**Clean Code examinering – Rapport** **John Sjöö .NET2020**

Programmet som har refaktoriserats är gjort efter ett MVC-mönster där Model innehåller PlayerData-klassen samt mappen ”Games” som innehåller spelens logik och det generella interfacet för båda spelen. View håller reda på samtliga in och utmaningar från användaren och dessa ut och inmatningar beskrivs i ett interface som heter ”IUserInterface”. Till sist har vi Controllern som sköter programmets flöde.   
  
Jag valde att implementera ytterligare ett spel likt Bulls and cows där användaren istället gissar på ett hemligt tal. Spelet hjälper dig med ledtrådar i form utav meddelanden ”Higher” eller ”Lower”.  
  
För att beskriva flerspelsdynamiken i programmet så ärver båda spelen av ett interface som heter ”IGuessGame”. Båda spelen delar beteende och har därför likadana metoder men med annorlunda logik. För att lägga till ett nytt spel behövs endast en ny klass med logik som ärver utav IGuessGame-interfacet. Sedermera implementeras spelet via konstruktorn i Controllern med Dependency injection. Slutligen kallas metoden ”Play” med instansen av ditt nya spel som parameter. För att uppnå den här dynamiken har jag inspirerats av ”Strategy-pattern”.  
  
En ny resultat-fil skapas för varje ny typ av spel då vi skickar med en sträng ([”GameType”].GetType().Name;) som blir en del utav filnamnet.  
  
I Controllern har jag delat upp de olika ansvarsområdena i olika metoder för att göra det tydligt i vilken ordning operationerna utförs. Dessa metoder är också generella för båda spelen och återanvänds oavsett vilket spel du använder.   
  
Utöver dessa ändringar så har jag ändrat några namn på variabler och metoder som är mer beskrivande av deras uppgift eller syfte.  
  
MsTest har gjorts på logiken som finns i Bull and cow game och blir godkända. Även namnen på metoderna i testerna har namn som är beskrivande för vad dom testar.