Projektopgave: Snake

Carsten Witt

5. januar 2015

Om Opgaven

Projektopgaven indgår i kursets samlede bedømmelse. Opgaven skal laves i grupper af 2 til 4 personer, men den fortrukne gruppestørrelse er 3 til 4. I skal selv danne grupperne og registrere dem på Campusnet senest d. 5. januar kl. 23:59. Selve opgaven skal også afleveres via CampusNet.

Opgaven går overordnet ud på at designe og dokumentere et computerspil baseret på det klassiske *Snake* spil (også kendt som *orm* eller *slange*). Se figur 1. Detaljerede beskrivelser af spillet, varianter, historie, etc., kan findes på hjemmesiden `http://en.wikipedia.org/wiki/Snake_(video_game)`.

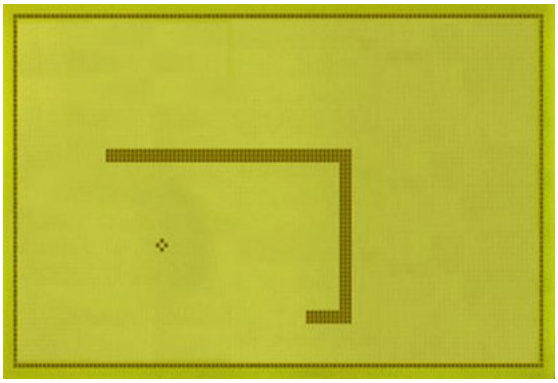
Som udgangspunkt for opgaven anvender vi en simpel version af Snake, kaldt Simpel-Snake, som er inspireret af en udgave på en historisk mobiltelefon. Simpel-Snake foregår på følgende måde: Spillet spilles på en todimensionel diskret torus af $n \times m$ felter, hvor n og m er parametre til spillet. Torussen adskiller sig fra et gitter, idet den øverste række grænser op ad den nederste og ligeledes den yderst til venstre kolonne ad den yderst til højre, hvorved man ved eksempelvis at bevæge sig ud gennem den højre side derved bevæger sig ind gennem den venstre side. I det følgende vil vi referere til torussen som *banen*. Hvert felt på banen er enten tomt eller optaget af et af objekterne beskrevet nedenfor.

På ethvert tidspunkt under spillet befinder der sig en *slange* på banen. Slangen optager altid en sammenhængende mængde af felter med to yderpunkter, der betegnes som henholdsvis *hovedet* og *halen*. Desuden opretholder slangen en *bevægelsesretning*, som enten er *venstre*, *højre*, *op* eller *ned*.

Ved spillets start består slangen kun af to felter beliggende i banens midte, nærmere betegnet de felter i banen givet ved koordinaterne $(\lfloor n/2 \rfloor, \lfloor m/2 \rfloor)$ og $(\lfloor n/2 \rfloor, \lfloor m/2 \rfloor + 1))$, hvor den første koordinat udgør hovedet mens bevægelsesretningen til start er til venstre. Desuden placeres der en *prik* på et uniformt tilfældigt felt uden for slangens koordinater.

Spillet forløber derefter i et antal *runder*, indtil spilleren eventuelt taber som følge af en *kollision* (se nedenfor). Hver runde består normalt af følgende skridt (det fremhæves, at der som udgangspunkt ikke findes et ur, som automatisk indleder nye runder, hvorved hver runde udelukkende indledes som følge af input fra spilleren):

1. Spilleren beslutter at rykke slangens hoved et felt horisontalt til venstre eller højre, eller et felt vertikalt op eller ned, hvilket angives via input fra tastaturet. Rykkets



Figur 1: Snake kørende på en Nokia 3310 mobiltelefon.

retning må ikke være modsat bevægelsesretningen fra forrige runde. Bevægelsesretningen opdateres så tilsvarende, hvorved det felt, der grænser op ad slangens hoved i bevægelsesretningen, betegnes nu som *målfeltet*.

