

# 133 Sänka skepp

P-uppgiften ska göras individuellt. Läs EECS:s hederskodex innan du börjar!  
Val av P-uppgift görs på övning - du kan inte välja uppgift här Dessa sidor är enbart för referens.

**Varudeklaration:** Datastrukturer. Viss skärmhantering. I extrauppgiften tillkommer sortering.

Du ska skriva ett program som lägger ut en mängd fartyg på en yta (fartygspositioner läses in från fil). Användaren ska sedan kunna beskjuta ytan och programmet ska meddela resultatet (envägs beskjutning). Spelet avslutas då användaren så önskar eller efter att alla fartyg är sänkta.

Programmet skall efter varje skott:

- Skriva ut resultatet av skottet, träff eller bom.

Användaren ska via menyer kunna göra sin olika val. Huvudmenyn kan se ut så här:

Dina valmöjligheter (1-3):

- 1) Beskjuta fiendefartyg
- 2) Fuska lite, tjuvkika på fiendefartygen
- 3) Avsluta

Ditt val (1-3):

De olika menyerna innebär:

**Beskjutning mot fiendefartygen:** På skärmen ska spelplanen och skottmarkeringarna visas i en schackbrädesliknande figur. Användaren ska få skjuta ett eller flera skott och därefter återgå till huvudmenyn. Under spelplanen ska träffprocenten skrivas ut.

**Fuska lite, tjuvkika på fiendefartygen:** Alla spelare har olika moral och alla måste tillfredsställas. Spelplanen presenteras med fiendefartygen fullt tydligt markerade tillsammans med skottmarkeringarna.

**Avsluta:** Användaren ska få möjlighet att titta på de ej sänkta fartygens lägen före avslutning. Spelplanen presenteras på samma sätt som vid tjuvkiken.

Du bestämmer, som programkonstruktör, enväldigt spelplanens storlek (t ex 8x8). Fartygens positioner läses in från fil.

För att göra spelet mindre förutsägbart ska du ha minst tio olika filer med fartygsplacementar. Programmet ska slumpa fram vilken fil som ska användas. I filen `random.txt` får du tips till slumpfunktionen.

I presentationen av spelplanen ska det tydligt framgå:

- Rutnätet, varje koordinat har en egen ruta, se bilden nedan.
- Resultaten av skotten, t.ex.

**Träffar** markeras med '#'

**Bommar** med 'o'

**Fartyg** , (ej träffade) med 'X'

- Koordinater i kanterna.

```
-----  
|   | 1 | 2 | 3 | 4 |  
-----  
| A | O | X | O |   |  
-----  
| B |   | # |   | X |  
-----  
| C |   | O |   | X |  
-----
```

**Extrauppgift, betyg C:** Inför felkontroll av menyalternativ och inmatning.

Skriva ut ett felmeddelande om skottet var ogiltigt, dvs utanför planen eller rutan redan beskjuten. Användaren får i detta fall skjuta på nytt.

**Extrauppgift, betyg B:** Slumpa fram fartygens positioner istället för att läsa från fil. Du bestämmer fartygens antal samt storlek, vilket presenteras i början av programmet.

Du skall programmera på ett sådant sätt att man i efterhand lätt kan förändra fartygsuppsättningen.

Regler för placering av fartygen:

- Det ska finnas fartyg av olika längder, 1 - 5 rutor
- Fartygen ska kunna hamna både lodrätt och vågrätt
- Fartygen får inte överlappa
- Två fartyg får aldrig ligga precis intill varandra, utan det måste alltid finnas tomma rutor emellan.

Om det blev en träff och fartygets samtliga rutor är träffade är fartyget sänkt. Detta ska meddelas användaren. Dessutom ska programmet markera "beskjuten" på alla kringliggande rutor eftersom det inte kan ligga några fartyg där.

Programmet ska dessutom uppdatera en high-score mellan körningarna. Om spelaren är bland de tio bästa (träffprocenten är måttet på spelarens skicklighet) ska namn efterfrågas. Listan lagras efter körningen på en textfil. Denna läses in vid ny programstart. Listan ska sorteras.

**Extrauppgift, betyg A:** Gör ett grafiskt gränssnitt där man klickar för att beskjuta.

**Datafiler och hjälpfiler:** random.txt