



Examinationsuppgift 2: Sänka skepp

Grundkrav:

I den här versionen av Sänka skepp ska en spelare antingen kunna möta en annan spelare eller en datorstyrd spelare (eller låta användaren välja inne i spelet).

Varje spelare har en egen spelplan likt ett rutnät, en 10x10 matris, där fem stycken skepp placeras. Skeppen placeras i början av spelet och det ska vara möjligt att placera skeppen vertikalt eller horisontellt med hjälp av mus och/eller tangentbordet.

Skeppen ska placeras innanför en spelares spelplan, det ska alltså inte vara möjligt att placera ett skepp utanför sin egen spelplan.

En spelare har fem stycken skepp i olika storlekar:

<i>Skepp</i>	<i>Storlek</i>
Hangarfartyg / Aircraft carrier	5x1
Kryssare / Battleship	4x1
Ubåt / Submarine	3x1
Fregatt / Destroyer	3x1
Korvett / Patrol boat	2x1

Spelar man mot en annan spelare ska skeppen inte synas efter att man placerat ut dem. Men skulle man möta en datorstyrd spelare ska ens egna skepp synas.

Spelet ska vara turbaserat, vilket i det här fallet blir att varje spelare har ett skott som den kan placera i motståndarens rutnät, sedan är det den andra spelarens tur.

När man placera ett skott i motståndarens rutnät ska det tydligt visas om det är en träff eller en miss.

Om man sänker ett skepp ska skeppet synliggöras på spelplanen.

En spelare vinner när man har sänkt alla skeppen på motståndarens spelplan. Det ska tydligt visas vem som vunnit och vem som förlorat när en spelrunda tagit slut.

Det ska finnas minst två klasser i projektet förutom de som skapas automatiskt från början. Spelet ska vara spelbart och kunna styras med hjälp av mus och tangentbord.

Varje spelares rutnät ska ritas ut, och en sprite sheet måste nyttjas i spelet på något lämpligt sätt. Storleken på fönstret ska vara 1280 x 720.

Exempel på funktionalitet (ca poäng):

- Variabel storlek på rutnätet (1 - 4p)
- Variabelt antal skepp (1 – 4p)
- Antal skjutna skott räknas och visas på skärmen (1p)
- Ett synligt skott ska skjutas dit man placerar ett skott (2 - 3p)
- Skeppens namn skrivs ut på lämpligt ställe (1 - 3p)
- HUD (1 - 3p)
- Möjlighet att ändra fönsterstorleken (2 - 3p)
- Meny (1 - 3p)
- Omstart (1p)
- Flera olika typer av skott (t.ex. täcker flera rutor) (2 - 4p)

Exempel på bra kodstruktur

Stor vikt kommer att läggas på hur koden är strukturerad.

- Namngivning (1p)
- Läsbarhet (1p)
- Klassindelning (1 - 3p)
- Blockstorlek (1 - 3p)
- Konsekvent (1p)

Betygsskala:

E: Grundkrav

D: Grundkrav + 5p

C: Grundkrav + 10p

B: Grundkrav + 20p, varav 6p kodstruktur

A: Grundkrav + 25p, varav 8p kodstruktur

Fråga din lärare om du är osäker på vad som ger poäng och hur mycket poäng man får, innan du börjar implementera ny funktionalitet.

Man ska kunna motivera tänkt betyg för en lärare!

För att betyget skall registreras krävs att du lägger upp ditt projekt på it's learning **före** deadline (se it's learning) samt att du gjort avslutningsenkäten. Annars tilldelas ett nytt projekt som skall göras på egen hand utanför labbtid.

Lycka till!

/Lärarna ☺