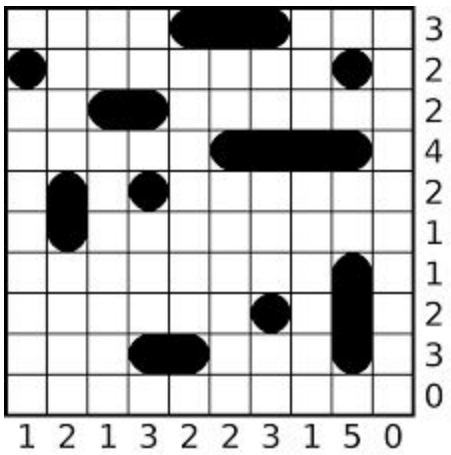
Uppgift Sänka skepp

Ni ska skapa ett Sänka skepp-spel. Sänka skepp går till så att två spelare har var sitt bräde varpå de placerar ut ett gäng skepp. Placeringen av skeppen är hemlig för den andre spelaren. Skeppen är placerade så att inget skepp vidrör det andra, inte heller diagonalt.

Motståndaren gissar sedan vart den andra spelarens skepp är belägna och väljer en ruta att “skjuta” på. Den andra spelaren berättar då om skottet träffade eller missade. Om skottet träffade, så får den som sköt fortsätta sin tur. Ett skepp sjunker när hela skeppet har blivit träffat (så om det upptar tre rutor gäller det för motståndaren att träffa alla tre). Vinnaren är den som först lyckas hitta och sänka alla motståndarens skepp.



Bilden ovan visar ett spelbräde där antalet rutor som upptas av skepp står i slutet av raden respektive kolumnen.

Programmet ska se till att ingen spelare lägger ett skepp på en ogiltig plats, samt registrera när någon vinner. Utöver detta ska spelet spara statistik över hur många vinster respektive spelare har.

Användaren ska kunna mata in hur hen vill lägga sina fartyg, samt få en presentation av de rutor som hen har skjutit på.

Del 1 - Spelbräde

Skapa en eller flera funktioner som skriver ut ett spelbräde med 10x10 rutor, ett par newlines/radbrytningar och sedan ett till spelbräde om 10x10 rutor. Skriv sedan en funktion som tar skepp-positioner som argument, och som fyller den övre spelplanen med skeppens positioner.

Input och ouput

Skriv en funktion som tar input från användaren på vart hen vill ha sina skepp. I klassiska Sänka skepp ingår ett skepp med 4 rutors längd, 2 skepp med 3 rutors längd, 3 skepp med 2 rutors längd, och 4 skepp med 1 rutas längd, men det är okej om ni har ett annat upplägg. Oavsett har varje skepp en längd, startkoordinater och en riktning (horisontell eller vertikal).

Om användaren placerar skepp ovanpå ett annat skepp ska programmet protestera och be om ny input. Detsamma gäller om skeppet skulle råka hamna utanför spelplanen.

Filhantering

I denna variant av spelet ska ni spela mot en dator. Datorns placering av skepp kan i början av utvecklingen hårdkodas in i programmet, men byts sedan ut mot att man laddar in olika placeringar från fil.

Del 2 - Spela mot motståndare

På den nedre spelplanen samlas den mänskliga spelarens försök att träffa motståndarens skepp. Alla försök markeras på brädet, med en symbol om försöket träffade ett skepp och en annan om det missade.

Villkor och loopar

Programmet ska agera motspelare med den mänskliga användaren. Det räcker dock om den är en relativt dålig motspelare och bara slumpar sina drag. Om datorn eller användaren träffar ett skepp ska den som träffat få ett till drag innan turen går vidare. Se till att den mänskliga spelarens drag uppdateras på nedre brädet och att datorns drag uppdateras på det övre.

Se till att spelet registrerar vem som vinner och hur många försök det tog. Skapa en fil där denna statistik sparas. Om filen redan finns så ska resultatet läggas till det tidigare.

När programmet avslutas ska användaren få en förfrågan om hen vill spara sin egen placering av skepp. Om hen svarar ja så sparas inmatningarna i en fil, som sedan kan användas som “datorns” placeringar i framtida spel.

Del 3 - Spara upplägg

Lägg till funktionalitet som låter användaren välja vilken fil hen vill använda för att mata in datorns placering av skepp.

Utökningar

- Datorn spelar lite smartare. Om den träffar ett skepp, så är nästa drag på en ruta som angränsar till den ruta där träffen gjordes.
- Lägg till en möjlighet att spara ett oavslutat spel, för att kunna uppta det vid ett senare tillfälle.
- Utöka spelet så att det går att generera motståndar-brickor automatiskt.
- Spara tillräckligt mycket data för att kunna visa
    - Totala antalet spelade spel
    - Genomsnittligt antal drag
    - Eller annan statistik du tycker vore intressant.
- Låt användarna registrera sig med användarnamn och visa sedan antal vunna spel per användare, samt antal drag det tog att vinna.
    - Extra bonus: Skydda användarna med lösenord.
- Spara en historik över alla hittills spelade spel, så att användaren kan ta upp slutresultatet tillsammans med datum och tid för spelet
    - Extra bonus: Spara vilken ordning dragen gjordes, så användaren kan spela upp en repris av spelet