Reflektion TDDD57

Länk GitHub: <https://github.com/StahlErik/ToyCarMania>

Spel och interaktion är svårt. Det finns oändligt många sätt att göra det på, och det finns nödvändigtvis inget ”rätt”. Om rätt skulle definieras som intuitivt så uppstår problemet att intuitivt är olika för olika användare. Med det sagt är ändå målet för många utvecklare att göra spelet så intuitivt för så stora massor som möjligt. I och med AR har en uppsjö av olika applikationer uppkommit, både spel och verktyg. I min erfarenhet så har en hel del (inte alla) AR-spel försökt ta gamla koncept och applicerat dom på nya format. Till exempel ett strategispel där många knappar finns i gränssnittet, men AR-upplevelsen är mest till för att rita upp världen på ett häftigt sätt. Ett annat exempel är schack, också ett statiskt spel, där spelaren flyttar pjäser med fingrarna. Jag hittade dock ett spel som nyttjade tekniker lite annorlunda, Knightfall. Det här demospelet var ett Tower defence, men där du som spelare siktade med kameran och sköt sedan katapulter och pilar med knapptryckningar. Detta blev intressant då man verkligen kunde komma nära med kameran, sikta och skjuta. Mina tankar började här spinna vidare i idéen att helt komma bort från knappar, och att använda kameran som ett sikte. Eftersom AR är ett visuellt baserat koncept, som används främst i väldigt lätta och rörliga saker som mobiler och surfplattor, känns det för mig naturligt att komma ifrån knapptryckningar. Så då började jag skapa ett spel där det enda jag som spelare behövde göra, var att sikta med kameran. Om jag nu med siktet kan få en bil att köra dit jag pekar, så blir det plötsligt väldigt mycket enklare att styra. Om världen också reagerar på vad som händer med bilen och omgivningarna, så blir det genast ett roligt spel. Bara för att kontrollerna är enkla är inte spelet tråkigt. Flappy Bird hade en enda kontroll-input och sålde stort. När jag hade utvecklar klart mitt spel kände jag att det fanns många funktioner i mån om tid man skulle kunna lägga till, t.ex. multiplayer, hinderbanor, andra typer av fiender och liknande. Men kontrollerna hade jag låtit vara kvar, för dom fungerade väldigt enkelt och intuitivt för mig.

Kursen TDDD57 är en kurs för de med intresse. Jag har intresset, men inga tidigare kunskaper inom C# eller Unity, som jag använde. När jag fastnade i kodandet blev det lätt frustrerande. Jag kanske hittade en lösning på nätet, men av någon anledning fungerade det inte för mig. Och ofta tog det väldigt lång tid innan jag listade ut varför. Hade jag haft sviktande intresse tror jag därför denna kurs hade varit väldigt utmanande, då resurserna för hjälp är något begränsande. Jag tycker det är bra med möjlighet att träffa handledare på några utvalda tillfällen, men jag hade väldigt gärna sätt ett par kodstugor eller liknande också. Dessa tillfällen kunde vara bra för att fråga andra kursare hur dom tacklat problem, tips på tekniker mm. Även läraren kunde ge lite tips till alla med saker dom märkt att många fastnat på. Även ett bra tillfälle för alla att diskutera deras tolkningar på spelinteraktion och vad som utgör bra sådan. Kodstugan hade inte behövt vara oftare än varannan eller var tredje vecka. I övrigt gillar jag det fria upplägget, något som passar kursens innehåll. När det flyter på är det väldigt roligt att programmera spel.