

## Kursen

- Bättre ramar för vad kursen gick ut på och vad man förväntas leverera, t.ex. skulle man kunna ha ett visst antal förubestämda projekt att välja bland, som lämpar sig för arbetssättet. Det borde även framgå tydligare att fokus ska ligga på arbetssättet kring projektet, dokumentation och liknande, och inte på själva produkten. Det hade inte varit helt fel om hela kursen låg kring agil systemutveckling, och tydligare så. Som det var nu tror jag inte vi var de enda som trodde detta var en projektkurs där man skulle skriva lite dokument vid sidan av.

## Projektarbetet

- Vi valde att arbeta genom Scrum för att kunna utveckla iterativt och flexibelt, men projektet vi valde passade inte riktigt in i arbetssättet och vi kände att vi inte kunde utnyttja styrkorna med Scrum till sin fulla potential. Mycket låg i de korta sprintsen, det blev en ständig avvägning mellan att forma applikationen i strategisk ordning och att göra acceptabla implementationer av bakomliggande kod, för att snabbt klara av att skriva de som var högst prioriterade ur användarsyfte.
- Vi valde att inte använda oss utav en så kallad "estimated cost" i vårans Project Backlog, dels då vi hade alltför lite tidigare erfarenhet av de saker som skulle genomföras för att kunna göra en rimlig uppskattning och dels på grund ut av den korta tidsrymd som projektet skulle vara. Vidare upptäckte vi tidigt när vi försökte estimerar tid att varje element ur backlogen var så beroende på hur projektet i övrigt såg ut, att en estimering av tiden skulle bli så intetsägande att det helt skulle sakna mening att ens skriva ned den.
- Att arbeta i grupp innebär flera fördelar, men vi upptäckte också vissa problem. Något som tidigt visade sig svårt var att planera varje sprint, så att flera element inte var beroende på varandra. Flera gånger under projektarbetets gång befann vi oss i en situation där en eller flera utvecklare var tvungna att vänta på att ett element i veckans sprint blev färdigt, innan de själva kunde påbörja sitt arbete. Detta innebär att vi tappade viktig tid där vi borde varit mer produktiva. Sista veckorna var vi noga med att ta in detta i beräkning när vi valde element att implementera under sprinten, ibland fick högre prioriterade element lämna företräde för mindre prioriterade, som dock var mindre beroende av övriga element på veckans sprintlog.

## Projektet

- Valet att skriva vår applikation i C++ visade sig under projektets gång vara lite naivt. Vi valde C++ för att vi ville lära oss, för att kunna köra applikationen på fler

plattformar (iOS inte minst), och för en eventuell prestandaförbättring. Eftersom kursen innehöll såpass mycket dokumentation, och inte tillät oss så många timmar av ren programmering som vi förväntat oss, innebar övergången till ett nytt programmeringsspråk att allt tagit lite mer tid än vad som vore optimalt.

- Vi borde ägnat aningen mer tid till att skriva modeller ända ned till låg nivå, och på så sätt planerat applikationens struktur på ett lämpligare sätt. När vi bytte från egenskriven fysik till box2d innebar det att en del av projektet behövde skrivas om då vi inte hade ett väl definierat interface för fysiken. Andra klasser byggde istället till stor del på fysikens implementation.
- Vi spenderade mycket tid på att testa olika bibliotek för att importera filer direkt i C++/NDK. Först när det var färdigimplementerat i C++ insåg vi att det inte skulle fungera i Android. I Android har man inte tillgång till det interna filsystemet då filerna packas in i en APK fil som är en typ av zip. Vår lösning blir istället att ladda filerna då appen startar eftersom att Android kommer åt sina resurser väldigt lätt, detta fungerar också då resurserna inte tar upp mycket minne och alla resurser används från start. Hur filhantering går till i Android är något vi borde läst på mer om innan vi började leta efter bibliotek till C++, men det var samtidigt svårt att förutse då ingen ut av oss arbetat med NDK tidigare.
- Vi implementerade rendereringen genom OpenGL ES 1.0 i C++ och hade lite små problem med det i början då det inte ville renderera något på Android, koden var först testad på iOS. Vår lösning var att skriva renderering i OpenGL ES 2.0 istället men det gav samma problem på android som tidigare. Efter en del felsökning fick vi fram att OpenGL var implementerat på olika sätt för Android och iOS även om vi körde det i C++ vilket gav att vi fick ge vika smått på att C++ koden skulle vara helt plattformsoberoende. Vi kör nu i 1.0 med en `#ifdef __APPLE__` kontroll för iOS-specifikt beteende.
- Bytte fysikmotor sent eftersom det var mer komplext än beräknat (proprietär till Box2D).
- Blev tvungen att strunta i exception-hantering eftersom Android NDK inte har stöd för det.
- Tungt att lära sig GNU Make, Ant och hur kompilering av C/C++ fungerar. (Man får mycket gratis i Java).
- Aldrig tidigare använt OpenGL ES.