er tildelt, så afgør det tilfældigt.

Hele det her system skal også tage højde for spillerens race, for hver race lægger noget til deres abilities. Dette kan ses i Players Handbook under **Ability score increase** for hver race og subrace. Som udgangspunkt betyder dette at den blot skal huske at lægge værdierne til til sidst, men i nogle tilfælde, må man selv vælge hvilke abilities man øger. I det tilfælde skal man følge de parametre for øgelsen givet i regelsættet, og så først øge primary abilities, så constitution og så abilities fra baggrund.

Når alt dette er gjort så man have et system som kan generere en karakter med navn, race, klasse og baggrund, med en stat block som er nogenlunde fornuftigt.

Dette er slutningen på del 1 af det super seje kodeprojekt.

Baggrund	Primær	Sekundær	Tertiær
Acolyte	Wisdom	Intelligence	Charisma
Charlatan	Charisma	Dexterity	Wisdom
Criminal	Dexterity	Charisma	Intelligence
Entertainer	Charisma	Dexterity	Strength
Folk Hero	Wisdom	Strength	Charisma
Guild Artisan	Charisma	Wisdom	Intelligence
Hermit	Wisdom	Intelligence	Strength
Noble	Charisma	Intelligence	Strength
Outlander	Strength	Wisdom	Dexterity
Sage	Intelligence	Wisdom	Charisma
Sailor	Strength	Wisdom	Charisma
Soldier	Strength	Dexterity	Charisma
Urchin	Dexterity	Wisdom	Charisma

Tabel 1: Hvilke abilities en baggrund ligger op til