

Saints of Chalmers

Ett fightingspel i två dimensioner

Frej Karlsson

frejk@student.chalmers.se

Institutionen för Informationsteknik

Arvid Björklund

cid@student.chalmers.se

Institutionen för Informationsteknik

Ludvig Ekman

eludvig@student.chalmers.se

Institutionen för Informationsteknik

Sammandrag

25 maj 2016

LSP310 V16

Kommunikation och ingenjörskompetens

Chalmers tekniska högskola

Innehåll

1	Inledning	1
1.1	Syfte	1
2	Metod	2
2.1	Tekniska val	2
2.2	Arbetsflöde	3
2.3	Avgränsning	3
3	Teori	4
4	Resultat	4
4.1	Beskrivning av spelet	4
5	Diskussion	5
5.1	Kodstruktur	5
5.2	Design	5
5.3	Slutsatser	6

1 Inledning

Datorspel har lyst upp människors vardag ända sedan de första spelen kom på sextiotalet. På åttiotalet blev arkadspel populära, med vilka man kunde spela en begränsad tid mot en direkt myntbetalning. Detta satte tydliga specifikationer för spelen, det skulle gå fort, det skulle inte gå att spara progress, och de som spelade mycket skulle belönas för att få incitament att spela mer. Arkadspelen seglades förbi redan nästa decennium, då konsollerna slog igenom. Nu ägde spelaren apparaten, så en tidsbegränsning var inte längre viktig. Spelen utvecklades till äventyr, och det var möjligt att fastna i samma spel i timmar. Denna utveckling följde med in i tvåtusen-talet tillsammans med persondatorn, och först för några år sedan såg man nästa steg i utvecklingen.

Idag har den första generationen där många spelade varje dag vuxit upp, och har som småbarnsföräldrar inte längre lika mycket tid till att spela datorspel. Detta gör att marknaden öppnas upp till nya möjligheter. De med liten tid uppskattar en kort intensiv upplevelse.

En spelgenre som åter blir populär är street fighting, där man möter en eller flera motståndare på en plattform. Det är också ett sådant spel som i detta projekt har skapats.

1.1 Syfte

Syftet med projektet är att göra ett underhållande spel med möjligheter till vidareutveckling. Spelet har genren Fighting där två spelare möter varandra i en arena med mål att döda motståndaren, liknande spelen Smash Bros¹ och Street Fighter². För att skapa ett underhållande fightingspel krävs det att spelet fungerar, att spelaren intuitivt förstår hur man spelar och vad målet är.

Ekman (2001)

(Ekman, 2001)

Just dessa tre punkter arbetar vi främst med. Att spelet fungerar är en självklarhet men att spelaren intuitivt förstår hur man spelar är svårare. Hur får vi användaren att förstå att spelet är ett fightingspel och att det går ut på att döda motspelaren? Vi vill kunna förmedla detta med hjälp av

¹<http://www.smashbros.com/en-uk>

²https://sv.wikipedia.org/wiki/Street_Fighter_II

en känsla som skapas av bland annat hintar baserade på grafik och ljud.

Hur får vi användaren att förstå målet och vägen dit utan att ge direkta anvisningar? (Karsllon, 2017)

2 Metod

Projektet hade som instruktioner att göra en skrivbords- (*desktop*) eller mobilapplikation (*mobile application*) i programmeringspråket java. Utifrån dessa premisser valdes flera tekniska hjälpmedel för att hjälpa till att organisera, koda och kommunicera.

2.1 Tekniska val

Vi har valt att använda dessa tekniska hjälpmedel:

IntelliJ

IntelliJ är en IDE (Integrated Development Environment, eller Integrerad utvecklingsmiljö) vi använt oss av till projektet. IntelliJ förenklar kodskrivandet genom att tillhandahålla en hel del verktyg som gör det enklare att skriva koden på ett effektivt och smidigt sätt, bland annat genom att redan innan man kompilerar ge varningar på direkta syntaxfel, onödig kod och duplicerad kod.

Git

Git är ett versionshanteringsprogram som används för att skriva kod tillsammans på ett effektivt och smidigt sätt.

JUnit

JUnit är ett ramverk som används för att skriva tester för javakod. Detta för att se till att koden funkar på rätt sätt och som det är tänkt. Genom att köra dessa tester varje gång man gör en ändring märker man enkelt och snabbt när något inte längre fungerar.

LibGDX

LibGDX är ett ramverk för spelutveckling både i 3D och 2D. Den tillhandahåller verktyg för att måla ut grafik, skapa animationer och dylikt.

Gradle

Gradle är ett verktyg som följde med i LibGDX, vi använde det för att hålla koll på våra beroenden (*dependencies*) så att man inte själv behöver hitta och ladda ner dessa varje gång man startar programmet på en ny dator.

Java

Java är det programmeringsspråk vi använt oss av vid utvecklandet av spelet. Språket är objektorienterat och kan enkelt användas för att utveckla applikationer till både till mobiler och datorer.

