

Skapa ett spel som spelas på en kvadratisk spelplan med ett grafiskt gränssnitt. Exempel på sådana spel är TicTacToe, Femtonspelet, Memory, Othello, Minesweeper. Använd dig av model-view-control (MVC) samt andra designmönster. Du jobbar antingen själv eller tillsammans med en annan student med en metod som kallas parprogrammering. Ni presenterar inför klassen de designval ni gjort.

Av exemplen ovan så har TicTacToe och femtonspelet ungefär samma svårighetsgrad, Memory är lite jobbigare och Othello/Minesweeper är betydligt svårare.

Spelet

Spelet ska

- vara spelbart mellan två människor som turas om att välja en ruta (i de spel det är tillämpligt)
- meddela användaren på något sätt om man försöker välja en redan vald ruta
- Spelet ska implementera nedanstående interface

```
public interface Game {  
    public boolean move(int i, int j);  
    public String getStatus(int i, int j);  
    public String getMessage();  
}
```

- Spelet spelas enligt interfacet Game, endast metoderna i Game får användas. Själva spelet går till så att användaren klickar på en ruta och i och j avläses. Därefter anropas metoderna i modellen: Först move(i,j) och sedan getMessage() och sen eventuellt getStatus(i,j) för uppdatering av alla knapparna. För de svårare spelen får interfacet modifieras i viss mån. Kolla med Martin först innan ni gör några ändringar.

För TicTacToe kan klassens skal se ut såhär:

```
public class TicTacToeModel implements Game {  
    // Implementera Game-metoderna  
    // Deklarera variabler och övriga metoder som ni  
    // tycker behövs  
}
```

- View/Control-klassen/klasserna får inte känna till att det är ett specifikt spel som spelas. Utan den/de klasserna ska kunna spela ett generellt spel av typ Game som spelas på ett nxn-bräde.
- Hur det grafiska gränssnittet ska se ut är frivilligt så länge ett rutnät om n x n rutor presenteras. Det ska även vara möjligt att se ett felmeddelande när man klickar på en upptagen ruta, men hur detta presenteras är frivilligt. Kommer ni på ytterligare funktionalitet att ha med i spelet är det tillåtet att ta med förutsatt att tiden tillåter och att betygskraven redan är uppfyllda.

Krav

För **godkänt** ska spelet förutom att uppfylla det som beskrivits ovan minst uppfylla följande krav:

- Spelet ska ha ett grafiskt gränssnitt och använda sig av MVC
- Enum används på lämpligt ställe
- Designmönstret Singleton används på lämpligt ställe
- Spelet har felhantering där ett objekt av klassen Exception kastas
- Rutnätets storlek ska vara dynamiskt och bestämmas genom indata från användaren (i de spel där det är tillämpligt). Exempelvis kan konstruktorn i er View/Control ta in en int som parameter som anger rutnätets storlek.

Som i tidigare uppgifter ska koden även vara välstrukturerad, dokumenterad och följa objektorientering samt en etablerad stilguide.

För betyget **väl godkänt** ska spelet, utöver kraven för godkänt, minst uppfylla nedan krav

- Spelet innehåller en abstrakt klass eller interface (utöver Game) på lämpligt ställe
- Ytterligare ett designmönster används på motiverat ställe
- Felhantering med en egenskriven klass som ärver från Exception
- Spelet innehåller minst en testklass med testmetoder
- Man ska kunna välja mellan att spela mot en människa eller mot en datorstyrd spelare

samt ett av följande alternativ

- Har ni valt femtonspelet, TicTacToe eller något annat spel av samma svårighetsgrad så välj ett annat spel än det i grunduppgiften och implementera det som ett fullständigt Game och gör ett spelbart spel med er View/Controll.
- Om ni väljer Othello, Memory, Minesweeper eller något annat spel av samma svårighetsgrad i grunduppgiften så gäller det som VG-krav också och ni behöver således inte implementera ett spel till.

Redovisning

Redovisningen är en muntlig presentation inför klassen där ni visar ert spel och motiverar hur koden uppfyller de olika kraven. Lägg upp presentationen så att de lösningar ni är mest nöjda med prioriteras. Ni ska båda kunna redogöra för samtlig kod och för ungefärlig tid ni spenderat som navigatör/förare om ni har parprogrammerat.

Tid för redovisning är cirka 15 min per grupp.

Redovisning sker **onsdag 1 februari**.

Vad händer om jag inte redovisar onsdag 1 februari?

Om du/ni utan giltigt skäl inte redovisar onsdag 1a februari så gäller följande:

1. Du kan inte erhålla ett VG i betyg
2. Du får göra en enskild *skriftlig kompletteringsuppgift* där du argumenterar för att ditt program uppfyller samtliga krav för godkänt. Lämna in på Slack som pdf (**är det inte en pdf kommer jag inte ens bemöda mig att öppna filen och ni får IG. Jag tänker inte ladda ned programvara bara för att kunna läsa er filer**). 2-3 a4, 12 pt Times New Roman, normal radavstånd.

Lämnas in senast fredag 10/2 för att få ett betyg på kursen nu i vår.

Om du har giltigt skäl för att inte kunna redovisa onsdag 1 februari (sjuk, vård av barn, personlig angelägenhet eller annat skäl som anses som giltig frånvaro på en arbetsplats):

- Orsak till varför du inte kan redovisa ska meddelas senast onsdag 1 februari 23.59.
- Du behöver inte göra en kompletteringsuppgift men du och den du jobbar med får redovisa inför Martin **torsdag 2a februari** eller **fredag 10a februari**

Vad händer om jag redovisar onsdag 1 februari men inte når kraven för godkänt?

- Du kan inte få VG längre
- Du och din grupp har möjlighet att redovisa på nytt i mån av tid för Martin **torsdag 2a februari** eller **fredag 10a februari**