

# Bobspil

af: Rasmus Olsen & Tobias Fjeldsted



Opgaven der skal løses er at udvikle et virtuelt bobspil, der skal kunne håndtere to spillere.

## Enkle regler for Bob

I virkelighedens bob har man en spilleplade, der er kvadratisk med bänder på siderne. Denne plade har fire huller, et i hvert hjørne. Man har desuden et antal runde bobbrikker, som er 3-4 cm. i diameter, og 1 cm. høje. Disse brikker har desuden et hul i midten med en diameter på ca. 1 cm. Brikkerne findes i to forskellige farver (oftest rød og grøn), hvor hver farve tildeles en spiller.

Langs bänderne på pladen ca. 10 cm. inde på pladen er optegnet (ofte med rød) en linie for hver kant. Hver spiller tilhører 1-2 linier. Når en det er en spillers tur skal han placere en sort bobbrik (som ofte er lidt større end de førnævnte), på hans tilhørende linie. Derefter skal spilleren støde med en kø til denne brik, således at den rammer en af de andre bobbrikker, der har spillerens tildelte farve, og denne ryger i hul. Lykkes dette får spilleren et skud mere.

Vinderen er den, der først får alle sine bobbrikker i hul.

## Overvejelser i forbindelse med program

### 1. *Gameplay*

Når det bliver en spillers tur vælger han et sted på sin linie hvor den sorte bobbrik skal placeres. Dette kunne f.eks. gøres med et museklik. Derefter styres retningen som køen skal føres i f.eks. med musen. Derefter skal kraften hvormed stødet foretages styres ved at spilleren holder museknappen nede på køen og fører den hen i mod den sorte brik. I det øjeblik, hvor køen og brikken mødes overføres kraften fra køen til brikken, og

det simuleres og vises hvorledes brikkerne vil støde sammen og hvilke eventuelle brikker, som ryger i hul.

## 2. Programmeringsopgaver

I forbindelse med implementeringen af ovenstående gameplay står vi foran nogle programmeringsmæssige udfordringer:

- Muselyttere: der skal implementeres muselyttere på den måde at det er muligt at foretage stød på den ovenover beskrevne måde.
- Grafik: Grafikken er en vigtig del af spillet, da spilleren skal kunne se hvad resultatet er af hans stød. Denne skal implementeres på den mest optimale måde, således at brikkerne vises på den korrekte måde ved kollisioner.
- Fysik: Når to bobbrikker støder sammen i virkeligheden kan man betragte sammenstøddet som et *elastisk stød*. Udfordringen er at beregne i hvilken retning og i hvor høj fart brikkerne bevæger sig efter støddet, hvilket dog kan klares idet impuls og kinetisk energi er bevaret i støddet. Målet rent programmeringsmæssigt er i at implementere de fysiske love korrekt, således at spilleren oplever en virkelighedstro reaktion i sammenstød mellem bobbrikkerne. Derudover skal en gnidningsmodstand imellem bordet og brikkerne implementeres, således at brikkerne ikke bliver ved med at bevæge sig.

## 3. Udvidelsesmuligheder

Spillet i sin simple form indeholder mange udvidelsesmuligheder, som ud fra det oprindelige design, skal være mulig at implementere.

- Sort i hul: Når en spiller skyder den sorte brik i hul, skal modstanderen have mulighed for at tage en af spillerens brikker op af hullerne og placere den på et område i midten af pladen.
- Brikken er bob: Når en brik tilfældigvis lander således at man kan se linien igennem hullet på den, må spilleren, hvis linie den ligger på, støde direkte til den uden at miste sin tur.
- Pust: en brik tilfældigvis lander således at man kan se et af hullerne i pladen igennem hullet på brikken har spilleren lov til at puste til brikken. Dette kunne implementeres på en anden måde i computerversionen af spillet.
- Brik ud over kanten: Når en brik ryger ud over kanten i virkelighedens bob skal den placeres i et af hjørnene på pladen, bagved et af hullerne, for derfor at gøre det meget vanskeligt at få denne brik i hul. Denne regel kan implementeres i programmet, hvis brikkerne har mulighed for at ryge ud over banderne. Brikkerne kunne stige i chancen for at ryge ud over proportionalt med deres hastighed i det øjeblik, hvor de kolliderede med banden.