

Developer manual

Vi har utvecklat ett spel till Android och det har vi gjort med hjälp av ett bibliotek/framework som heter Libgdx. Vi har även använt oss av en fysikmotor som heter Box2D. Libgdx fungerar så att man har ett separat java-projekt med all sin källkod samt utöver detta så har man ett projekt per platform som man vill utveckla för. När man kör igång projektet på en android telefon så startar man java-projektet "zombie-world-android". Libgdx sköter att källkoden som används är ifrån "zombie-world" paketet.

När spelet startas så skapas en instans av ZombieWorld.java denna klass ärver från Libgdx-klassen Game som har en create metod. Denna metod körs igång och skapar en ny screen. Vår applikation är uppbyggd av flera screens. De två viktigaste är MainMenu och GameScreen. När man klickar på Start i MainMenu så skapas en ny skärm, GameScreen, det är här all magi händer.

Man kan säga att GameScreen fungerar som vår controller. Det finns en render metod som körs idealt 60ggr per sekund (60 FPS), med reservation av fördröjning.

Metoden show i GameScreen körs direkt när man byter till denna screen. Här så skapar vi allt vi behöver för att spelet ska kunna spelas. Vi har en modell-klass, WorldModel, samt en vy-klass, GameView.

GameScreen

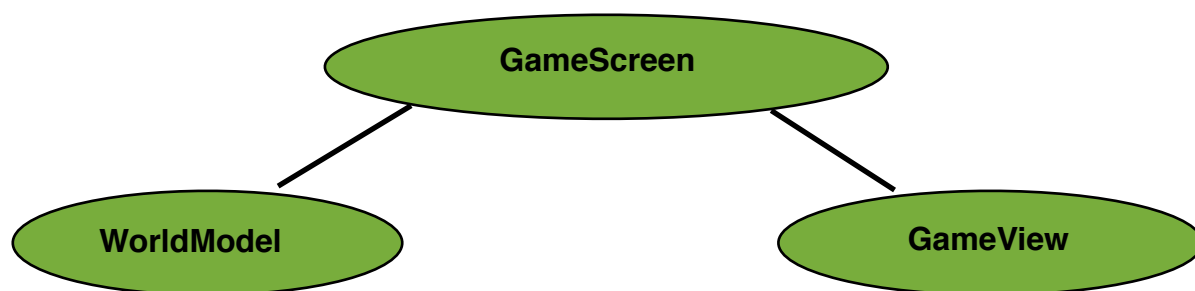
Den här klassen har en render-metod som sköter allt.

GameView

Den här klassen renderar all grafik på skärmen.

WorldModel

Den här klassen har en Box2D värld (world) där all vår fysik hanteras. Alla objekt som springer runt i världen är Box2D bodies som interagerar med varandra mha vår fysikmotor. All artificiell intelligens som finns i vårt spel hanteras av en klass som heter AIModel. Denna updateras hela tiden i WorldModel.



Vi har ytterligare en viktig klass som bearbetas under spelets gång och det är StageModel. I den klassen så hanteras UI:n dvs de två joysticks som finns och QuickSelection menyn. StageModel innehåller statiska variabler som representerar Joysticks och allt annat som går att interagera med på skärmen. I render metoden i GameScreen så kollar applikationen om användaren har rört något på skärmen och processerar detta, tex spelarens movement.