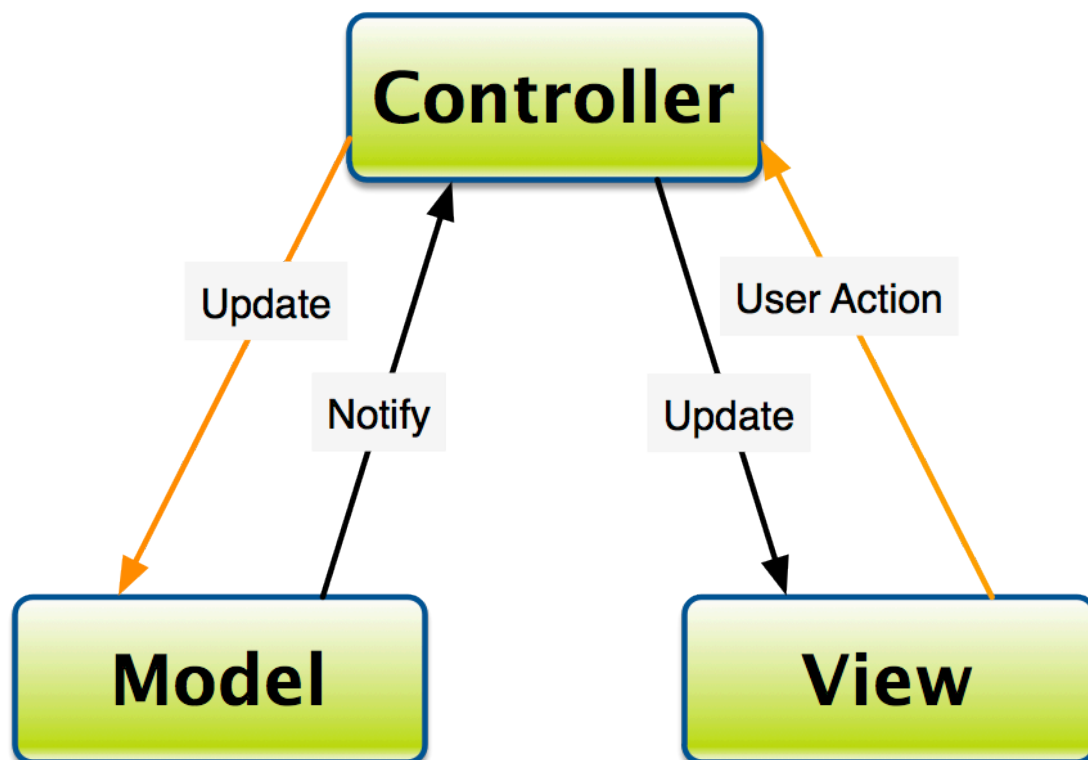


Coloristance: Design Decisions

Latest version: 2013-05-31

Designbeslut i Coloristance-projektet gjordes huvudsakligen utifrån tre grundläggande komponenter: spelidé, gruppens kunskap om programmering och MVC. Utöver det så fanns tankar om intuitivitet, "touchability", tillgänglighet (framför allt för olika devices) med i tankarna och satte prägel på designen, dessa var dock senare designbeslut när grunden redan var lagd.

När det gällde vilka funktioner Coloristance skulle ha så blev det en kompromiss mellan vad vi ville ha och ansåg mest viktigt för spelupplevelsen (spelidén) och vilken kompetens som fanns inom gruppen samt vad vi trodde oss ha möjlighet att hinna lära oss (kunskap). När detta var fastställt användes MVC (Model, View, Controller) som ett sätt att strukturera upp vilka filer som skulle finnas, vad som skulle finnas i dem och på vilka sätt de skulle kommunicera med varandra via metoder osv.

