Projektarbete

Jonas Hedman Engström
Emil Jacobsen
Jacob Kärrlander
Anton Lövström
Staffan Sandberg

9 maj 2013

Innehåll

1 Inledning

Vi har utvecklat ett spel för Windows vid namn “Johan Kannel - Click Detective”. Det är skrivet i C++ och OpenGL. Spelet går ut på att klicka bort rätblock som dyker upp på skärmen. Allt grafiskt och alla ljud i spelet är skapade av gruppens medlemmar.

Vi arbetade iterativt och tog hela tiden små steg i att göra spelet bättre. Vi hade en hel del buggar men lyckades till slut få fram ett spel som vi fann spelbart och smått underhållande. Vid två tillfällen under arbetets gång bestämde vi oss för att helt byta riktning på spelet och börja om, därför blev slutresultatet väldigt annorlunda mot det vi tänkte skapa från början.

2 Bakgrund

2.1 Syfte

Vårt primära syfte var att lära oss att använda OpenGL och C++ i spelutveckling samt lära oss att arbeta i en grupp och slutföra ett långsiktigt projekt.

Sekundärt så ville vi också få en produkt som vi kunde vara något stolta över och potentiellt kunna sälja.

2.2 Målgrupp

Målgrupp var inte så viktigt för oss eftersom vårt huvudsakliga syfte var att lära oss. Dock kan sägas att spelet passar sig bäst för personer som vill spela något snabbt, utmana sig själva och andra när det kommer till highscore och testa/träna sin reaktionsförmåga.

När spelet väl var klart kom vi att tänka på att man kan spela spelet på en smartboard. Smartboards är något som blir allt vanligare i svenska skolor vilket gör att allt fler lärare måste lära sig att hantera dessa. Vårat spel kan då vända sig till målgruppen "lärare som måste vänja sig vid smartboards".

2.3 Metod

I början använde vi oss i princip av vattenfallsmetoden. Vi satte upp en ambitiös, långsiktig tidsplan med väldefinerade ansvarsområden.

När vi sedan omfokuserade projektet så övergick vi till ett mer iterativt arbetssätt. Vi kortade ner tidsavstånden mellan prototyperna och satte upp mindre ambitiösa mål.

När inte heller det andra arbetssättet fungerade så tog vi det ännu längre åt det hållet. Vi kom på en mycket enkel spelidé som skulle gå att iterera över enkelt och snabbt. Vi var väldigt försiktiga med att sätta upp långsiktiga mål (det fanns inte så mycket tid kvar heller) och fokuserade på att göra små, men tydliga framsteg för att hålla motivationen uppe. Det här arbetssättet fungerade överlägset bäst för vår grupp.

3 Resultat

Vi lyckades producera ett smått trasigt men relativt funktionellt spel i slutändan. Vårt huvudmål uppfylldes för flera personer i gruppen. Resultatet blev mycket olikt våra initiala målsättningar, men de sista kravspecifikationerna vi hade uppfylldes till stor del.

3.1 Features

Spelet kom att innehålla två olika spellägen:

- Alpha. Detta spelläge går ut på att klicka bort tio rätblock på kortast möjliga tid. Spelläget fick sitt namn då den implementerades i alfa-versionen av spelet.
- Survival. Detta går ut på att man ska hålla ut så länge som möjligt mot rätblock som kommer inglidandes från högra sidan av skärmen. Om ett rätblock hinner nå den vänstra sidan av skärmen utan att bli bortklickat så förlorar man ett av de tre "liven". När "liven" tar slut förlorar man. Blocken kan ha fem olika färger och beroende på vilken färg det har så rör det sig olika snabbt. Desto snabbare blocket rör sig desto mer poäng får man för att klicka bort det.

Spelet innehåller också en elementär konfigurationsmeny där kan du ändra på saker och ting, allt från att höja och sänka volymen till att tysta den.

Spelet har en semi-fungerande textimplementering som vi använder för att förmedla information som poäng, tid och liknande till spelaren.

3.2 Buggar

Vi hade mycket problem med buggar i den tredje och sista fasen av projektarbetet. Det snabba, iterativa arbetssättet ledde till implementeringar som vi inte visste var trasiga förräns det hade hunnit bli svårt att hitta roten till problemet. Flera svåra buggar förblev olösta ända in i slutet och drar ner kvalitén avsevärt.

Rätblocken i spelet placeras pseudoslumpmässigt och tyvärr lyckades vi aldrig lösa problemet med att de hamnade "på varandra". Algoritmen som undersökte det var samma som klick-detekteringen och fungerade därför inte till fullo heller.

På viss hårdvara gick spelet knappt att köra. Som tidigare nämnts fungerade spelet bäst med grafikkort från Nvidia. Textimplementeringen fungerar heller inte så bra med smalare bokstäver som versala I till exempel, då det blir extra stora avstånd till den bokstaven.

