

De ändringar jag har gjort var att säkerställa självständighet mellan klasserna. I den förra koden var Dice-klassen beroende av DiceApp-klassen för sin grafiska representation, vilket skapade en för stark koppling mellan de två klasserna.

För att lösa detta har jag flyttat all kod som hanterar den grafiska representationen av en tärning från DiceApp-klassen till Dice-klassen. Det innebär nu att varje Dice-instans skapar och hanterar sitt eget li-element, vilket representerar tärningen i DOMen.

Inom DiceApp-klassen har jag modifierat add, remove och roll metoderna för att utnyttja dessa nya egenskaper i Dice-klassen. Nu när add-metoden anropas skapas en Dice-instans och dess roll-metod anropas, sedan läggs dess li-element till i DOMen genom att appenda ett child till DiceApps egenskap "ulForContent". Inom remove-metoden tas Dice-instansens li-element bort från DOM innan instansen själv tas bort. Inom roll-metoden anropas roll-metoden för varje Dice-instans, vilket uppdaterar både dess value-egenskap och dess grafiska representation utan att behöva påverak DiceApps klassens "ulForContent" egenskap.

Genom dessa ändringar bör Dice-klassen nu kunna användas oberoende av DiceApp-klassen.