Mattias Selldén
Systemutveckling IT och Samhälle, HT20, GPA131

Sänkaskepp

Som VG-uppgift har jag valt att skapa ett klassiskt sänkaskepp-spel. Eftersom jag programmerat lite i bl.a. Python tidigare och redan var bekant med datatyper, arrayer, listor, metoder och även lite rekursion så var jag sugen på att göra något mer objektorienterat för att lära mig det bättre.

Tanken har funnits från början att kunna bygga vidare på spelet även efter kursen och att det på sikt ska kunna köras som en webapp.

För att få klasserna mer oberoende av varandra och för att senare även kunna använda koden för att lära mig enhets-testning så har jag provat att koppla ihop klasserna med hjälp av interface, mest som experiment. Jag är medveten om att det är lite onödigt i detta fall, och att det gör koden lite mer svårläslig. Men det har som sagt varit mest för lärandets skull.

Programmet består för närvarande av följande klasser:

Program
Här snurrar meny-loopen tills programmet avslutas. Väljer man att spela skapas ett objekt av klassen Game och metoden PlayGame anropas. Man kan också välja att anropa PrintHelp för att visa hjälpavsnittet.

Game
I klassen Game instansieras resten av objekten i programmet. Objekten som instansieras är bland annat Player, ShipGrid och LogGrid och alla båtar.

Player
Innehåller spelarens namn, ShipGrid och LogGrid.

Grid (abstract)
Innehåller en 2D-array (Grid[,]) för spelplanen samt X- och Y-axel. Här finns metoden för att skriva ut spelplan.

ShipGrid : Grid
Ärver från klassen Grid. Innehåller en lista av typ Ship där alla båtar lagras. Innehåller metoder för att hitta ledig position för båtarna och för att lägga till dem i listan och rita ut dem på spelplanen. Här finns också metoder för att hantera de skott som motståndaren skjuter.

LogGrid : Grid
Innehåller en lista på kvarvarande koordinater att skjuta. I LogGrid.Grid[,] loggas skotten man skjuter.

Ship
Håller information om varje båt så som id, längd, typ och antal träffar.

Help
Innehåller metoden PrintHelp som skriver ut hjälpavsnittet.

Graphics
En hjälpklass med metoder för att skriva ut ascii-art och färger i programmet. Kanske hade det varit bättre att göra denna klass static? Fördelar / nackdelar?

Tänkta förbättringar och möjliga vidareutvecklig:

-Skrota interfacen? Eller är det värt att ha kvar dem för kommande testnings skull?
-Se över accessmodifiers lite noggrannare.
-Se över status-metoden så den bara visar kvarvarande båtar
-Fixa en bättre Name-parser som hanterar om spelarens namn slutar på ”s” eller inte.
-Är det bättre att instansiera och injicera alla objekt redan i Program klassen? Nu händer det mesta i Game.

-Bättre dator-AI, skjuta rent runt tidigare träffar innan nästa slumpade skott.
-Möjlighet att välja svårighetsnivå.
-Möjlighet att själv välja antal och typ av båtar.
-Möjlighet att välja spelläge: ett skott i taget, tre skott i taget, fortsätta efter träff.
-Byta så att varje ruta på planen innehåller ett objekt, även rutor med bara hav. Skulle bla. möjliggöra högre shipId än 7.
-Rita snyggare spelplan och snyggare båtar.
-Skriva ut träffarna direkt i planen utan att rita om hela konsoll-fönstret.
-Spara spelet, lagra alla object-state till fil. För att sedan kunna ladda dessa igen.
-Skapa topplista.
-Logga till fil.

-Konvertera till web-app.
-Spela online mot någon annan.

Jag tar gärna emot feedback på hur man skulle kunna strukturera programmet på bättre sätt 😊