4 Diskussion

4.1 Arbetsprocess

Vid två tillfällen under projektarbetets gång så struntade vi helt i vår dåvarande vision och tänkte om angående det mesta.

4.1.1 Storslagna Planer

Vår första målsättning för spelet var mycket ambitiös. Vår tidsplan innefattade en lång period av individuell inlärning då gruppmedlemmarna skulle lära sig olika saker inom sina ansvarsområden. Vi hade inga konkreta kortsiktiga mål i början av tidsplanen och situationen spårade ur då inga gruppmedlemmar tog så mycket eget ansvar eller lade så mycket tid som våra ambitiösa mål krävde.

Vårt mål var då att utveckla ett fysikbaserat skjutspel som spelas online, spelare mot spelare. Bara en fysikmotor hade varit överambitiöst i sig. Att vi dessutom skulle lära oss nätverkskodning och OpenGL var alldeles för mycket trots att vi planerat in tid för inlärning.

När inlärningsperioden var slut (vi förlängde den dessutom) så hade vi helt klart lärt oss alltför lite och vi hade problem med att implementera de simplaste av alla features som vi hade planerat för grovskissen. Det hela slutade med att vi hade möten och bestämde oss för att arbeta om vår vision och sikta lägre.

4.1.2 Management

Efter att vi insåg att vi aldrig skulle lyckas skapa ett spel liknande det vi tänkte oss i början bestämde vi oss för att byta genre och utveckla ett “management”-spel. Spelet skulle gå ut på att styra över en militär utpost genom att sköta dess ekonomi och produktion och till slut se till att basen överlever under en zombieapokalyps.

Problemet med detta spel var inte att det var för svårt att göra utan att det skulle ta för mycket tid och arbete innan spelet blev ett spel. När vi struntade i det här spelet och började på vår tredje och sista idé, så kodade Emil en prototyp som var måttligt underhållande med hjälp av koden från management-spelet på en dag.

4.1.3 Jackpot

Vår tredje och sista spelidé kallade vi för “Project Click”. Idén var simpel: låt spelaren klicka bort block. Det kraftfulla med det spelet var att idéer om små förbättringar och variationer av spelet kom som automatiskt för oss. Det var mycket enkelt att lägga till små förfinande lager på spelet, som redan i ett tidigt skede kändes som ett faktiskt spel.

4.2 Resultat

Problemet med rätblocken som hamnade i varandra skulle vi förmodligen kunna lösa om vi försökte. Algoritmen vi använde genererade en ny pseudoslumpmässig position för ett block om det kolliderade med ett annat, och förlängde alltså for-loopen en hel iteration. Istället för att generera en helt ny position vore kanske ett bättre sätt att korrigera positionen, flytta den lite till höger, vänster, upp eller ner och sedan söka efter kollision igen.

Problemet med smala karaktärer i textimplementeringen skulle relativt enkelt gå att lösas med en systerfil till fontfilen som skulle innehålla data om varje teckens bredd. Bäst vore dock att frångå en bitmap-lösning och istället använda sig av ett bibliotek som Freetype, som fungerar på samma sätt som operativsystem och webläsare när det kommer till att rasterisera text. Alltså generera tecknen ur geometrisk information istället för att klippa och klistra i en textur under runtime.

Algoritmen vi använde för bortklickning av block var bristfälllig. Mot kanterna av blocken var det

mycket mindre chans att klick registrerades som en träff. Algoritmen fungerade så att muspekarens tvådimensionella "fönsterposition" översattes till det närmaste synliga föremålet i 3D-rymden när man klickade. Buggen hade en koppling till grafiken eftersom att klick-detekteringen fungerade avsevärt mycket sämre på ATI-/AMD-grafikkort än Nvidia-grafikkort.

Problemet med dålig optimering på viss hårdvara berodde huvudsakligen på textimplementeringen. Det som skiljer texten i spelet från andra features, är att den använder texturer mycket. Sannolikt är alltså att det är i vårt hanterande av texturer som problemet ligger. Det är inte omöjligt att problemet skulle helt eller delvis lösas med kompression av texturerna i VRAM. Det ger ofta en mycket signifikant prestandaökning och det är väldigt ovanligt att inte komprimera texturer men vi hann inte med det under projektarbetets tidsram.

Alpha-spelläget skulle kanske bli bättre om blocken kom ett i taget istället för alla på en gång. Survivalspelläget skulle mått bra av bättre balansering mellan de olika blockens hastigheter och poäng. Det skulle nog varit roligare om ökningen av svårighetsgraden skedde logaritmiskt istället för exponentiellt så att det inte blev praktiskt omöjligt att överleva så tidigt.

Något som verkligen saknades i spelet var highscore-listor.

Konfigurationsmenyn var inte särskilt användbar, men lärorik.