2. Det følgende sker dernæst afhængig af målfeltets indhold:

   (i) Hvis målfeltet indeholder en prik, så forsvinder prikken med det samme fra feltet (betegnet som at prikken bliver *ædt*), og spilleren får tildelt et point. Samtidig tilføjes målfeltet til slangen og udgør slangens nye hoved, mens halen forbliver den samme. Desuden placeres der en ny prik på et uniformt tilfældigt felt uden for slangen.

   (ii) Hvis målfeltet indeholder del af slangen bortset fra halen, sp er der tale om en *kollision*, hvorved spilleren har tabt og spillet er slut.

   (iii) Ellers bliver målfeltet optaget af slangen og udgør nu slangens nye hoved. Samtidig tømmes feltet med slangens hale. I stedet udgør det felt, som tidligst blev optaget blandt de resterende felter med slangen, den nye hale. Alle øvrige felter, som var del af slangen i den forrige runde, forbliver sådan.

Opgaven består af en *grundlæggende del* samt af en *avanceret del*. Alle grupper skal først lave den grundlæggende del, som derved er obligatorisk. Efter dette haves en fungerende version af Simpel-Snake, som kan bruges som udgangspunkt for den avancerede del, som er åben for en mangfoldighed af retninger afhænging personlige interesser og kompetencer (en ikke-udtømmende liste af oplagte muligheder er givet nedenfor).

Grundlæggende Del

I den grundlæggende del af opgaven skal I konstruere Simpel-Snake som beskrevet ovenfor. Interaktionen med spilleren skal foregå via en simpel grafisk brugergrænseflade og primært benytte tastaturet som input. Størrelsen af torussen skal angives som kommandolinjeparametre til programmet, idet I er tilladt at antage $n, m \in \{5, \cdots, 100\}$.

Avanceret Del

I den avancerede del af opgaven skal I bygge videre på Simpel-Snake fra den grundlæggende del, idet I frit kan eksperimentere med udvidelser og variationer på spillet. En liste af *mulige* udvidelser og variationer i tilfældig rækkefølge er givet her:

Tidspres Bevæg slangen i bevægelsesretningen som beskrevet ovenfor i fastlagte tidsskridt uden spillerens indblanding, som det kendes fra de fleste Snake-varianter. Dermed presses spilleren til at reagere hurtigt for at undgå mulige kollisioner.

Acceleration I forlængelse af tidspres øges hastigheden af spillet langsomt, eksempelvis hver gang en prik bliver ædt.

Sværhedsgrad Tilføj en mekanisme så brugeren kan ændre på sværhedsgraden; for eksempel ved hjælp af øget tidspres eller acceleration.

Banedesign Tilføj mure til banen, der begrænser slangens mulige bevægelser. Hvis slangens hoved rammer en mur, er der ligeledes tale om en kollision. Forbind evt. felter på banen gennem "ormehuller".

Banegenerering I forlængelse af banedesign genereres banen efter et skema tilfældigt ved hver start af spillet. Eksperimentér med mulige strategier for, hvordan en bane kan genereres, således at alle felter stadigvæk kan nås af spilleren.

Tid Tilføj et stopur til spillet, som viser den tid, der er forløbet siden spillets start.

Bonus Tilføj yderligere bonusser til spillet; for eksempel en "super-prik", der giver ekstra point, men som kun findes på banen i kort tid.

Animation og Udseende Animér bevægelsen af slangen. Arbejd med forskellige designs til slangen, prikkerne, etc., eventuelt ved brug af dynamisk tegning.

Dimension Design en 3D-variant af spillet.

Highscore Tilføj en liste over de bedste præstationer i spillet.

Sejr Tilføj en særlig tilstand til spillet, hvori spillet betegnes som værende *vundet* af spilleren, som indtræffer når slangen optager samtlige prikker på banen.

Spillerstrategi Design og implementér en kunstig intelligens til at erstatte spilleren, som forsøger at vinde spillet gennem simple spilstrategier.

Spilindstillinger Tilføj indstillinger i brugerfladen, så man nemt kan ændre størrelse af banen, sværhedsgrad, etc., alt efter behov og dynamisk under programudførslen.

Multiplayer Design en flerpersoners variant af spillet.