För att se länkar till dessa verktyg se bilaga 1 (vi har inte gjort bilagorna än)

IntelliJ valdes då de flesta i projektgruppen var erfarna med just denna IDE. Den har också stöd för vår projekt management tool Gradle och test plugin JUnit. Gradle valde vi för att vår grafiska motor LIBgdx använder sig av detta. JUnit var en obligatorisk del i projektet. För att koppla ihop den kod gruppmedlemmarna skriver används verktyget git som också var obligatoriskt.

2.2 Arbetsflöde

För att få ett bra arbetsflöde har gruppen träffats ca 2 gånger i veckan på så kallade scrum möten som bland annat går ut på att stämma av vad som behövs göra. Utöver detta har gruppen haft möjlighet att kontakta varandra dygnet om via kommunikationsverktyget Slack³. Detta har gjort så att vi kan ha en kontinuerlig utvecklingsprocess.

2.3 Avgränsning

Målet med projektet är ett funktionerande spel, där allt som kan göras ska fungera ordentligt. Därför var det viktigt att i början bestämma vilka funktioner och delar som är essentiella för spelidén, och göra dessa delar först. När grunden sedan är färdig går det att fortsätta på det som inte är absolut nödvändigt.

Det viktigaste, själva grunden, återfinns i syftet: "(...)två spelare möter varandra i en arena med mål att döda motståndaren." För detta krävs att spelarens karaktär:

³www.Slack.com

- Kan röra på sig
- Kan ta skada
- kan göra skada
- Kan förlora genom att tappa all hälsa

Tillhörande detta är grundläggande funktionalitet som att det måste finnas en arena att spela i, att spelarna ska kunna ha något att stå på och att det ska synas hur skadad en spelare är, samt grafik till allt detta.

Nästa steg i prioritering är sådant som inte är helt essentiellt för spelet, men som ger en helhet till spelet. Det innefattar:

- Startskärm
- Val av karaktär
- Val av arena

Till sist finns det som varken är essentiellt eller ger inramning. Det är extrafunktionalitet, som att spelaren kan använda sig av föremål som ger extra egenskaper, eller att måla ut en dynamisk bakgrund. Att göra flera olika karaktärer, olika arenor, olika attacker.

3 Teori

4 Resultat

4.1 Beskrivning av spelet

När applikationen startas öppnas en startskärm, därefter möts användarna av en sida för att välja karaktärer att spela med. När valet är klart får de välja vilken arena de vill spela på, och därefter startar spelet. I spelet finns möjligheten att pausa, och då dyker en pausmeny upp. När spelet avslutas – vanligen av att den ena spelaren har vunnit – visas avslutningssidan.

Startskärm

Var inte prioriterad i projektet, och hanns inte med. Hade

Val av spelare

En sida med en slags rutnätslayout där båda spelarna kan välj mellan de existerande karaktärerna.

Val av arena

Spelssida

I själva spelet visas spelkaraktärerna och den plattform de står på i profil, alltså i två dimensioner.

Pausmeny

Avslutningssida

5 Diskussion

5.1 Kodstruktur

MVC

5.2 Design

För att förenkla beslutsprocessen om olika designval fick ofta andra fightingspel stå som förebilder. Att kolla på andra spel gjorde det enklare att koncentrera sig på själva kodningen.

Healthbar

Sättet att visa spelarnas hälsa skiljer sig från liknande spel. Normalt sett har varje karaktär en så kallad Healthbar (hälsomätare), som visar hur många hälsopoäng (healthpoin) karaktären har. Om karaktären blir skadad med 10 skadepoäng (hitpoin) förlorar den 10 Healthpoin.

I spelet betar sig hälsomätaren annorlunda. Karaktärerna delar på samma hälsomätare, och har från början lika stor del av den. När en karktar skadas med 10 hitpoints skadas den inte bara genom att förlora 10 healtpoints, dessutom ökas den andra karaktärens hälsa med 10. Den totala hälsan minskas också med tiden, på så sätt har spelet en tidsbegränsning

där det till slut räcker med ett slag åt vardera håll för att döda den andra karaktären, om hälsan innan är helt jämnt fördelad mellan de båda.

Plattform

dsads

Chilla på detta lite det är in progress copy pasta från avgränsning, för mycket diskussion för att vara där (y)!

Under projektets gång har vi oftast valt att följa denna prioritering. När det har varit roligare att måla på olika karaktärer har det å andra sidan inte varit förbjudet. Målet har varit att hinna med både det som har högsta prioritet samt det som har näst högst, men det om bara det som har högst prioritet hinns med är det inte hela världen.

Om avgränsningen gör att extrafunktioner inte hinns med är det fortfarande viktigt att spelet enkelt ska kunna vidareutvecklas. I projektet finns till exempel en så kallad map factory, en klass för att välja flera olika arenor, även fast det bara finns en arena.

5.3 Slutsatser

Källor

Referenser

Ludvig Ekman. Ludvig den bästiga. *Dunno*, 2001.

Fuereuju Karsllon. How to nollkitta. *Tostesest*, 2017.