

Sänka skepp

Sänka skepp är en riktig semesterklassiker. Normalt har man två deltagare, men en utökning till flera är inte svårt att tänka sig. De båda spelarna placerar ut sin armada på ett rutat papper. Uppgiften är att skuta ner motståndarens båtar innan ens egna är helt sänkta.

Armadan består av ett antal (olikformade) båtar där varje båt tar upp ett antal rutor på pappret. Vanligast är att båtar bestående av en länga av 2 till 5 rutor, t.ex. 4 st tvåor, 2 st treor, 2 st fyror och 1 st femma. Båtarna får roteras 90°, men måste placeras en ruta ifrån en intilliggande båt.

```
      123456789012
A  xx          xx
B           x
C  xxx  x  x  x
D           x  x  x
E  xxxxx x  x  x
F           x  x
G  xx          xx
```

Spelarna avlossar i tur och ordning ett skott mot en vald motståndare genom att säga t.ex. *skott A1* där A1 motsvarar rad A och kolumn 1. Motståndaren svarar med *miss* om skottet missare dennes båtar, *träff* om skottet träffar någon av dennes båtar eller *sänkt* om skottet träffar sista hela rutan i en båt. Spelaren noterar svaret i sitt eget (ursprungligen tomma) sjökort över motståndaren armada, för att undvika redan beskjutna rutor eller rutor som omöjligen kan innehålla båtar.

Din uppgift är att skapa ett Sänka skepp-spel. De funktionella och icke-funktionella krav som beskrivs nedan måste uppfyllas. Nedan finns även idéer till ytterligare funktioner du kan lägga till om du har tid och lust.

Funktionella krav

- Programmet ska följa gängse Sänka skepp-regler.
- Programmet ska hantera minst 2 spelare, människor och datorspelare.
- Inför varje omgång ska en procent sats över varje spelares träffprocent och skadeprocent presenteras. Träffprocenten är andelen skott som givit träff, och skadeprocenten är andelen av motståndarens alla båtrutor som är träffade (av någon spelare).
- Det ska finnas en lista över bästa resultat som sparas när programmet avslutas. Resultat kan t.ex. lägst antal skott innan vinst.

Icke-funktionella krav

- Java ska användas som programmeringsspråk.
- Programmet ska ha en god objektorienterad design. Designen ska finnas dokumenterad, t.ex. i diagramform.
- Programmet ska vara så lätt att lära sig att en normalbegåvad labbassistent kan hantera det efter några minuters utbildning.

Idéer till ytterligare funktioner

- Grafisk visning/styrning i stället för konsolbaserad dito.
- Fler än 2 spelare.
- Spelarstatistik
- Placeringsförslag
- Nätverksspelare
- API-stöd för att definiera nya båttyper, t.ex.
xx
xxxx
- Användarstöd för att definiera nya båttyper