Pause Tilføj en funktion til at pause spillet.

Gem/Hent Tilføj en funktion til at gemme og hente spil.

Andet Find på egne udvidelser til og varianter af spillet.

Aflevering, Præsentation og Eksamen

I skal aflevere en rapport, der dokumenterer de udviklede spilvarianter, samt de overvejelser, som I har gjort jer omkring dem. Vi anbefaler, at rapporten skrives i LaTeX og at I benytter versionskontrol til udviklingen, således at I til enhver en tid har adgang til samtlige versioner af spillet, idet en ny version indledes for hver ny tilføjelse. I skal også aflevere kildeteksten til spillet, og vi anbefaler at benytte Model-View-Controller til designet af den grafiske brugergrænseflade. Følgende filer skal uploades via CampusNet:

- En `pdf`-fil indeholdende jeres rapport.
- En eller flere `java`-filer indeholdende jeres kildekode samt eventuelle ekstra filer, der skal bruges til at afvikle programmet (grafik, lyd, etc.). Hvis I har mange filer, må I gerne pakke dem sammen i en `zip`-fil sammen med en `txt`-fil, som kortfattet beskriver, hvordan programmet afvikles, eventuelt pakket som en `jar`-fil.

Efter aflevering skal I præsentere jeres projekt i grupper. Til sidst afholdes der en kort, mundtlig gruppeeksamen. Den endelige karakter gives på baggrund af en helhedsvurdering af jeres tidligere afleveringsopgaver, studieplan, projektopgaven og mundtlige eksamen i sammenfatning. I vurderingen lægges vægt på følgende:

- Kvalitet af dokumenterede overvejelser og valg i forbindelse med projektet (afgrænsning, design, implementation, evaluering, etc.).
- Kvalitet af programmet (korrekthed, robusthed, funktionalitet, effektivitet, etc.).
- Kvalitet af beskrivelse af programmet (virkemåde, opbygning, etc.).
- Kvalitet af projektrapport (præcision, struktur, etc.).
- Kvalitet af brugergrænseflade (overskuelighed, brugervenlighed, etc.).
- Kvalitet af tidligere obligatoriske opgaver, studieplan og individuel eksamen.

Frister og Tidspunkter

Opgaven skal afleveres senest søndag d. 18. januar kl. 23:59 på CampusNet. Eksamenen foregår fra onsdag d. 21. januar til fredag d. 23. januar og vil blive opgivet senere.

Noter, Uddybning og Råd

- Lav den grundlæggende del færdig og skriv rapporten til den, inden I kaster jer over den avancerede del, så den er frisk i erindringen.
- Tag udgangspunkt i Model-View-Controller, når I skal designe og implementere programmet, og beskriv jeres implementation i forhold til MVC i rapporten.

- Brug god tid på at tænke det overordnede design og struktur af jeres program igennem, inden I begynder den videregående implementation.
- Implementér og dokumentér hver tilføjelse til spillet hver for sig, så I ender med en række gradvist mere veludviklede spil, som I kan sammenligne med hinanden.
- Evaluér og sammenlign de udviklede spil med hinanden.
- Selv om den grundlæggende del er forholdsvist afgrænset, så er der stadig ting at overveje. Hver enkelt tilføjelse, som I vælger, skal I sørge at afgrænse og definere rammerne for præcist.
- Notér alle overvejelser, argumenter, resultater, alternativer, etc., undervejs i processen, og brug disse som et udgangspunkt for rapporten.
- Det foreslås at skrive jeres rapport på følgende måde:
  - Organisér jeres noter i afsnit til rapporten.
  - Udvælg relevante noter til hver afsnit og tilføj noter til forglemmelser.
  - Bind noterne sammen til et sammenhængende udkast af hele rapporten.
  - Forbedr udkastet iterativt indtil I har den færdige rapport.
- Brug figurer og metakommunikation i rapporten, skriv kort og præcist og undgå knudrede sætninger og fyldord, idet I sørger for at holde jer på et tekstuelt niveau som værende rettet mod en medstuderende af samme forudsætninger som jer selv.