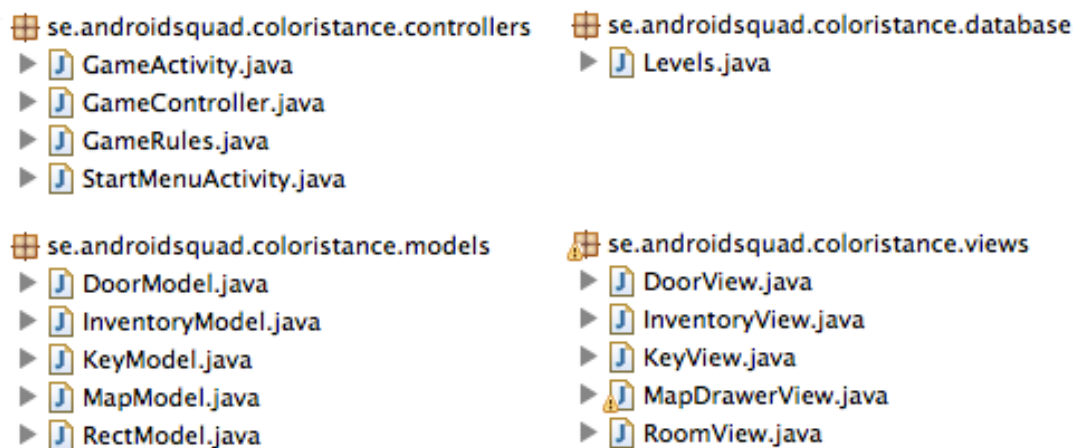


Att MVC användes berodde mycket på att vi tidigt identifierade ett behov av en tydlig struktur för att kunna dela på den totala uppgift gruppen hade. Detta behov bollades med handledaren som nämnde MVC som en metod som kunde ge tydliga riktlinjer sett till programmeringen för att lättare kunna skapa tydliga delar som sedan skulle gå att fördela inom gruppen.

För att så snabbt som möjligt få ett grepp om vad som behövdes, användes whiteboard där tre kolumner (M-V-C) ritades upp och sedan gick gruppen gemensamt igenom spelidén och försökte placera ut korresponderande filer.

MVC- tänket användes under hela projektet men som riktlinjer och inte regler då det i vissa fall var svårt att sära på kod-segment. Dessa problem grundades i Android-bibliotekets utformning i kombination med kunskapsnivån inom gruppen. Där det ansågs nödvändigt/värdefullt sett till MVC-metoden så införskaffades kunskapen men smärre blandningar av de tre områdena förekommer i koden precis som viss duplicering, vilket var aktiva beslut då tiden att införskaffa kunskapen för att höja kvaliteten ansågs som ett för högt pris. Exempel på Android-bibliotekets inverkan är att Activities i vår kod är en blandning mellan View och Controller där vi i de flesta fallen försökt att dela upp koden så att klassen med en Activity är mer utav en Controller.

Spelidén är inte begränsad sett till externa beroenden och skulle kunnat länkas till en extern databas eller en highscore som delades mellan enheter vilket är en framtida möjlighet. Idag har dock Coloristance inga externa beroenden, gruppen resonerade att det var viktigare att skapa ett välfungerande spel än att kunna jämföra highscores på ett ickefungerande/buggat spel och lade därför kraft på att ha en mindre avancerad men mer välfungerande release vid kursens slut. På samma sätt valde gruppen att banor inte hämtades utifrån utan fanns inkluderade i projektet. Även detta är ett möjligt senare steg (lägga banor på en server som spelet läser från) då det skulle vara intressant att ge användaren möjlighet att skapa egna banor och dela dem med andra användare.



För att tydligt följa MVC skapades paket i projektet som hette exempelvis "coloristance.views" och i detta paket låg alla views. Utöver Model, View och Controller-paket skapades ett databaspaket och ett test-paket för att hålla Coloristance som program tydligt avgränsat.

User Interface

Vi insåg tidigt vikten av ett tilltalande användargränssnitt för att spelet skulle lyckas. Ett snyggt spel fångar potentiella spelares uppmärksamhet så att de blir

motiverade att ladda ner applikationen. Väl spelandes kan de njuta av spelidén istället för att irritera sig på den grafiska utformningen.

Det första beslutet vi tog var att använda Androids egna färgschema för att det skulle se ut och kännas som Android och på så vis ge proffsigt intryck. Detta då användaren är van vid färgsammansättningen som är framtagen av personerna bakom operativsystemet. Majoriteten av spelkomponenterna har utformats efter *flat design* med anledning av den nya trend som råder. Tydliga exempel på detta är Windows 8 och Google Now som övergått mot kantigt och platt istället för rundade hörn som till exempel Apple tillämpar. Genom detta har vår ambition varit att skapa en bekant miljö för spelaren som upplevs som modern och tilltalande. Applikationen har delats upp i tre huvudsakliga delar där respektive del innehåller viktig information för spelaren. Dessa delar är karta, rum och inventory som tillsammans skapar spelmiljön Coloristance. Komponenternas utplacering grundades i våra tidigare spelerfarenheter för att säkerställa att spelaren upplever utplaceringen som logisk. För att applikationen ska vara spelbar oavsett skärmstorlek så har allting utvecklats mot att vara skalbart. På så vis är det enkelt att använda touch-funktionen oavsett om skärmen är 3.5" eller 10.1". En stor del av arbetet har varit att skapa en intuitiv spelmiljö som är enkel och går fort att sätta sig in i. Detta har uppnåtts genom att använda testpersoner som aldrig tidigare testat applikationen. Efter deras feedback har vi kunnat göra justeringar med tydlighet och intuitivitet som ledord.