

## **Hur vi applicerat MVC**

I vårt spel använder vi en lite anpassad variant av designmönstret MVC. Det som gör att vi inte följer det klassiska MVC-mönstret helt är bland annat att biblioteket är uppbyggd på ett sätt som gör det lite komplicerat att följa MVC helt.

### **Import av biblioteksklassen Vector2 i modellklasserna**

Vi valde att tillåta import av klassen Vector2 i våra modellklasser då detta underlättar och bidrar till en bra kontinuitet i koden. Många av de metoder och klasser som används i samband med fysik har en Vector2 som parameter. till exempel så beskrivs nästan alla positioner som en Vector2 pos(float x, float y). Vi vill att positioner skall lagras i modellen och därmed blir det klumpigt att skriva isär posX och posY som två olika variabler med varsin get- och set-metod. Vi skulle kunna skriva en egen klass som fungerar precis som en Vector2, men detta innebär ändå att vi måste konvertera till en Vector2 vid många tillfällen i samband med skapandet av fysikelement.

### **Render-metod i Controller-klasser?**

Render-metoden i våra klasser är autogenererade metoder som måste finnas då dessa klaser implements klassen "Screen" från libgdx. Render är den metod som körs varje gång spelet ska renderas/uppdateras. Därför använder vi denna metod för att kalla på andra metoder som uppdaterar fysik och även grafik. Själva utmålandet av grafiken sker såklart i view-klassernas draw-metoder.